Intelligent Systems and Optimization

Research in Computing Science

Series Editorial Board

Editors-in-Chief:

Grigori Sidorov (Mexico) Gerhard Ritter (USA) Jean Serra (France) Ulises Cortés (Spain)

Associate Editors:

Jesús Angulo (France)
Jihad El-Sana (Israel)
Alexander Gelbukh (Mexico)
Ioannis Kakadiaris (USA)
Petros Maragos (Greece)
Julian Padget (UK)
Mateo Valero (Spain)

Editorial Coordination:

Alejandra Ramos Porras Carlos Vizcaino Sahagún

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. Volumen 147, No. 8, agosto de 2018. Tiraje: 500 ejemplares. Certificado de Reserva de Derechos al Uso Exclusivo del Título No.: 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. Certificado de Licitud de Título No. 12897, Certificado de licitud de Contenido No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: Grigori Sidorov, RFC SIGR651028L69

Research in Computing Science is published by the Center for Computing Research of IPN. Volume 147, No. 8, August 2018. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 147(8)

Intelligent Systems and Optimization

Miguel González Mendoza (ed.)



Instituto Politécnico Nacional, Centro de Investigación en Computación México 2018

ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2018

Instituto Politécnico Nacional (IPN) Centro de Investigación en Computación (CIC) Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal Unidad Profesional "Adolfo López Mateos", Zacatenco 07738, México D.F., México

http://www.rcs.cic.ipn.mx http://www.ipn.mx http://www.cic.ipn.mx

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX, DBLP and Periodica

Printing: 500

Printed in Mexico

Editorial

This volume of the journal "Research in Computing Science" contains selected papers related to intelligent systems, optimization and their applications. The papers were carefully chosen by the editorial board on the basis of the at least two reviews by the members of the reviewing committee or additional reviewers. The reviewers took into account the originality, scientific contribution to the field, soundness and technical quality of the papers. It is worth noting that various papers for this volume were rejected.

The volume contains 24 papers, which treat various aspect of modelling using Artificial Intelligence techniques in intelligent systems and optimization.

I would like to thank Mexican Society for Artificial Intelligence (Sociedad Mexicana de Inteligencia Artificial) and COMIA for their support in preparation of this volume.

The entire submission, reviewing, and selection process, as well as preparation of the proceedings, were supported for free by the EasyChair system (www.easychair.org).

Miguel González Mendoza ITESM México Guest Editor

August 2018

Table of Contents

	Page
Algoritmo basado en el forrajeo de bacterias para la optimización de un smart grid	11
Betania Hernández-Ocaña, Oscar Chávez-Bosquez, José Hernández-Torruco, Edgar A. Portilla-Flores, Luis G. Montané Jiménez, Margarita I. Hernández-Hernández, Jorge J. Brito-Ríos	
Diseño de una aplicación Web para el seguimiento y control del sobrepeso u obesidad a través del paradigma del IoT	25
Uriel Ramos-Deonati, Giner Alor-Hernández, Isaac Machorro-Cano, José Luis Sánchez-Cervantes, Lisbeth Rodríguez-Mazahua	
Hacia transacciones distribuidas coordinadas por agentes para la arquitectura de microservicios	39
Xavier Limón, Alejandro Guerra-Hernández, Angel J. Sánchez-García, Juan Carlos Peréz Arriaga, Juan Luis López-Herrera	
Máquina de estados finita con comportamientos de dirección para manipular agentes de juego RedCell	55
Propuesta de algoritmo GRASP para el problema de la tardanza total ponderada en una máquina	69
Rosmery Caridad Acosta-Núñez, Guillermo Airam Piloto-Cuellar, Ricardo Hernández-Zitlalpopoca, Santiago Omar Caballero-Morales	
Una comparativa de diferentes parámetros en la construcción evolutiva de descriptores para la clasificación de imágenes de texturas de piezas arqueológicas	83
Diseño y desarrollo de una arquitectura IoT en contexto con la plataforma FIWARE	95
Alicia Martínez, Haidée Onofre, Hugo Estrada, Daniel Torres, Olivia Maquinay	

Sistema de monitoreo de crecimiento de lunares para la detección temprana de cáncer de piel tipo melanoma basado en dispositivos móviles Android	107
Leonel Rosas Arias, Jesús Daniel Rojas Cid,	
Daniel Benito Román Ocampo, José Antonio López Tello, Waldemar Pérez Bailón	
Algoritmos para el control de indicadores clave de desempeño para Smart Cities	121
Elsa Estrada, Adriana Peña Pérez Negrón, Graciela Lara López, Luis Casillas	
Hacia un sistema de seguridad y alerta temprana de riesgos basado en agentes cognitivos	135
Wulfrano Arturo Luna-Ramírez, Anasol Peña-Rios, Emmanuel Ferreyra, Gustavo Ortiz Hernández, Carlos R. Jaimez-González	
Generación de control de dirección para vehículo autónomo por medio de programación genética1	149
Edgar Miguel Aguilar Diaz, Katya Rodríguez Vázquez	
Simulador con heurísticas de empaquetamiento para la asignación de memoria y despacho de procesos	161
Hilda Castillo Zacatelco, Rafael de la Rosa Flores, Claudia Zepeda Cortés, Ana Patricia Cervantes Márquez, Esly Cuautle Aguilar	
Implementación de agentes mediante Máquinas de Estado Finito de Comportamiento (MEFC) aplicados a la simulación de multitudes	175
Comparación de la convergencia de cuatro algoritmos evolutivos al explorar conjuntos de soluciones discretas y continuas	189
Ibis Azael Arenas-Marín, Anabel Martinez-Vargas, M.A. Cosío-León	
Cuerpos y planos geométricos usando realidad aumentada y computación afectiva	203
Ramón Zatarain-Cabada, María Lucia Barrón-Estrada, María Blanca Ibañez-Espiga, Aldo Uriarte-Portillo	

Ernest: sistema embebido para control de casas inteligentes mediante correos electrónicos con cifrado SSL	215
José Francisco Rodríguez Arellano, Emmanuel Dávila Delgado, Mario Alberto Ruíz Durán, Martín Isaac Falcón Segovia, Salvador Abraham Medina Rangel, Cristian Jael Mejía Aguirre	
Reducción de dimensionalidad en microarreglos: GN y GA	229
Modelo de datos para el monitoreo de variables agro-climatológicas de huertos urbanos utilizando internet de las cosas y cómputo en la nube	239
Algoritmo descentralizado de ruteo de tráfico vehicular con información local inspirado en el algoritmo de hormigas	249
Middleware aplicado a la simulación de fluidos con técnicas de realidad aumentada	265
Ambiente inteligente de aprendizaje de depuración de errores en programas Java	277
María Lucia Barrón-Estrada, Ramón Zatarain-Cabada, Minerva Valencia-Rodríguez, Gloria E. Peralta-Peñuñuri	
Optimización basados en caos para la búsqueda de objetivos	291
Upper Confidence Bound o Upper Cofidence Bound Tuned para General Game Playing: Un estudio empírico	301
Iván Francisco-Valencia, José Raymundo Marcial-Romero, Rosa María Valdovinos-Rosas	
C-PSO: Optimización por cúmulo de partículas incrustando la ley de Coulomb	311
Ricardo Solano Monje, Nayeli Joaquinita Meléndez Acosta, Cosijopii García García, Homero Vladimir Ríos Figueroa	

Algoritmo basado en el forrajeo de bacterias para la optimización de un *smart grid*

Betania Hernández-Ocaña¹, Oscar Chávez-Bosquez¹, José Hernández-Torruco¹, Edgar A. Portilla-Flores², Luis G. Montané Jiménez³, Margarita I. Hernández-Hernández¹, Jorge J. Brito-Ríos¹

¹ Universidad Juárez Autónoma de Tabasco, DAIS, Tabasco, México
² Instituto Politécnico Nacional, CIDETEC, Ciudad de México, México
³ Universidad Veracruzana, FEI, Xalapa, México
{betania.hernandez,oscar.chavez,jose.hernandezt}@ujat.mx,
{magui_hernandez96,jor.95}@hotmail.com, aportilla@ipn.mx, lmontane@uv.mx

Resumen. En este artículo, un algoritmo basado en el forrajeo de bacterias E.Coli llamado TS-MBFOA es implementado para optimizar un modelo matemático que minimiza a un *Smart Grid*, el cual es una red de distribución eléctrica combinada con modernas tecnologías de información que busca reducir los costos y la contaminación al medio ambiente. En este *Smart Grid* se hacen uso de fuentes de energía renovables como la eólica y solar, además de una unidad de generación convencional a partir del combustible diésel. TS-MBFOA fue implementado y probado en un primer experimento para calibrar los parámetros propios del algoritmo. En un segundo experimento, el algoritmo previamente calibrado fue ejecutado 30 veces de manera independiente para observar su rendimiento. Los resultados encontrados muestran que TS-MBFOA es mejor en comparación con dos algoritmos evolutivos utilizados para resolver el mismo problema de optimización.

Palabras clave: optimización, forrajeo de bacterias, algoritmo de inteligencia colectiva, red de distribución eléctrica inteligente.

Bacterial Foraging-Based Algorithm for Smart Grid Optimization

Abstract. In this paper, an E. Coli bacterial foraging-based algorithm called TS-MBFOA is implemented to optimize a mathematical model that minimizes a *Smart Grid*, which is an electrical distribution network combined with modern information technologies which seeks to reduce costs and environmental pollution. In this *Smart Grid*, renewable energy sources such as wind and solar are used, as well as a conventional generation unit from diesel fuel. TS-MBFOA was implemented and tested in a first experiment to calibrate the algorithm parameters. In a second experiment, the previously calibrated algorithm was executed 30 times independently to observe its performance. The results found show that

TS-MBFOA is better compared to two evolutionary algorithms used to solve the same optimization problem.

Keywords: optimization, bacterial foraging, swarm intelligence algorithm, smart grid.

1. Introducción

Debido al gran consumo de la energía eléctrica causado por la alta densidad demográfica que ha alcanzado nuestro planeta, a la preocupación del cambio climático por los altos índices de contaminación a causa del uso de aparatos eléctricos que emiten dióxido de carbono y a las fuentes de generación de energía eléctrica que pueden ser renovables o no renovables, se requiere del control y administración del consumo de energía eléctrica. Entre algunas medidas que se han tomado se encuentran las *Smart Grids*. Éstas son una herramienta tecnológica que permite un mejor desempeño de la energía eléctrica de manera inteligente (autónoma). Además, integra en una red las acciones de los usuarios y consumidores que se encuentran conectados a ella de manera inteligente, con el fin de conseguir un suministro eléctrico, seguro y sostenible [7]. La implementación generalizada de esta tecnología podría incidir en la reducción de los precios para todos los consumidores y reducir las emisiones de carbono por medio del mejoramiento de la eficiencia, de la respuesta a la demanda y de la gestión de la carga de la red eléctrica [4].

México ha puesto en marcha mecanismos para disminuir las emisiones contaminantes derivadas de la generación y el uso de la energía. Desde el punto de vista de la producción, uno de los principales objetivos de la Reforma Energética es que nuestro país utilice combustibles más limpios y mejores tecnologías. En esta lógica se inscriben los esfuerzos para sustituir el diésel y el combustóleo por gas natural en la generación de electricidad, al igual que el ingreso de empresas que utilicen procesos menos contaminantes para la extracción y transformación del petróleo [16].

El Centro de Innovación y Desarrollo Tecnológico en Cómputo (CIDETEC) del Instituto Politécnico Nacional (IPN) está trabajando en el desarrollo de modelos matemáticos para las *Smart Grids* y uno de ellos es propuesto como un problema de optimización que calcula las potencias óptimas de una red de dispositivos de generación de energía para suministrar a una carga durante un periodo de 24 horas. En esencia, cada hora se resuelve un problema de optimización de acuerdo a las condiciones de la red y restricciones de operación de la misma. Al final se tiene una gráfica de las potencias óptimas y la suma de las 24 funciones objetivo en la cual se propone qué solución es mejor utilizar.

Este tipo de problemas de optimización son complejos y aunque por programación matemática pueden resolverse, puede resultar complicado y caótico implementarla. Actualmente existen las metaheurísticas, herramientas que permiten resolver problemas de optimización con o sin restricciones, combinatorios o numéricos de manera aproximada, es decir, generan una o más soluciones

factibles cercanas al óptimo. Un problema de optimización es también conocido como un problema general de programación no-lineal y se puede definir como: Minimizar $f(\overrightarrow{x})$ sujeta a: $g_i(\overrightarrow{x}) \leq 0$, i=1,...,m y/o $h_j(\overrightarrow{x})=0$, j=1,...,p. Donde $\overrightarrow{x} \in \mathbb{R}^n$ tal que $n \geq 1$, es el vector de soluciones $\overrightarrow{x} = [x_1, x_2, ..., x_n]^T$, donde cada x_i , i=1,...,n está delimitada por el límite inferior y superior $L_i \leq x_i \leq U_i$; m es el número de restricciones de desigualdad y p es el número de restricciones de igualdad (en ambos casos, las restricciones podrían ser lineales o no lineales). Si denotamos con F a la región factible (donde se encuentran todas las soluciones que satisfacen al problema) y con S a todo el espacio de búsqueda, entonces $F \subseteq S$.

Los algoritmos bio-inspirados forman parte de las metaheurísticas y se dividen en dos grandes grupos: 1) los Algoritmos Evolutivos (AEs), cuyo funcionamiento se basa en emular el proceso de evolución natural y la supervivencia del más apto [6] y 2) los Algoritmos de Inteligencia Colectiva (AICs) que basan su funcionamiento en comportamientos sociales y cooperativos de organismos simples e inteligentes como insectos, aves y bacterias [8].

A partir de las ideas iniciales de Bremermann [3], en el año 2002 Passino [14] propone un AIC llamado Algoritmo de Optimización del Forrajeo de Bacterias (BFOA, por las siglas en inglés de Bacterial Foraging Optimization Algorithm), en el cual cada bacteria E.Coli trata de maximizar su energía obtenida por unidad de tiempo empleada en el proceso de forrajeo, donde también evade sustancias nocivas. Más aún, las bacterias se pueden comunicar entre sí mediante la segregación de sustancias. En BFOA se tienen cuatro procesos principales: (1) quimiotáxis (nado-giro), (2) agrupamiento, (3) reproducción y (4) eliminación-dispersión. Las bacterias son soluciones potenciales al problema y su ubicación representa los valores de las variables de decisión del problema. Las bacterias pueden moverse (generar nuevas soluciones) mediante el ciclo quimiotáxico; se genera además un movimiento mediante la atracción que ejercen soluciones en zonas prometedoras del espacio de búsqueda, se permite la reproducción de las mejores soluciones y finalmente se eliminan del cúmulo aquellas bacterias localizadas en zonas de baja calidad.

En lo referente a propuestas de solución de instancias del problema de optimización global con restricciones, en 2009 se propuso una versión simplificada de BFOA llamada Algoritmo de Optimización del Forrajeo de Bacterias Modificado (MBFOA, por sus siglas en inglés) [11]. En MBFOA se tiene un mecanismo para el manejo de las restricciones basado en Reglas de factibilidad, las cuales consisten en: a) Entre dos soluciones factibles, aquella con el mejor valor de la función objetivo es seleccionada. b) Entre una solución factible y otra no factible, la factible es seleccionada y c) Entre dos soluciones no factibles, aquella con la menor suma de violación de restricciones es seleccionada [5]. Por último, una disminución de parámetros respecto a los del BFOA original.

Además MBFOA se aplicó a un conjunto de problemas de diseño en ingeniería química y mecánica obteniendo resultados competitivos. Por otro lado, MBFOA se aplicó a la resolución de un problema de diseño mecánico bi-objetivo en presencia de restricciones [12]. Una propuesta más reciente basada en MBFOA

usa un operador similar al de mutación (usado en AEs) como operador de nado dentro del proceso quimiotáxico, además del uso de un nado aleatorio en este mismo proceso y un mecanismo de sesgo para la población inicial basado en el rango de las variables también fue anexado. Esta propuesta es llamada TS-MBFOA (por sus siglas en inglés) [9].

En este artículo se propone utilizar la metaheurística TS-MBFOA para minimizar la *Smart Grid* o Red de Energía Eléctrica Inteligente propuesta por el CIDETEC-IPN, puesto que en el estado del arte no se reportan trabajos similares usando este algoritmo, sin embargo, se han utilizado AEs para resolver instancias relacionadas con los *SmartGrids* [1]. Este documento está organizado de la siguiente manera: En la Sección 2 se presenta el modelado matemático del *Smart Grid*. En la Sección 3 se describe brevemente a TS-MBFOA. En la sección 4 se presentan los resultados obtenidos y la discusión de éstos. Por último en la Sección 5 se presentan las conclusiones y trabajos futuros.

2. Smart grids

Las redes de distribución eléctrica suelen alcanzar su máxima capacidad de operación y además deben adaptarse a ciertos cambios del medio ambiente que influyen en su funcionamiento. Esta problemática se soluciona mediante el uso de Generadores Distribuidos (GeDis) que permiten la explotación de Fuentes de Energía Renovable (FER); en este caso eólica y solar, y alternativa, que se refiere a los combustibles. A pesar de que los GeDis solucionan la problemática planteada, al introducir el uso de energías renovables surge el reto de su administración, que se vuelve complicado debido a la fuerte intermitencia que presentan dichas energías a causa de la climatología [2].

Las microrredes (MRs) permiten coordinar los GeDis para proporcionar electricidad de forma exitosa, lo cual las convierte en soluciones fiables para el reto mencionado. Una MR suele estar conformada por Sistemas de Almacenamiento de Energía (SAE), sistemas de generación híbrida a partir de FER y unidades de generación convencional a partir de combustibles como el diésel. En términos monetarios, los SAE son muy costosos y deben operarse de forma segura, garantizando que tengan una larga vida útil. Adicionalmente, las unidades de generación convencional tienen un alto costo de operación debido a la logística necesaria para transportarlas y a los materiales con los que están construidas. Es por esto, que los GeDis deben gestionarse de forma óptima para el funcionamiento de las MRs.

En este caso de estudio, se debe optimizar la generación de energía en una MR que se encuentra en un lugar Remoto No Interconectable (RNI). En estos lugares, rurales en la mayoría de los casos, la generación fiable de energía se consigue a través de generación híbrida de potencia en donde se emplean FER, generación convencional con diésel y SAE. El reparto de potencias para suministrar energía a la carga de una Microrred Remota No Interconectable (MR-RNI), se calcula como una función de despacho económico usualmente empleada para generadores de gran escala de potencia [19], en donde se establece una función de costo a

minimizar sujeta a restricciones de capacidad de los generadores y de balance entre la carga generada y demandada.

Microrred Remota No Interconectable (MR-RNI) 2.1.

La MR-RNI está conformada por un conjunto de cargas y un sistema híbrido de generación de potencia que a su vez se compone de un sistema de Generación Eólico (GE), un sistema de Generación Solar Fotovoltáico (GSF), un Generador de Diésel (GD) y un SAE. Las variables de control son las potencias de los generadores y el SAE, y fungen como entradas del sistema de optimización. Las salidas de control, son el reparto de carga por cada generador y el costo de generación. A continuación, se describe el cálculo del costo para cada generador.

Costo de Generación Diésel. Los GD tienen una función de costo asociada a la potencia del generador. Dicha función se encuentra dada por la Ecuación 1, en donde i es la fuente de generación i, P_i y F_i son la salida de potencia eléctrica y el costo de operación de la fuente i, respectivamente, mientras que α , β y γ son los coeficientes de costo.

$$f_i(P_i) = \alpha_i + \beta_i P_i + \gamma_i P_i^2. \tag{1}$$

Para el GD, la función de costo queda como lo muestra la Ecuación 2:

$$f_1(P_1) = 14.88 + 0.3P_1 + 0.000435P_1^2.$$
 (2)

Costo de Generación Solar Fotovoltáica. Para el generador GSF, la función de costo se define en la Ecuación 3:

$$f_3(P_3) = \alpha I^P P_3 + G^E P_3, \tag{3}$$

$$f_3(P_3) = \alpha I^P P_3 + G^E P_3,$$

$$\alpha = \frac{r}{[1 - (1+r)^{-N}]},$$
(3)

en donde P_3 es la potencia de salida de GSF, α es la tasa de retorno de inversión y se calcula con la Ecuación 4, r corresponde a la tasa de interés (se considera de 0.09 para el caso base), N se refiere a la vida útil, que ha sido establecida en 20 años, I^p indica el costo de inversión por unidad de potencia (\$5000/kW) y G^E es el costo de operación y mantenimiento por unidad de potencia generada (c1.6/kW).

De acuerdo a la Ecuación 3, la función de costo para GSF queda establecida con la Ecuación 5:

$$f_3(P_3) = 545.016P_3. (5)$$

Costo de Generación Eólica. Siguiendo el esquema de costos presentado en GSF, y tomando los mismos valores de I^p y G^E , la función de costo para la generación eólica se define con la Ecuación 6:

$$f_4(P_4) = 152.616P_4. (6)$$

Costo de Sistema de Almacenamiento de Energía. Finalmente, para el SAE se considera un banco de baterías de 2kW. El costo I^p por unidad de

15

Betania Hernández-Ocaña, Oscar Chávez-Bosquez, José Hernández-Torruco, et al.

almacenamiento instalada es de \$1000/kW, mientras que el G^E conserva el valor de \$c1.6/kW\$ dando como resultado la siguiente función de costo definida en la Ecuación 7:

$$f_2(P_2) = 119P_2. (7)$$

2.2. Declaración del problema de optimización

Para la optimización de la generación de energía se emplea el método de despacho económico, en donde se deben encontrar las potencias que serán suministradas por los generadores de la MR-RNI en cada hora del día, de tal forma que el costo sea lo menor posible. Para este caso de estudio, el vector de variables de diseño se define en la Ecuación 8:

$$\overrightarrow{p} = [P_1, P_2, P_3, P_4]^{\tau}. \tag{8}$$

Función Objetivo. La formulación clásica para el despacho económico es definida con la Ecuación 9:

$$Min \ F = \sum_{i=1}^{NG} f_i(P_i).$$
 (9)

La función objetivo considerada en la gestión de la generación para la MR-RNI en la hora τ es definida con la Ecuación 10:

$$Min\ f(\overrightarrow{p}_{\tau}) = w_1 C_f f(P_1(\tau)) + w_2 f(P_2(\tau)) - w_3 f(P_3(\tau)) - w_4 f(P_4(\tau)),$$
 (10)

en donde w_1 , w_2 , w_3 y w_4 son los pesos asignados a los generadores, los cuales suman 1 y han sido establecidos en 0.25, mientras que C_f es el costo del combustible diésel, que se asignó en USD \$1. Considerando que el costo de generación se debe calcular por cada hora del día, la función objetivo para el costo total durante un día completo queda definida con la Ecuación 11:

$$Min \ F = \sum_{\tau=1}^{24} f(\overrightarrow{p}_{\tau}) \qquad \overrightarrow{p}_{\tau} \in \mathbb{R}^{4}. \tag{11}$$

Restricciones de diseño. Para que la gestión de la generación se realice de forma correcta, deben considerarse las siguientes restricciones:

Balance de potencia: La suma de las potencias suministradas por los generadores debe ser igual a la carga total del sistema, esto es definido en la Ecuación 12:

$$P_1 + P_2 + P_3 + P_4 = P_L. (12)$$

 $Modelo\ del\ banco\ de\ baterías$: El poder de salida del generador PV y la carga demandada en cierta hora t, determinan el poder de carga y descarga dentro y fuera del banco de baterías. t es un entero que representa la t-ésima hora.

Algoritmo basado en el forrajeo de bacterias para la optimización de un smart grid

Tabla 1. Parámetros del Sistema de Almacenamiento de Energía (SAE).

Parámetros	%
Eficiencia de carga (n_C)	85
Eficiencia de descarga (n_D)	100
Estado máximo de carga (SOC^{max})	95
Estado mínimo de carga (SOC^{min})	40

El Estado de Carga (SOC) del banco de baterías en cualquier hora, SOC(t), depende del SOC en la hora previa SOC(t-1). Para el flujo de energía de la hora t-1 a t, se debe considerar la Ecuación 13:

$$SOC_t = SOC(t-1) - \alpha_D P_2(t) + \alpha_C P_3(t) + \alpha_C P_4(t), \tag{13}$$

en donde $\alpha_D = \frac{n_D}{Bc_{max}}$, $\alpha_C = \frac{n_C}{Bc_{max}}$, mientras que n_C y n_D son la eficiencia de carga y descarga de la batería, respectivamente. La Ecuación 14 se puede aplicar para la dinámica de la batería [17]:

$$SOC(t-1) = SOC(0) - \alpha_D \sum_{\tau=1}^{t} P_2(\tau) + \alpha_C \sum_{\tau=1}^{t} P_3(\tau) + \alpha_C \sum_{\tau=1}^{t} P_4(\tau), \quad (14)$$

en la cual SOC(0) es considerado como el SOC inicial de la batería. $\alpha_C \sum_{\tau=1}^t P_3(\tau) + \alpha_C \sum_{\tau=1}^t P_4(\tau)$ es el poder aceptado por la batería en la hora t y $\alpha_D \sum_{\tau=1}^t P_2(\tau)$ es el poder de descarga de la batería en el tiempo t.

La capacidad disponible de la batería no debe ser menor que la capacidad mínima permitida ni mayor que la capacidad máxima permitida [17], esto es definido por la Ecuación 15:

$$SOC^{min} \le SOC(t) \le SOC^{max},$$
 (15)

es decir, detalladamente con la Ecuación 16 :

$$SOC^{min} - SOC(t) \le 0$$
, $SOC(t) - SOC^{max} \le 0$. (16)

Los parámetros del banco de baterías se muestran en la Tabla 1.

Finalmente, el problema de optimización queda descrito por las Ecuaciones 17 a 20.

$$Min \ F = \sum_{\tau=1}^{24} f(\overrightarrow{p}_{\tau}) \qquad \overrightarrow{p}_{\tau} \in \mathbb{R}^4, \tag{17}$$

sujeto a:

$$g_1(\overrightarrow{p}_{\tau}) = P_1(\tau) + P_2(\tau) + P_3(\tau) + P_4(\tau) = P_L(\tau),$$
 (18)

$$g_2(\overrightarrow{p}_{\tau}) = SOC^{min} - SOC(t) \le 0, \tag{19}$$

$$g_3(\overrightarrow{p}_{\tau}) = SOC(t) - SOC^{max} \le 0. \tag{20}$$

Betania Hernández-Ocaña, Oscar Chávez-Bosquez, José Hernández-Torruco, et al.

La carga $P_L(\tau)$ fue obtenida de [18] y los límites superior e inferior de cada variable se encuentran descrito por las Ecuaciones 21 a 24:

$$0 \le P_1(\tau) \le GD_{nominal},\tag{21}$$

$$0 \le P_2(\tau) \le SOC(0) \times Bc_{max} - SOC^{min} \times Bc_{max}, \tag{22}$$

$$0 \le P_3(\tau) \le P_{pv}(\tau),\tag{23}$$

$$0 \le P_4(\tau) \le P_{wind}(\tau),\tag{24}$$

en donde $GD_{nominal}$ se refiere a la capacidad nominal del generador diésel, que para este caso es de 5000 VA (5 kVA). El límite superior para $P_2(\tau)$ depende de la capacidad máxima de la batería (Bc_{max}) que es de 2000 VA (2 kVA) y el estado inicial de la misma. Los datos de $P_{pv}(\tau)$ y $P_{wind}(\tau)$ donde P_{pv} es la potencia de salida fotovoltáica y P_{wind} es la potencia de salida eólica y fueron tomados de [13].

3. TS-MBFOA

El TS-MBFOA es un algoritmo derivado de MBFOA propuesto para resolver PONR [9], en el cual una bacteria i representa una solución potencial y se denota como $\theta^i(j,G)$, donde j es el ciclo quimiotáxico y G es el ciclo generacional. Una generación consta de un proceso quimiotáxico, agrupamiento, reproducción y eliminación-dispersión.

En el proceso de quimiotáxis dos nados se intercalan, en cada ciclo solo un nado de explotación o exploración es realizado. El proceso comienza con el nado de explotación (nado clásico). Sin embargo, una bacteria no necesariamente intercalará exploración y explotación en los nados, ya que si la nueva posición de un nado dado, $\theta^i(j+1,G)$ tiene una mejor aptitud (basado en las reglas de factibilidad) que la posición original $\theta^i(j,G)$, otro nado en la misma dirección se llevará a cabo en el siguiente ciclo. De lo contrario, un nuevo giro será calculado. El proceso se detiene después de N_c intentos.

El nado de exploración usa la mutación entre bacterias y es calculado con la Ecuación 25:

$$\theta^{i}(j+1,G) = \theta^{i}(j,G) + (\beta)(\theta_{1}^{r}(j,G) - \theta_{2}^{r}(j,G)), \tag{25}$$

donde $\theta_1^r(j,G)$ y $\theta_2^r(j,G)$ son dos bacterias diferentes seleccionadas aleatoriamente de la población. β es un parámetro definido por el usuario utilizado en el operador de agrupamiento el cual define la cercanía de la nueva posición de una bacteria con respecto a la posición de la mejor bacteria de la población, en este operador, β es un parámetro de control positivo para escalar los diferentes vectores en (0,1], es decir, escalas de la zona donde una bacteria puede moverse.

El nado de explotación es calculado con el Ecuación 26:

$$\theta^{i}(j+1,G) = \theta^{i}(j,G) + C(i,G)\phi(i),$$
 (26)

donde $\phi(i)$ se calcula con el operador de giro original de BFOA definido en la Ecuación 27:

$$\phi(i) = \frac{\Delta(i)}{\sqrt{\Delta(i)^T \Delta(i)}},\tag{27}$$

 $\Delta(i)^T$ es un vector aleatorio generado con elementos dentro de un intervalo[-1,1]. C(i,G) es el tamaño de paso aleatorio de cada bacteria actualizado con la Ecuación 28:

$$C(i,G) = R * \Theta(i), \tag{28}$$

donde $\Theta(i)$ es un vector generado de forma aleatoria de tamaño n con elementos dentro del rango de cada variable de decisión decisión: $[U_k, L_k], k = 1,n$, y R es un parámetro definido por el usuario para escalar el tamaño de paso, este valor debe ser cercana a cero (por ejemplo 5.00e-04). La inicial C(i,0) se genera utilizando $\theta(i)$. Este tamaño de paso aleatorio permite que las bacterias se puedan mover en diferentes direcciones dentro del espacio de búsqueda y evita la convergencia prematura como se sugiere en [10].

En el ciclo medio del proceso quimiotáxico es aplicado el operador de agrupamiento con la Ecuación 29, donde β es un parámetro positivo definido por el usuario entre (0,1):

$$\theta^{i}(j+1,G) = \theta^{i}(j,G) + \beta(\theta^{B}(G) - \theta^{i}(j,G)), \tag{29}$$

donde $\theta^i(j+1,G)$ es la nueva posición de la bacteria i, $\theta^B(G)$ es la actual posición de la mejor bacteria generacional y β es un parámetro llamado factor de escalamiento, el cual regula qué tan cerca estará la bacteria i de la mejor bacteria θ^B . Sin embargo, si una solución viola el límite de las variables de decisión, una nueva solución de x_i es generada aleatoriamente entre los límites inferior y superior $L_i \leq x_i \leq U_i$ de las variables de decisión.

En la reproducción se ordenan las bacterias con base en la técnica de manejo de restricciones, eliminando a las peores bacterias $S_b - S_r$ y duplicando a las mejores cada cierto número de ciclos, definido por el usuario con el parámetro RepCycle.

En la eliminación-dispersión se elimina a la pe
or bacteria de la población $\theta^w(j,G)$ (basado en las reglas de factibilidad) y se genera una nueva aleatoriamento

Aunque en su propuesta original de TS-MBFOA se utiliza un mecanismo de sesgo para generar la población inicial aleatoria y un buscador local, en este artículo no se hace uso de dicho mecanismo para consumir menos costo computacional. El pseudocódigo de TS-MBFOA es presentado en el Algoritmo 1.

4. Resultados

TS-BFOA fue adaptado y ejecutado para resolver el problema del *Smart Grid* en tres equipos de cómputo con las siguientes características: una PC con Memoria RAM de 4.0GB, procesador de 2.3Ghz y un sistema Operativo

Algoritmo 1: Pseudocódigo de TS-MBFOA.

```
1 Crear una población inicial de bacterias aleatorias \theta^i(j,0) \ \forall i,\ i=1,\ldots,S_b
 2 Evaluar f(\theta^i(j,0)) \ \forall i, i = 1, \dots, S_b
   for G=1 to GMAX do
 3
        for i=1 to S_b do
            for j=1 to N_c do
 5
                 En el proceso quimiotáxico intercalar los nados propuestos con las Ecs.
 6
                 25 y 26. Aplicar el operador de agrupamiento con la Ec. 29 usando \beta para la
                 bacteria \theta^i(j,G)
 7
             end
        if G \mod RepCycle == 0 then
             Realizar el proceso de reproducción ordenando la población de acuerdo a las reglas
10
             de factibilidad y eliminar a S_r peores bacterias y duplicar el resto de bacterias
             S_b - S_r.
11
12
        Realizar el proceso de eliminación-dispersión eliminando a la peor bacteria \theta^w(j,G) de
        la población actual considerando la técnica de manejo de restricciones.
13
        Actualizar el vector de tamaño de paso usando la Ec. 28
14 end
```

Windows de 64bits; y dos PC con Memoria RAM de 8.0GB, procesador de 2.4GHZ e igual sistema operativo. El algoritmo fue programado en lenguaje M en el software de desarrollo Matlab R2014a.

Dos experimentos fueron realizados, el primero consistió en calibrar los parámetros del algoritmo y en un segundo experimento se obtuvo un conjunto de soluciones para observar el rendimiento del algoritmo ante tal PONR y comparar su mejor resultado contra el obtenido por el algoritmo DE/rand/1/bin (una versión clásica del algoritmo Evolución Diferencial que emplean en CIDETEC [15]) y contra otro algoritmo evolutivo llamado C-LSHADE [20].

Primer experimento. Para la calibración de los parámetros del algoritmo se realizaron 87 ejecuciones al algoritmo con diversa combinación de parámetros y un número de generaciones de 15,000. Los rangos probados para cada parámetro fueron: S_b entre [10, 200], N_c entre [5,100], S_r entre [1, $S_b/2$], RepCycle entre [10, 200] y R, B entre [0,1]. El mejor resultado obtenido de todas las ejecuciones independientes fue el valor -564959.112.

Durante las ejecuciones del algoritmo en este primer experimento se observó que a mayor número de bacterias y ciclos quimiotáxicos, el tiempo de ejecución del algoritmo aumentaba de segundos a minutos; con la mejor combinación de parámetros encontrada el algoritmo tarda 14 minutos, en promedio, en resolver el problema con resultados factibles. Otro aspecto encontrado fue que al aumentar el número de bacterias a reproducir (RepCycle) la calidad de los resultados del algoritmo disminuía en comparación al usar valores entre 40 y 60 como máximo.

Además, los valores cercanos a cero para el tamaño de paso y factor de escalamiento, R y B, respectivamente, favorecían el rendimiento del algoritmo al generar soluciones de mayor calidad de acuerdo a la función objetivo.

Tabla 2. Detalles de la mejor solución encontrada por TS-MBFOA.

Hora	P1(Diésel)	P2(SAE)	P3(Solar)	P4(Eólica)	Carga	F. Objetivo
00:00	2496.3651	1.6701	1.0000	0.9649	2500	745.2825
01:00	1575.6793	423.8042	0.5166	500	2500	-6147.3127
02:00	2318.9864	15.5409	1.0000	514.4727	2850	-18540.63612
03:00	1620.3690	729.4394	0.1916	600	2950	-806.8983
04:00	1726.7810	122.2190	1	1000	2850	-34196.74276
05:00	2260.0548	44.1808	0.9957	194.7687	2500	-5523.7952
06:00	1592.7925	505.2331	0.5558	51.4187	2150	13392.21051
07:00	1962.2575	20.7425	266	1	2250	-35095.00219
08:00	2190.8451	38.9401	70	0.2148	2300	-7697.4964
09:00	1752.4745	240.1307	327.1400	0.2548	2320	-36970.82458
10:00	1916.7460	195.5353	237.5564	0.1623	2350	-26010.00763
11:00	2203.3221	487.8177	126.0000	132.8602	2950	-7027.6669
12:00	1179.4551	480.6729	589.8720	7.61391 E-05	2250	-65828.93738
13:00	665.9984	1094.7179	535.9387	23.3450	2320	-41244.7373
14:00	0.0081	1079.7260	700	570.2659	2350	-85010.15548
15:00	0.0038	1079.7280	560	710.2682	2350	-71276.1857
16:00	71.2158	1092.9549	405.9995	879.8298	2450	-56363.05636
17:00	755.8825	1099.9999	63.0000	1231.1175	3150	-22708.51646
18:00	915.8824	1100	0.9084	1293.2093	3310	-16576.23946
19:00	2399.4340	850.2600	0.3059	1000	4250	-12090.66619
20:00	3562.3782	187.5911	0.0307	500.0000	4250	-11849.35186
21:00	2449.0000	$2.67 ext{E-} 14$	1	550	3000	-20281.31989
22:00	2422.4054	63.8122	0.1824	463.6000	2950	-14991.08009
23:00	1474.9268	1100.0000	0.7396	74.3335	2650	30139.01468

Segundo experimento. Para observar el rendimiento del algoritmo ante el problema de optimización del Smart Grid se optó por utilizar la mejor combinación de parámetros proporcionada por el primer experimento, la cual fue: $S_b = 10$, $N_c = 8, S_r = 5, R = 0.015, B = 0.040, RepCycle = 60$ y GMAX = 15000. Se realizaron 30 ejecuciones independientes con un número de 15,000 generaciones. La mejor solución obtuvo un valor de -551960.121 como la sumatoria de las 24 funciones objetivo de la ejecución. Tal valor es negativo debido a que representa un ahorro monetario al operar con FERs, en lugar de usar únicamente el Generador Diésel (GD) y el SAE. Entre más energía suministren las FER y menos el GD y el SAE, el ahorro es mayor. Los valores de las variables de esta solución son presentados en la Tabla 2 con los cuales se realizó la gráfica de la Figura 1, en donde se observa que aunque el GD (P1) tenga asignación de carga durante todo el día, de las horas 13 a la 19 la energía generada proviene en gran parte de las FER, lo cual permite un ahorro monetario significativo en el suministro.

Al comparar el mejor resultado del TS-MBFOA contra el algoritmo C-LSHADE y DE/rand/1/bin, se puede observar en la Tabla 3 que el algoritmo basado en el forrajeo de bacterias obtiene mejor resultado para el problema de minimización del MR-RNI, sin embargo se requiere de mayor número de evaluaciones que las

21

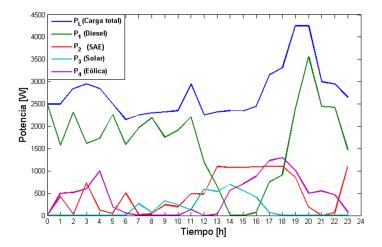


Fig. 1. Visualización del suministro de energía durante las 24 horas del día para el funcionamiento de la MR-RNI, obtenido por el TS-MBFOA

otras dos versiones de algoritmos evolutivos. Cabe mencionar que la evaluaciones son el número de veces que una solución es evaluada en la función objetivo. Para el caso de TS-MBFOA el número de evaluaciones es calculado por $S_b \times N_c \times GMAX$.

Tabla 3. Detalles de la mejor solución encontrada por TS-MBFOA.

Estadística	C-LSHADE	${f DE/rand/1/bin}$	TS-MBFOA
Mejor	$-5.33 \mathrm{E}{+05}$	$-5.32\mathrm{E}{+05}$	-5.52E + 05
Peor	$-5.32 \mathrm{E} \! + \! 05$	$-5.32\mathrm{E}{+05}$	$-3.45 \mathrm{E}{+05}$
Mediana	$-5.32\mathrm{E}\!+\!05$	$-5.32\mathrm{E}{+05}$	$-4.98 \mathrm{E}{+05}$
$\mathbf{Promedio}$	$-5.32\mathrm{E}{+05}$	$-5.32\mathrm{E}{+05}$	$-4.81\mathrm{E}{+05}$
Desv.Est.	$8.73\mathrm{E}{+01}$	$3.30 \mathrm{E}{+00}$	$4.86\mathrm{E}{+04}$
Evaluaciones	$2.88\mathrm{E}{+05}$	$2.88\mathrm{E}{+05}$	$1.20{\rm E}{+06}$

5. Conclusiones

En este trabajo un algoritmo basado en el forrajeo de las bacterias E.Coli llamado TS-MBFOA fue implementado para resolver un Problema de Optimización Numérico con Restricciones que busca minimizar un *Smart Grid*, el cual es una red inteligente de energía que hace uso de generadores distribuidos que permiten la explotación de fuentes de energía renovables como la eólica y solar así

como de combustibles, por ejemplo el diésel. El modelado matemático del Smart Grid fue propuesto por el Centro de Innovación y Desarrollo Tecnológico en Cómputo (CIDETEC) del Instituto Politécnico Nacional (IPN). En dicho modelo se calculan las potencias óptimas de una red de dispositivos de generación de energía para suministrar a una carga durante un periodo de 24 horas. En esencia, cada hora se resuelve un problema de optimización de acuerdo a las condiciones de la red y restricciones de operación de la misma. Al final se tiene una gráfica de las potencias óptimas y la suma de las 24 funciones objetivo en la cual se propone qué solución es mejor utilizar.

Dos experimentos fueron diseñados. En el primero, 87 ejecuciones independientes fueron realizadas con diferentes valores a los parámetros del algoritmo con el objetivo de obtener la mejor combinación de valores que permitiera el mejor rendimiento del algoritmo ante tal problema. Como resultado de este primer experimento se obtuvo que el mejor redimiento del algoritmo fue encontrado con una población de 10 bacterias, 8 ciclos quimiotáxicos, 5 bacterias a reproducir cada 60 generaciones con un tamaño de paso de 0.015 y un factor de escalamiento de 0.040. Además se observó que entre mayor era el número de bacterias y ciclos quimiotáxicos más tiempo de ejecución requería el algoritmo.

En un segundo experimento, el algoritmo fue ejecutado 30 veces de manera independiente usando para los parámetros los valores obtenidos en el experimento previo. Los resultados obtenidos fueron comparados contra dos algoritmos evolutivos dando como mejor algoritmo la propuesta TS-MBFOA según las estadísticas presentadas. Como lectura a la mejor solución encontrada por el algoritmo basado en el forrajeo de bacterias, se hizo uso de su gráfica la cual indicó que aunque el generador de diésel (P_1) tenga asignación de carga durante todo el día, de las horas 13 a la 19 la energía generada proviene en gran parte de las fuentes de energía renovables como la solar (P_3) y eólica (P_4) , lo cual permite un ahorro monetario significativo en el suministro.

Como trabajo futuro se buscará que TS-MBFOA disminuya el número de evaluaciones para encontrar soluciones altamente competitivas contra otros algoritmos del estado del arte, además de hacer uso de una población normalizada con el fin de disminuir el tiempo de ejecución de dicho algoritmo.

Referencias

- Bharathi, C., Rekha, D., Vijayakumar, V.: Genetic algorithm based demand side management for smart grid. Wireless Personal Communications 93(2), 481–502 (Mar 2017)
- Bordons, C., García-Torres, F., Valverde, L.: Gestión Óptima de la energía en microrredes con generación renovable. Revista Iberoamericana de Automática e Informática Industrial RIAI 12(2), 117–132 (2015)
- Bremermann, H.: Chemotaxis and optimization. J. Franklin Inst 297, 397–404 (1974)
- 4. Díaz Andrade, C., Hernández, J.: Smart grid: Las tics y la modernización de las redes de energía eléctrica estado del arte. Revista S&T 9(18), 53-81 (2011)

- Deb, K.: An Efficient Constraint Handling Method for Genetic Algorithms. Computer Methods in Applied Mechanics and Engineering 186(2-4), 311–338 (2000)
- Eiben, A., Smith, J.: Introduction to Evolutionary Computing. Natural Computing Series, Springer-Verlag (2003)
- Energía y sociedad: Smart grids (2017/03/21), http://www.energiaysociedad. es/manenergia/4-5-smart-grids/
- 8. Engelbrecht, A.: Fundamentals of Computational Swarm Intelligence. John Wiley & Sons (2005)
- Hernández-Ocaña, B., Pozos-Parra, M., Mezura-Montes, E., Portilla-Flores, E., Vega-Alvarado, E., nez, M.C.Y.: Two-swim operators in the modified bacterial foraging algorithm for the optimal synthesis of four-bar mechanisms. Computational Intelligence and Neuroscience 2016, 1–18 (2016)
- 10. Kasaiezadeh, A., Khajepour, A., Waslander, S.: Spiral bacterial foraging optimization method: Algorithm, evaluation and convergence analysis. Engineering Optimization 46(4), 439–464 (2014)
- 11. Mezura-Montes, E., Hernández-Ocaña, B.: Modified bacterial foraging optimization for engineering design. In: Proceedings of the Artificial Neural Networks in Enginnering Conference (ANNIE'2009), vol. 19, pp. 357–364. in Cihan H. Dagli et al. (editors), ASME Press Series, Intelligent Engineering Systems Through Artificial Neural Networks (2009)
- Mezura-Montes, E., Portilla-Flores, E., Hernández-Ocaña, B.: Optimization of a mechanical design problem with the modified bacterial foraging algorithm. In: Proceedings of the XVII Argentine Congress on Computer Sciences. La Plata, Argentina (October 2011)
- Mikati, M., Santos, M., Armenta, C.: Modelado y simulación de un sistema conjunto de energía solar y eólica para analizar su dependencia de la red eléctrica. Revista Iberoamericana de Automática e Informática Industrial RIAI 9(3), 267–281 (2012)
- 14. Passino, K.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine 22(3), 52–67 (2002)
- 15. Portilla-Flores, E., Mezura-Montes, E., Alvarez-Gallegos, J., Coello-Coello, C., Cruz-Villar, A., Villarreal-Cervantes, M.: Parametric reconfiguration improvement in non-iterative concurrent mechatronic design using an evolutionary-based approach. Engineering Applications of Artificial Intelligence 5(24), 757–771 (2011)
- 16. Secretaría de Energía Oaxaca, México: Evaluación rápida del uso de la energía. SENER 1(1) (2016)
- 17. Tazvinga, H., Xia, X., Zhang, J.: Minimum cost solution of photovoltaic-diesel-battery hybrid power systems for remote consumers. Solar Energy 96, 292-299 (2013)
- 18. Tazvinga, H., Zhu, B., Xia, X.: Energy dispatch strategy for photovoltaic-wind-diesel-battery hybrid power system. Solar Energy 108, 412–420 (2016)
- Wood, A., Wollenberg, B.: Power generation, operation, and control. John Wiley & Sons, NJ (2012)
- Zapata, M.Z.: Control de parámetros del algoritmo evolución diferencial con variantes combinadas para la solucián de problemas de optimización en mecatrónica (Agosto 2017)

Diseño de una aplicación Web para el seguimiento y control del sobrepeso u obesidad a través del paradigma del IoT

Uriel Ramos-Deonati¹, Giner Alor-Hernández¹, Isaac Machorro-Cano¹, José Luis Sánchez-Cervantes², Lisbeth Rodríguez-Mazahua¹

¹ Instituto Tecnológico de Orizaba, División de Estudios de Posgrado e Investigación,

Veracruz, México

² CONACYT - Instituto Tecnológico de Orizaba,

Veracruz, México

{deonatiuriel, imachorro}@gmail.com, {galor, lrodriguez}@itorizaba.edu.mx,

jlsanchez@conacyt.mx

Resumen. En la actualidad, el sobrepeso y la obesidad son dos problemas de salud globales que afectan la calidad de vida de las personas y, por lo tanto, su vida laboral. El Internet de las cosas (IoT, por sus siglas en inglés) es un nuevo paradigma que está ganando terreno rápidamente en el escenario de la tecnología inalámbrica moderna. La idea básica de este concepto es la presencia en el entorno de diversas cosas u objetos interconectados, tales como los identificadores de radiofrecuencia (RFID), sensores, actuadores, teléfonos inteligentes, entre otras cosas u objetos. En este trabajo se presenta el diseño de una aplicación Web para contribuir en el seguimiento y control del sobrepeso u obesidad, bajo el paradigma del IoT, incorporando técnicas de inteligencia artificial aplicadas al cuidado de la salud. Adicionalmente, se presenta una arquitectura de componentes y el prototipo inicial, el cual está dividido en cinco capas principales (capa de presentación, capa de integración, capa de servicios en el IoT, capa analítica de datos y capa de datos) que integran el modelo interno del sistema, así como el aspecto de la aplicación. Además, se describen las principales funcionalidades de la aplicación Web, su diseño y un análisis de distintas plataformas de aprendizaje automático para el soporte del sistema de recomendaciones médicas, considerando utilizar el algoritmo de clasificación C4.5 para analizar la información obtenida a través de los dispositivos wearables.

Palabras clave: inteligencia artificial, IoT, obesidad, RFID, sobrepeso, wearables.

Design of a Web Application for the Monitoring and Control of Overweight or Obesity through the IoT Paradigm

Abstract. Currently, overweight and obesity are two global health problems that affect the quality of life of people. The Internet of Things (IoT) is a new

paradigm that is quickly gaining acceptance on the ecosystem of modern wireless technology. The basic idea of this concept is the presence in the environment of various things or interconnected objects such as radiofrequency identifiers (RFID), sensors, actuators, smart phones, to mention but a few. This work presents the design of a Web-based application for monitoring and controlling critical parameters related to overweight or obesity by using the IoT paradigm and incorporating artificial intelligence techniques. Additionally, a layered architecture design of components and its initial prototype are presented. The architecture is divided into five main layers (presentation layer, integration layer, layer of services in the IoT, analytical data layer and data layer) that integrate the internal model of the system. In addition, the main functionalities of the Web application, the design and an analysis of different machine learning platforms for the support of the medical recommendations system are described, considering the use of the C4.5 classification algorithm to analyze the information obtained through the Wearables devices.

Keywords: artificial intelligence, IoT, obesity, overweight, RFID, wearables.

1. Introducción

La obesidad es un estado patológico caracterizado por la acumulación excesiva de grasa corporal. En 2014, más de 1,900 millones de adultos mayores de 18 años en todo el mundo tenían sobrepeso, más de 600 millones eran obesos y 41 millones de niños menores de 5 años tenían sobrepeso o eran obesos [1]. De hecho, la mayoría de la población mundial vive en países donde el sobrepeso y la obesidad causan la muerte de más personas que la insuficiencia renal. Es por esto que el sobrepeso y la obesidad son problemas que afectan la calidad de vida de las personas, particularmente en el aspecto laboral.

Por otra parte, la atención médica es un área de aplicación importante en el IoT, ya que éste está adaptado para mejorar la calidad del servicio y reducir costos. Actualmente, diversos sensores o dispositivos médicos se utilizan para controlar parámetros médicos, tales como la temperatura corporal, el nivel de glucosa en sangre y la frecuencia cardíaca, entre otros. Los recientes avances en sensores, comunicación inalámbrica y tecnologías de procesamiento son la fuerza impulsora detrás de la aplicación del IoT en los sistemas de atención médica. Últimamente, se están desarrollando sensores corporales portátiles, mejor conocidos como wearables, para monitorear continuamente las actividades o los parámetros de los pacientes en tiempo real. En este contexto, el IoT ofrece a los sistemas de salud una interconexión de los diversos dispositivos heterogéneos para obtener información rápida, completa y precisa de los parámetros de salud de un paciente [2].

Asimismo, los entornos de vida asistida facilitan la vida diaria de las personas con discapacidades y condiciones médicas crónicas. Gracias a la capacidad de los sensores para administrar datos de manera eficiente, es posible proporcionar a los pacientes servicios de vida asistida en tiempo real.

Además, el uso del IoT en el cuidado de la salud contribuye a servicios innovadores, como la recopilación de datos vitales del paciente a través de una red de

sensores conectados a dispositivos médicos. También, contribuye a la entrega de datos a la nube de un centro médico para su almacenamiento y procesamiento, y garantiza el acceso ubicuo o el intercambio de datos médicos (por ejemplo, registros de salud) [3]. Del mismo modo, los desafíos en el sector salud son una oportunidad para desarrollar e implementar el IoT y así contribuir a la mejora de los servicios.

En este trabajo se propone el diseño de una aplicación Web para contribuir en el seguimiento y control del sobrepeso u obesidad bajo el paradigma del IoT, incorporando técnicas de inteligencia Artificial aplicadas al cuidado de la salud, de tal manera que motive a la participación ciudadana y la responsabilidad del autocuidado. Al maximizar la participación ciudadana en el autocuidado se reduce el número de pacientes que padecen sobrepeso u obesidad y, por lo tanto, minimiza el costo social generado para prevenir y atender a los pacientes que padecen estas enfermedades.

Este trabajo está estructurado de la siguiente manera: la sección 2 presenta los trabajos relacionados con el sobrepeso u obesidad y el IoT; la sección 3 explica la metodología de la investigación utilizada para el desarrollo de la aplicación Web basada en IoT; la sección 4 presenta las funcionalidades del prototipo; finalmente, la sección 5 presenta las conclusiones de la investigación y el trabajo futuro.

2. Trabajos relacionados

El uso de dispositivos basados en el IoT está empezando a transformar el estilo de vida de millones de personas y particularmente en las diversas actividades del cuidado de la salud. En este sentido, el IoT está aportando grandes oportunidades para la monitorización, análisis, diagnóstico y recomendaciones de enfermedades como el sobrepeso u obesidad; es por ello que muchos investigadores focalizan sus investigaciones en este tópico. A continuación, se presenta una revisión del estado del arte de los trabajos que usan tecnologías Web e IoT en el cuidado de la salud, particularmente en relación al sobrepeso u obesidad.

Vilallonga *et al.* [4] presentaron un estudio de seguimiento de pacientes obesos después de la cirugía. Los pacientes consideraron el ahorro de tiempo como muy valioso y concluyeron que visualizar sus gráficos de progreso de manera constante fue muy motivador, esto evita que el cirujano pierda tiempo, esfuerzo y personal en el análisis rutinario de los pacientes, en prácticas como medición de peso, tejido graso, grasa magra e índice de masa corporal, los cuales típicamente se obtienen por medio de la báscula; esta información el paciente la envía desde su hogar y se logra dar un seguimiento de forma cómoda y oportuna obteniendo resultados favorables.

A su vez, Zaragozá *et al.* [5] describieron una plataforma de monitoreo destinada a establecer una red de sensores para niños obesos bajo tratamiento clínico; además, propuso el sistema ETIOBE, el cual se desarrolló para mejorar el tratamiento, promover los mecanismos de autocontrol y prevenir recaídas en niños obesos con tratamiento clínico. ETIOBE es la plataforma de monitoreo ubicua, ya que permite al clínico obtener información relevante de los pacientes (contextual, fisiológica y psicológica), que permite la adaptación del tratamiento, según la evolución del paciente.

Vazquez *et al.* [6] presentaron una arquitectura de salud móvil destinada a prevenir la obesidad infantil mediante la promoción de buenos comportamientos de salud, con un conjunto de aplicaciones mHealth. Las aplicaciones proporcionaron notificaciones y mensajes a los de adultos para mejorar los resultados.

Por otro lado, Kim *et al.* [7] presentaron el diseño y uso de iN Touch, una aplicación de autogestión móvil para el seguimiento de las observaciones de la vida diaria (ODL) en un programa de entrenamiento en salud para jóvenes de bajos recursos, urbanos y de minorías, con sobrepeso u obesidad.

Además, Alloghani *et al.* [8] propuso una aplicación de salud móvil con el propósito de aumentar los niveles de conciencia de los padres y los niños sobre los riesgos de la obesidad y ayudarlos a mantener un estilo de vida, de alimentación equilibrado y saludable.

Wibisono y Astawa [9] presentaron un sitio Web y aplicaciones móviles para un programa de pérdida de peso con tecnología de máquina a máquina (M2M), utilizando una escala de peso especial para cargar datos al servidor. El sitio Web o la aplicación de Android contiene básicamente cuatro páginas principales: 1) Home. Muestra el inicio de la aplicación; 2) Historial. Contiene estadísticas del peso del usuario para un rango de tiempo específico: los últimos siete días, el último mes, últimos tres meses o todo el programa; 3) Calorías de Alimentos. Muestra varios tipos de alimentos (indonesio y occidental) y las calorías contenidas, y 4) Información personal. Muestra información del usuario como nombre de usuario, nombre, apellido, correo electrónico, contraseña, sexo, altura, peso inicial, objetivo de peso, nivel de actividad diaria y la imagen del usuario. Esta información se completa previamente con el registro del usuario y se actualiza a través de esta página.

Por otra parte, Aupetit *et al.* [10] describieron el diseño de un tablero para la visualización de datos biométricos de un campamento de obesidad infantil, en Qatar. Este tablero evaluó el estado de salud de un sujeto frente al grupo, comparándolo con otro. Un experto en salud también proporcionó al sujeto un estado objetivo (por ejemplo, peso) para ser comparado, junto con recomendaciones sobre las actividades a mejorar.

Adicionalmente, Yang *et al.* [11] desarrollaron un estudio para evaluar la efectividad de la prevención de la obesidad en niños de 10 a 12 años de edad, con un sistema de plataforma móvil llamado HAPPY ME, que es una aplicación de teléfono inteligente junto con un dispositivo portátil, diseñado para mejorar los comportamientos saludables para prevenir la obesidad infantil.

Además, Laing *et al.* [12] realizaron un estudio de la efectividad de una intervención experimental basada en recomendaciones de dieta por medio de una aplicación inteligente para la pérdida de peso en pacientes mayores de 18 años con sobrepeso u obesidad, concluyendo que las aplicaciones para bajar de peso son útiles para las personas que están listas para controlar las calorías. Sin embargo, para el paciente con sobrepeso promedio, es improbable que la introducción de una aplicación de teléfono inteligente produzca un cambio de peso significativo.

De igual forma, Otto *et al.* [13] presentaron una arquitectura de hardware y software de un sistema de red de sensores inalámbricos para el monitoreo ambulatorio del estado de salud. El sistema consiste en múltiples nodos sensores que monitorean

el cuerpo, movimiento y actividad del corazón, utilizando un coordinador de red y un servidor personal; además, usaron una arquitectura del software que se basa en TinyOS, un sistema operativo de código abierto ampliamente utilizado para redes de sensores.

Ahmed *et al.* [14] presentaron una visión general de los sistemas existentes de monitoreo de la salud, teniendo en cuenta el enfoque del IoT, además de las tendencias recientes y el desarrollo de sistemas de monitoreo de la salud en términos de parámetros y marcos de salud, comunicación inalámbrica y problemas de seguridad, así como identificar limitaciones y ventajas.

Finalmente, Shin [15] definió el nuevo concepto de IoT-learning integrando las tecnologías IoT y el aprendizaje omnipresente, y propuso un programa personalizado de entrenamiento utilizando IoT-learning para el control de peso, el cual proporciona a los usuarios no sólo información personalizada, transmitiendo datos a través de la red y recibiendo la información para analizarla, sino también para proporcionar a los usuarios datos que coinciden con la información personalizada. IoT-learning proporciona a los usuarios (estudiantes) retroalimentación mediante la recopilación de información a través de sensores. El objetivo más importante de este aprendizaje es brindar capacitación personalizada a los usuarios con los datos obtenidos a través de la tecnología de sensores en el IoT. Este documento combina la tecnología de detección en el IoT y los sistemas de aprendizaje.

Como se muestra en la Tabla 1 (integrada por los elementos: a. wearable, b. conectividad, c. interoperabilidad, d. integración, e. parámetros médicos, y f. recomendación médica), los trabajos [5,6,7,13,14] cuentan con la inclusión de parámetros médicos, pero no proporcionan recomendaciones médicas.

Artículo	a	b	c	d	e	f
Vilallonga et al. [4]	No	Sí	Sí	No	Sí	Sí
Zaragozá et al. [5]	Sí	Sí	Sí	Sí	Sí	No
Vazquez et al. [6]	No	Sí	Sí	Sí	Sí	No
Kim <i>et al</i> . [7]	No	Sí	Sí	No	Sí	No
Alloghani et al. [8]	Sí	Sí	Sí	Sí	No	Sí
Wibisono y Astawa [9]	No	Sí	Sí	Sí	Sí	Sí
Aupetit et al. [10]	Sí	Sí	Sí	No	No	Sí
Yang et al. [11]	Sí	Sí	Sí	No	Sí	No
Laing et al. [12]	No	No	Sí	No	Sí	Sí
Otto et al. [13]	Sí	Sí	No	Sí	Sí	No
Ahmed et al. [14]	No	No	No	No	Sí	No
Shin [15]	Sí	Sí	No	No	Sí	Sí

Tabla 1. Comparativa de trabajos realizados.

Por otra parte, en [8,10] sí cuentan con recomendaciones, pero no integran parámetros médicos; sin embrago, [9] cuenta con todos los elementos, pero no usa algún dispositivo vestible para la recolección de información ya que se realiza de

forma manual, en relación con [14] que no tiene una integración ni realiza recomendaciones médicas. Además, [15] cumple con gran parte de los elementos, sin embargo, no contiene buena interoperabilidad ni integración.

Ante este contexto, al realizar un análisis comparativo se observa que el diseño de la aplicación Web para contribuir en el seguimiento y control del sobrepeso u obesidad bajo el paradigma del IoT propuesto en este trabajo, considera todos los aspectos analizados en la Tabla 1. En la siguiente sección se presenta la metodología de la investigación, la cual está integrada por módulos y capas que se utilizaron para el diseño de la aplicación Web relacioanda con el seguimiento y control del sobrepeso u obesidad bajo el paradigma del IoT.

3. Metodología de la investigación

La propuesta del diseño de una aplicación Web para el seguimiento y control del sobrepeso u obesidad aborda un tema prioritario sobre una problemática de salud pública, como lo es el sobrepeso u obesidad, el cual se pretende resolver mediante la conectividad, interoperabilidad e integración de dispositivos en el IoT, e incorporando técnicas de inteligencia artificial, aplicadas al cuidado de la salud. Por ello, la metodología planteada se divide en tres módulos, los cuales se muestran en la Fig. 1.

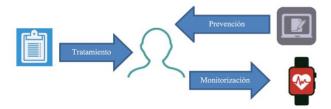


Fig. 1. Metodología de la investigación.

Monitorización. Permitirá medir a través de wearables los signos vitales de las personas con sobrepeso u obesidad para identificar variables críticas (calorías quemadas, sueño, ritmo cardíaco, Índice de Masa Corporal – IMC, número de pisos subidos, calidad del sueño, calorías consumidas, ejercicios, control de peso) y utilizarlas como parámetros para predecir problemas o patologías que afecten la salud de una persona, por medio de técnicas de inteligencia artificial. Además, en este módulo se pretende identificar patrones que mejoren o afecten el sobrepeso u obesidad en relación a la frecuencia cardíaca, ingesta de alimentos, calorías quemadas, consumo de agua, calidad del sueño, entre otras variables, con el propósito de obtener la información necesaria para la generación de recomendaciones y mensajes de motivación.

Prevención. En este módulo se consideran recomendaciones médicas que conlleven a acciones, procedimientos e intervenciones integrales que mantengan o mejoren la salud de las personas con sobrepeso u obesidad a través de la medición de parámetros y la identificación de variables críticas. Por ello, se planea mostrar recomendaciones en los casos que lo requieran; por ejemplo, si la frecuencia cardíaca

es más alta o baja de acuerdo con los valores normales, por el exceso o la falta de ejercicio, o si en la ingesta de alimentos disminuye el consumo de proteínas o se incrementa el consumo de grasas o carbohidratos.

Tratamiento. Consiste en apoyar a los usuarios con los cuidados paliativos necesarios para tratar los síntomas y los efectos secundarios causados por el sobrepeso u obesidad, además se pretende dar seguimiento a las recomendaciones realizadas para monitorizar la evolución e identificar qué recomendación es la que proporciona mejores resultados.

Para el diseño de los módulos, se presenta un diagrama UML (Unified Modeling Language) de clases, el cual muestra de forma detallada la estructura que contendrá el sistema, el cual cuenta con un motor de descargas que se encargará de obtener la información correspondiente de las API (Application Programming Interface) requeridas sin afectar el rendimiento de la aplicación, además de formular las consultas necesarias para brindar información de interés para los usuarios, médicos y a los servicios de salud que lo integren. Esta estructura como se observa en la Fig. 2, está pensada en un modelo orientado a objetos con 5 capas principales:

- 1. Capa de Presentación: Esta capa mantiene comunicación directa con el usuario, y contiene las siguientes clases:
 - Corazón: clase específica para mantener la información correspondiente a los datos del corazón, como son número de latidos por minuto, frecuencia cardíaca, entre otros.
 - Sueño: mantiene los elementos necesarios para monitorizar el tiempo y profundidad del sueño.
 - Dispositivo: mantiene la estructura para el control de los dispositivos del usuario.
 - Pasos: contiene el control de pasos y pisos subidos.
 - Líquido: mantiene estructura necesaria para la manipulación del líquido registrado, así como la cantidad.
 - Usuario: ejemplifica la estructura de almacenamiento del usuario.
 - Ejercicio: corresponde a la administración del ejercicio realizado.
 - Ingesta: realizado para mantener y controlar la información de ingesta diaria.
- 2. Capa de Integración. Esta capa está integrada por el proveedor del wearable o sensor, el componente selector de consulta y el formador de respuesta. Todos ellos colaboran para recibir datos y consultas del usuario y formular las respuestas correspondientes mediante la siguiente clase:
 - Motor: encargado de controlar las consultas, conectores a servicios y ejecución de consultas, procesamiento de información y organización, esta capa preprocesa los datos para su fácil entendimiento y consumo.
- 3. Capa de Servicios en el IoT: Esta capa es responsable de vincular, invocar, seleccionar y confirmar los servicios en IoT, contiene la siguiente clase:
 - Servicio: encargada de vincular y ejecutar los servicios médicos correspondientes.
- 4. Capa Analítica de Datos. Esta capa incluye el servicio de identificación de variables críticas y el sistema de recomendación médica.

 Capa de Datos: Contiene todos los datos útiles para la Plataforma Inteligente de Salud y está integrado por datos o eventos en tiempo real, variables críticas y recomendaciones médicas.

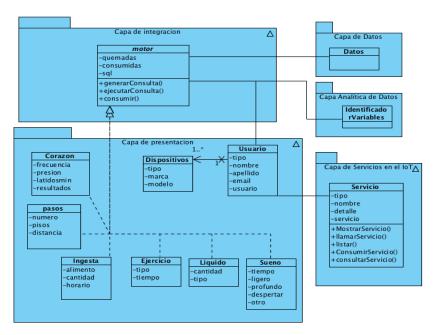


Fig. 2. Diagrama UML de clases del diseño de la arquitectura.

En la Fig. 3 se presenta un diagrama UML de paquetes, que representa la interacción de los paquetes que contienen los algoritmos de IA relacionados con el motor de procesamiento de la aplicación Web para el seguimiento y control del sobrepeso u obesidad.

Asimismo, en el diagrama UML de paquetes se presentan dos elementos que interactúan entre sí: el motor de procesamiento y los servicios en el IoT. Por otra parte, el paquete motor de procesamiento tiene conectividad con el conector de base de datos para obtener la información almacenada y con el procesador de información que se encarga de procesar los datos obtenidos por el conector y prepararlos de forma adecuada en el formato correcto, para enviarlos a los algoritmos de clasificación y recomendación.

Adicionalmente, el paquete Servicios en el IoT contiene el API de inteligencia artificial que almacena los algoritmos de clasificación y recomendación a utilizar en la aplicación Web. Los algoritmos de clasificación que se consideran utilizar son los algoritmos Naive Bayes (utilizados en la exploración inicial de los datos, buscan correlaciones entre atributos y son de entrenamiento rápido), el algoritmo C4.5 (procesa la información de una base de datos y construye diagramas lógicos) y aprendizaje profundo (emula el enfoque de aprendizaje que los seres humanos utilizan para obtener ciertos tipos de conocimiento, se considera como una forma de automatizar el análisis predictivo).

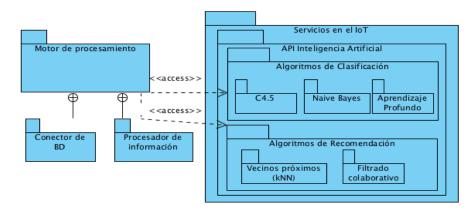


Fig. 3. Diagrama UML de paquetes que describe la interacción de los algoritmos de IA con la aplicación Web.

Tabla 2. Principales funciones de la aplicación Web relacionada con el sobrepeso u obesidad.

Funcionalidad	Descripción
¿Cuántos pasos dados por día?	Se obtiene la distancia recorrida y con base en eso las calorías quemadas.
¿Cuántos pisos subidos por día?	Por medio de esta información se obtienen las calorías quemadas.
¿En promedio cuantas horas se duerme por día en una semana?	Se logra obtener el promedio de sueño y relacionarlo con el ritmo cardíaco y otras variables.
¿Cuántos litros de agua natural se consume por día?	Con base en esta información se logra saber los niveles de hidratación del cuerpo.
¿En promedio cuál es el ritmo cardíaco por día?	Logra mantener monitoreado el corazón y saber si existe un incremento excesivo o disminución.
¿Cuántas calorías se ingieren por día?	Es necesario ya que es la base para mantener un control de sobrepeso u obesidad.
¿Cuántas calorías se consumieron hasta el día de hoy?	Esta información es necesaria ya que se logra obtener un registro total de cuántas calorías se consumieron y así poder dar una mejor recomendación.
¿Cuántas calorías se consumen al mes?	Necesario para mantener un control de sobrepeso.
¿En promedio cuántas calorías se consumen por día en una semana?	Es necesario ya que es la base para lograr una recomendación con base en calorías consumidas.
¿Cuál es el tipo de ejercicio realizado con mayor frecuencia?	Dependiendo del ejercicio se logra saber cuántas calorías se queman y da retroalimentación sobre gustos o rutinas.
¿En promedio cuántas calorías se queman por cada día de una semana?	Indica de forma detallada cuántas calorías son quemadas durante cada día de la semana sin tomar en cuenta la actividad a la que se relaciona.
¿Cuántas calorías se ingieren por mes?	Gracias a esta información se sabe cuáles son los meses en los que se consumen más calorías.

Funcionalidad	Descripción
¿Cuántos días de la semana se encuentra por debajo de las calorías consumidas en relación con las calorías quemadas?	Es importante obtener esta información ya que al contrario de las anteriores se sabe si se están quemando más calorías de las necesarias.
¿Qué día de la semana se consumieron menos calorías?	Esta información es importante ya que da una idea de qué día y en relación a que se consumieron menos calorías.
¿Qué ejercicio produce quemar más calorías?	Dependiendo el ejercicio que se realice da un indicador de qué ejercicio de los que se realizan beneficia más.

Por otra parte, los algoritmos de recomendación que se consideran utilizar son: el algoritmo de vecinos próximos KNN (utiliza los datos recolectados para calcular la similitud entre los elementos comunes, es sencillo, eficaz y funciona realizando la búsqueda de elementos con patrones de evaluación similares), y los algoritmos de filtrado colaborativo (mejoran el rendimiento en relación a la predicción utilizando las evaluaciones de los elementos afines para indicar una recomendación). Estos algoritmos se pretenden utilizar para la identificación de variables criticas (calorías quemadas, sueño, ritmo cardíaco, IMC, número de pisos subidos, calidad del sueño, calorías consumidas, ejercicios, control de peso) y la generación de recomendaciones.

Además, en la Tabla 2 se presentan las 15 principales funciones de la aplicación Web propuestas, relacionadas con el sobrepeso u obesidad. Los datos se presentan en relación con la estructura de la pregunta de interés a resolver que cumple el sistema y una descripción de la funcionalidad que debe realizar.

En la siguiente sección se presentan las funcionalidades de la aplicación Web para el seguimiento y control del sobrepeso u obesidad bajo el enfoque del IoT.

4. Prototipo de la aplicación Web

La aplicación Web propuesta proporciona un ambiente gráfico amigable que tiene el objetivo de contribuir al control del sobrepeso u obesidad y la motivación del usuario, además de mostrar datos estadísticos de la información obtenida, este diseño flexible le proporciona herramientas de registro de ingesta; asimismo, está basado en componentes para la Web y dispositivos móviles. A continuación, se presentan algunos elementos que forman parte de la propuesta.

4.1. Visualización de la monitorización del sobrepeso u obesidad

En la Fig. 4 se presenta la vista realizada para el apartado de monitorización, el cual se encarga de presentar datos estadísticos y diarios de un usuario, mediante 5 gráficas que presentan la información en tiempo real.

Esta vista muestra, a través de mosaicos, los datos diarios relacionados con el peso, la frecuencia de actividad física, las calorías quemadas, las calorías por consumir y el IMC, representados en gráficas que muestren la información en tiempo real. Además,

se presentan cuatro entradas, las cuales sirven como filtros combinables, que brindan la posibilidad de obtener detalles estadísticos, por semana, por mes o incluso por año.



Fig. 4. Diseño de monitorización del sobrepeso y obesidad.

Además, al ingresar a la información de cada variable (peso, la frecuencia de actividad física, las calorías quemadas, las calorías por consumir y el IMC) se plantea visualizar la monitorización de los datos de cada variable para comparar el progreso por medio de una vista retrospectiva, con el propósito de identificar las variables en las que se va mejorando, en las que se mantienen los datos o en las que es necesario mejorar, para motivar e incentivar a obtener el indicador de IMC en la clasificación de normopeso.

4.2. Sincronización de wearables

En el diseño de la interfaz del menú "Sincronizar dispositivo" de la pestaña "Mi perfil" se utiliza un botón desplegable para seleccionar el tipo de dispositivo (Smartband, Smartwatch, Sensor, Otro); un botón desplegable para seleccionar la marca del dispositivo (Fitbit, Garmin, Polar, Samsung, Motorola, Apple, entre otros), y un botón desplegable para que, de acuerdo a la marca del dispositivo, se seleccione el modelo que le corresponda. En la Fig. 5 se presenta el diseño del apartado de la sincronización de wearables.

Adicionalmente, el apartado de sincronización de wearables muestra tres listas, las cuales se cargan en función de la opción seleccionada; primero, se debe seleccionar el tipo de dispositivo a sincronizar, seguido de la marca, la cual despliega una lista de marcas soportadas, y con base en la marca seleccionada se recarga la lista de modelos, actualizando la lista de modelos disponibles para esa marca.

Asimismo, en la parte inferior se contempla el botón "Guardar y sincronizar", para respaldar los datos seleccionados y conjuntamente sincronizar la cuenta del usuario con el o los dispositivos que realizarán el monitoreo.

4.3. Recomendaciones

La solución propuesta para el apartado de recomendaciones se presenta en la Fig. 6 y cuenta con la sección: Sobrepeso u obesidad. En esta subsección se muestran las

recomendaciones de los signos vitales relacionados con el sobrepeso u obesidad (peso, calorías consumidas, calorías quemadas, actividad física, IMC).

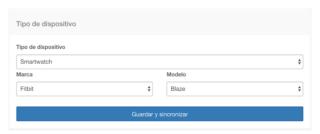


Fig. 5. Diseño del apartado de Sincronización de wearables.

También, en este apartado se presenta una serie de recomendaciones para el usuario, obtenidas con base en los datos almacenados utilizando algoritmos de inteligencia artificial, tales como el algoritmo de clasificación C4.5, el cual es un modelo de predicción y una extensión del algoritmo ID3, que soluciona algunos de sus inconvenientes. Asimismo, el algoritmo C4.5 procesa la información de una base de datos y construye diagramas lógicos, similares a los que se obtienen en sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que suceden de forma sucesiva, para la resolución de un problema. Este tipo de modelos se utilizan para predecir el valor de un atributo categórico (discreto o nominal) y su aplicación se realiza en diagnósticos médicos.

Adicionalmente, se plantea utilizar alternativas de software de aprendizaje automático y minería de datos que ofrecen algunas API de integración (Weka [16] y Apache Mahout [17]), las cuales se planea integrarlas al sistema con la finalidad de realizar recomendaciones por medio del análisis de la información almacenada.

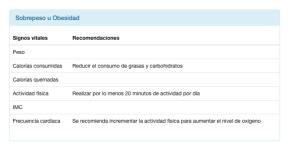


Fig. 6. Vista del apartado recomendaciones

Además, al realizar un análisis comparativo de las API se identificó que uno de los aspectos relevantes en las plataformas es el tipo de licencia, ya que todas tienen un API de desarrollo, algunos ofrecen versiones comunitarias incompletas, como son Rapid miner [18] y KNIME [19], pero en otros casos existen versiones libres como Weka [16] y Apache Mahout [17].

Por otro lado, el procesamiento de información se pretende realizar utilizando los datos históricos obtenidos por los wearables. Una vez almacenada la información en la base de datos la aplicación Web envía los datos por medio de un servicio

RESTFUL para el análisis y obtención de recomendaciones. Además, se plantea que las recomendaciones realizadas estén validadas por especialistas en el área de la salud.

En la siguiente sección se presentan las conclusiones a las que se llegaron al realizar el diseño de la aplicación Web para el seguimiento y control del sobrepeso u obesidad bajo el paradigma del IoT. Además, se plantea el trabajo a futuro y los agradecimientos.

5. Conclusiones y trabajo a futuro

El sobrepeso y la obesidad son problemas de salud transversales que afectan la calidad de vida de las personas, incluida su vida laboral. En este sentido, el IoT brinda grandes oportunidades para monitorear, analizar, diagnosticar, controlar y brindar recomendaciones de tratamiento para enfermedades crónico-degenerativas, como el sobrepeso u obesidad.

En este trabajo se propone el diseño de una aplicación Web que permita el seguimiento y, posteriormente, el control del sobrepeso u obesidad bajo el enfoque del IoT, incorporando técnicas de inteligencia artificial, aplicadas al cuidado de la salud. De manera similar, se discutió cómo, a través de varios trabajos de investigación, el IoT contribuye al diagnóstico y tratamiento de enfermedades como el sobrepeso u obesidad.

Como trabajo futuro se realizará un análisis comparativo de los trabajos más relevantes que abordan el papel del IoT en la prevención, el control, el tratamiento y la lucha contra el sobrepeso u obesidad. Además, se busca validar la aplicación Web para el seguimiento y control del sobrepeso u obesidad bajo el paradigma del IoT, en un caso de estudio de atención médica y, posteriormente, desarrollar la aplicación Web para el seguimiento y tratamiento del sobrepeso u obesidad de un paciente basado en el IoT.

Agradecimientos. Los autores agradecen al Consejo Nacional de Ciencia y Tecnología (CONACYT), al Tecnológico Nacional de México (TecNM), al Programa para el Desarrollo Profesional Docente (PRODEP) y a la Universidad del Papaloapan (UNPA) por el apoyo otorgado para la realización de esta investigación.

Referencias

- 1. Obesity and overweight, http://www.who.int/mediacentre/factsheets/fs311/en/ (2018)
- Ling, Li, Shancang, Li, Shanshan, Z.: QoS-Aware Scheduling of Services-Oriented Internet of Things. IEEE Transactions on Industrial Informatics 10(2), pp. 1497–1505 (2014)
- 3. Xu, L., He, W., Li, S.: Internet of Things in Industries: A Survey. IEEE Transactions on Industrial Informatics 10(4), pp. 2233–2243 (2014)
- Vilallonga, R., Lecube, A., Fort, J., Boleko, M., Hidalgo, M., Armengol, M.: Internet of Things and bariatric surgery follow-up: Comparative study of standard and IoT follow-up. Minimally Invasive Therapy & Allied Technologies 22(5), pp. 304–311 (2013)

- Zaragozá, I., Guixeres, J., Alcañiz, M., Cebolla, A., Saiz, J., Álvarez, J.: Ubiquitous monitoring and assessment of childhood obesity. Personal and Ubiquitous Computing 17(6), pp. 1147–1157 (2012)
- Vazquez, M., Jimenez, E., Nieto, J., Sanchez, J., Garcia, A., Torres, J.: Development of a Mobile Health Architecture to Prevent Childhood Obesity. IEEE Latin America Transactions 13(5), pp. 1520–1527 (2015)
- Kim, K., Logan, H., Young, E., Sabee, C.: Youth-centered design and usage results of the iN Touch mobile self-management program for overweight/obesity. Personal and Ubiquitous Computing 19(1), pp. 59–68 (2014)
- Alloghani, M., Hussain, A., Al-Jumeily, D., Fergus, P., Abuelma'atti, O., Hamden, H.: A
 mobile health monitoring application for obesity management and control using the
 internet-of-things. In: Sixth International Conference on Digital Information Processing
 and Communications (ICDIPC), pp. 19–24 (2016)
- Wibisono, G., Astawa, I.: Designing Machine-to-Machine (M2M) Prototype System for Weight Loss Program for Obesity and Overweight Patients. In: International Conference on Intelligent Systems, Modelling and Simulation (ISMS), pp. 1197–209 (2016)
- Aupetit, M., Fernandez-Luque, L., Singh, M., Srivastava, J.: Visualization of Wearable Data and Biometrics for Analysis and Recommendations in Childhood Obesity. In: International Symposium on Computer-Based Medical Systems (CBMS), pp. 678–679 (2017)
- Yang, H., Kang, J., Kim, O., Choi, M., Oh, M., Nam, J., Sung, E.: Interventions for Preventing Childhood Obesity with Smartphones and Wearable Device: A Protocol for a Non-Randomized Controlled Trial. International Journal of Environmental Research and Public Health 14(2), p. 184 (2017)
- 12. Laing, B., Mangione, C., Tseng, C., Leng, M., Vaisberg, E., Mahida, M., Bholat, M., Glazier, E., Morisky, D., Bell, D.: Effectiveness of a Smartphone Application for Weight Loss Compared with Usual Care in Overweight Primary Care Patients. Annals of Internal Medicine 161, pp. S5–S12 (2014)
- 13. Otto, C., Milenković, A., Sanders, C., Jovanov, E.: System Architecture of a Wireless Body Area Sensor Network for Ubiquitous Health Monitoring: Journal of Mobile Multimedia, vol. 1, pp 307–326 (2006)
- 14. Ahmed, M., Björkman, M., Čaušević, A., Fotouhi, H., Lindén, M.: An Overview on the Internet of Things for Health Monitoring Systems. Internet of Things. IoT Infrastructures. pp. 429–436 (2016)
- 15. Shin, S.A., Lee, N.Y., Park, J.H.: Empirical study of the IoT-learning for obese patients that require personal training: Advances in Computer Science and Ubiquitous Computing. LNEE, vol. 421, pp. 1005–1012. Springer, Singapore (2017)
- 16. Weka 3 Data Mining with Open Source Machine Learning Software in Java (2018)
- 17. Apache Mahout (2018)
- 18. Data Science Platform | RapidMiner (2018)
- 19. KNIME Open for Innovation (2018)

Hacia transacciones distribuidas coordinadas por agentes para la arquitectura de microservicios

Xavier Limón¹, Alejandro Guerra-Hernández², Angel J. Sánchez-García¹, Juan Carlos Peréz Arriaga¹, Juan Luis López Herrera¹

¹ Universidad Veracruzana, Facultad de Estadística e Informática, Xalapa, México
² Universidad Veracruzana, Centro de Investigación en Inteligencia Artificial, Xalapa, México

{hlimon, aguerra, angesanchez, juaperez}@uv.mx, jlopezh73@gmail.com

Resumen. La arquitectura de microservicios ha surgido como respuesta al rápido cambio tecnológico, la preocupación de mayor extensibilidad, escalabilidad y a la necesidad de ciclos de entrega de software cada vez más cortos. En esta arquitectura, el sistema, usualmente distribuido, es descompuesto en una serie de servicios altamente cohesivos e independientes. Cada microservicio puede contar con tecnologías de implementación y persistencia de datos diferente, obteniéndose así sistemas distribuidos heterogéneos, donde cada microservicio puede ser desarrollado y mantenido por equipos de trabajo diferentes. Dada su heterogeneidad y naturaleza distribuida, uno de los retos latentes en esta arquitectura es el manejo adecuado de transacciones distribuidas que se expanden por varios microservicios. En este trabajo se propone una aproximación basada en agentes para la arquitectura de microservicios, dicha aproximación puede verse como una capa autónoma que coordina las transacciones distribuidas del sistema. A diferencia de aproximaciones existentes, la propuesta es de alto nivel, lo que facilita su futura implementación en una tecnología de agentes específica, e integra aspectos de confiabilidad y robustez basados en coordinación entre agentes.

Palabras clave: sistemas multi-agente, microservicios, arquitecturas de software, transacciones distribuidas.

Toward Distributed Transactions Coordinated by Agents for the Microservices Architecture

Abstract. The microservices architecture has arisen as a response to fast technological changes, more extensibility and scalability concerns, and the necessity of shorter software delivery cycles. In this architecture, the system, usually distributed, is decomposed into a series of highly cohesive and independent services. Each microservice can have different implementation and data persistence technologies, thus obtaining heterogeneous distributed systems, where each microservice can be developed and maintained by different work teams. Given its heterogeneity and distributed nature, one of the latent challenges in this architecture is

the proper management of distributed transactions that are expanded by several microservices. In this paper we propose an agent-based approach for the microservices architecture, this approach can be seen as an autonomous layer that coordinates the distributed transactions of the system. Unlike existing approaches, the proposal is of a high level, which facilitates its future implementation in a specific agent technology, and integrates aspects of reliability and robustness based on coordination between agents.

Keywords: multi-agent systems, microservices, software architectures, distributed transactions.

1. Introducción

Los microservicios son un estilo arquitectónico de software inspirado en el cómputo orientado a servicios que en tiempos recientes ha gozado de gran popularidad [4]. La idea general de esta arquitectura es descomponer el sistema en servicios independientes, usualmente distribuidos, que hacen la menor cantidad posible de tareas relacionadas, esto es, servicios pequeños con alta cohesión que eventualmente colaboran entre si. Al ser independientes, cada microservicio cuenta con su propio ciclo de vida, lo que permite que sean desarrollados por diversos grupos de trabajo en momentos diferentes y utilizando diferentes tecnologías heterogéneas [11]. Todas las características antes mencionadas facilitan la integración e incluso experimentación de nuevas tecnologías, así como la extensibilidad y escalabilidad misma del sistema.

Los microservicios nacen en un momento donde existen diversas formas para lidiar con problemas de interoperabilidad derivados de componentes heterogéneos, siendo el ejemplo más relevante los servicios web RESTful [6], los cuales promueven el uso de formatos neutrales estándar, principalmente JSON, para el intercambio de datos. La combinación entre servicios web RESTful y microservicios es una de las más comunes en la actualidad [11] ya que brinda facilidad de distribución, flexibilidad de despliegue, tolerancia a fallos y velocidad de entrega.

Considerando su naturaleza independiente, abierta y heterogénea, los micro servicios suelen seguir un patrón de diseño conocido como base de datos por servicio [14], el cual establece que cada microservicio cuente con su propia base de datos, lo que puede incluir diversas tecnologías de persistencia SQL o no SQL.

Dada la potencial diversidad de tecnologías de persistencia de datos en microservicios, un problema abierto es el manejo de transacciones distribuidas que involucran a varios microservicios. Una transacción básicamente es un conjunto de operaciones que se deben de ejecutar de forma atómica, esto es, o se ejecutan todas o no se ejecuta ninguna, de esta forma se asegura la integridad de datos si algo sale mal en cualquier paso. La coordinación necesaria para realizar cada paso de la transacción, y potencialmente limpiar la transacción en caso de error, mientras al mismo tiempo se trata de no degradar significativamente el

rendimiento y escalabilidad del sistema, es un problema no trivial que requiere por si mismo de tecnologías y patrones de diseño propios.

Los sistemas Multi-Agente, MAS por su siglas en inglés, facilitan la realización de tareas coordinadas complejas en sistemas distribuidos, ya que son, por definición, sistemas distribuidos y cuentan con el nivel de abstracción adecuado para facilitar la comunicación entre componentes, así como la flexibilidad, modularidad, robustez, autonomía y escalabilidad necesarios [8].

Dadas las características mencionadas, es natural pensar en la posibilidad de unir los sistemas Multi-Agente con la arquitectura de microservicios. Está unión, aunque en la actualidad no muy común, puede brindar posibilidades interesantes como la que se presenta en este trabajo, donde se propone una aproximación basada en sistemas Multi-Agente para resolver el problema de transacciones distribuidas en la arquitectura de microservicios. Los objetivos principales de la aproximación propuesta son los siguientes:

- 1. Escalabilidad. Apropiada para sistemas de gran escala, con diversos nodos computacionales y una gran cantidad de microservicios.
- Confiabilidad. Es posible adoptar estrategias para mantener la integridad de datos incluso en situaciones de fallo.
- Heterogeneidad. Abierta a cualquier tipo de tecnología de implementación y persistencia de datos en los microservicios.
- 4. Alto nivel. El modelo de la propuesta, su flujo de trabajo, sus conceptos y estrategias asociadas pueden ser entendidas en términos de alto nivel gracias al uso de agentes. Este aspecto facilita el uso y configuración por parte de los desarrolladores de microservicios, así como la implementación de la propuesta en una tecnología de agentes específica.

El resto de este trabajo está organizado como sigue. En la sección 2 se amplía el contexto del problema de transacciones distribuidas en la arquitectura de microservicios introducido anteriormente, incluyendo aproximaciones existentes para resolverlo parcialmente. En la sección 3 se detalla la propuesta, describiendo la idea general, un modelo para representar transacciones, un formato de configuración para describir el flujo de una transacción y finalmente un conjunto de estrategias para el manejo de errores. Finalmente, en la sección 4 se presentan diversas conclusiones del trabajo acompañadas del trabajo futuro por realizar.

2. Contexto

Las transacciones distribuidas no son un tema nuevo en ingeniería de software, son un problema inherente de los sistemas distribuidos y existen varios esfuerzos propuestos en la literatura para tratarlas, tales como el coordinador de transacciones distribuidas de Microsoft (MSDTC) de la plataforma .NET [7], así como diversas implementaciones de manejadores de bases de datos como Oracle y MySQL que utilizan un protocolo de manejo de transacciones conocido como commit de dos fases (two-phase commit) [9].

Sin embargo, las soluciones tradicionales, como las antes mencionadas, suelen adolecer de los siguientes problemas en el contexto de microservicios [11]:

- Homogeneidad. Cada parte del sistema debe ser tecnológicamente compatible, ya sea en tecnología de persistencia o implementación, para que el mecanismo funcione.
- 2. Baja escalabilidad. Muchas soluciones trabajan de forma síncrona, esto es, los componentes pueden detener su trabajo o bloquear tablas de la base de datos a la espera de la respuesta de otros componentes.

Otra posibilidad, en el contexto de microservicios, es que los microservicios mismos se coordinen de forma síncrona para realizar las transacciones. Esta aproximación, aunque directa y heterogénea, puede adolecer del problema de escalabilidad antes mencionados, siendo difícil de implementar de forma adecuada y confiable, sobre todo en el aspecto de manejo de errores.

A raíz de los problemas mencionados, especialmente el referente a escalabilidad, en tiempos recientes se ha adoptado un modelo de persistencia conocido como consistencia eventual [1], dicho modelo no requiere de bloqueos y permite el trabajo concurrente sobre datos de transacciones incompletas y datos parcialmente actualizados. El modelo establece que de no haber más actualizaciones futuras, la base de datos eventualmente se vuelve consistente. La consistencia eventual permite un modelo de trabajo asíncrono, esto es, no se requiere de coordinadores en la transacción ni de esperar a que otro componente termine su trabajo, cada componente puede trabajar de forma independiente.

La consistencia eventual puede considerarse de consistencia débil y conlleva ciertos riesgos. Para que pueda ser implementado de forma adecuada, el desarrollador debe tener en cuenta las situaciones de concurrencia que pudieran generar inconsistencias, complementándolas con código de dominio [11]. Por ejemplo, en una aplicación para realizar compras, en vez de directamente substraer dinero de la cuenta del cliente, primero se reserva el crédito necesario para realizar una compra y si la transacción sale bien, se substrae el dinero de la cuenta como paso final.

Dado el panorama antes planteado, surge la necesidad de contar con formas concretas para lidiar con transacciones distribuidas en el contexto de microservicios. En la sección siguiente, se presenta una serie de patrones de diseño relacionados que intentan resolver el problema planteado.

2.1. Patrones de diseño relacionados

Posiblemente el patrón de diseño más importante y conocido para el manejo de transacciones distribuidas con consistencia eventual sea el patrón Saga [5]. Una saga es una secuencia de transacciones locales donde cada transacción actualiza datos de un solo microservicio. La primera transacción es iniciada por una petición externa, luego cada paso subsecuente es disparado tras haber completado el paso previo, si algo sale mal en cualquier paso, se ejecuta una serie de acciones de compensación que intentan limpiar la transacción. Este patrón puede ser implementado de dos formas generales:

 Orquestada. Existe un coordinador central de transacciones, quien orquesta cada paso de la transacción y coordina las operaciones de compensación en caso de fallos. Cada vez que un paso se completa, el microservicio le avisa al coordinador que ha terminado. Esta orquestación permite que los microservicios se mantengan desacoplados. La orquestación puede ser síncrona o asíncrona.

 Coreografiada. Basada en eventos asíncronos. Cada microservicio produce un evento cada vez que completa un paso de la transacción o un error ocurre, otros microservicios escuchan dicho evento y continúan con el proceso acorde.

La aproximación orquestada tradicional tiene la desventaja de tener un sólo punto central de fallo, además de que conlleva a la centralización de tráfico. Así mismo, el coordinador de transacciones puede ser por si solo un módulo muy complejo de crear y mantener. Otra desventaja de esta aproximación es que, para ser confiable, requiere de atomicidad entre la finalización de una operación local y la notificación al coordinador, de no haber dicha atomicidad se puede caer en casos en los que la operación local finalizó pero por algún error no se le pudo notificar al coordinador este hecho.

Al igual que la aproximación orquestada, la aproximación coreografiada requiere, para ser confiable, de atomicidad entre la finalización de la operación local y la publicación del evento asociado. Una aproximación posible es un patrón de diseño llamado Event Sourcing [14]. Event Sourcing establece la persistencia de entidades de dominio, tales como ordenes y clientes, como una secuencia de eventos de cambio de estado. Cada vez que el estado de una entidad de dominio cambia, un nuevo evento se agrega a su cola de eventos. Si se quiere recuperar el estado actual de la entidad de dominio, simplemente se ejecutan sus eventos en orden. Dado que almacenar un evento es una sola operación, esta aproximación es inherentemente atómica.

En Event Sourcing, las aplicaciones persisten eventos en un almacén de eventos, el cuál cuenta con los mecanismos necesarios para agregar y obtener los eventos de una entidad. Cuando se une con el patrón Saga, el almacén de eventos también sirve como intermediario para la coordinación coreografiada de transacciones, permitiendo que microservicios se suscriban a eventos que pueden almacenar otros microservicios.

Al utilizar Saga y Event Sourcing en conjunto, surge el problema de no tener directamente disponible el estado actual (o incluso uno próximo al actual) de una entidad para realizar consultas, para solucionar este problema se utiliza otro patrón de diseño conocido como CQRS [15] (Command Query Responsibility Segregation). En dicho patrón la aplicación se divide en dos partes: el lado de comando y el lado de consultas. El lado de comando maneja operaciones de creación, actualización y borrado, emitiendo los eventos pertinentes; esencialmente este es el lado de Event Sourcing. El lado de consultas maneja consultas, las cuales se ejecutan contra una o más vistas materializadas de la entidad; por ejemplo, una vista materializada puede ser obtenida a través de ejecutar los eventos de la cola de la entidad en el almacén de eventos. Para crear vistas de forma más eficiente, se pueden guardar imágenes periódicas de las entidades en una base de datos, de esta forma la reconstrucción de la entidad también se

simplifica al no tener que ejecutarse todos los eventos históricos para reconstruir el estado actual.

La aproximación coreografiada es una forma natural y descentralizada de implementar el patrón Saga, que promueve bajo acoplamiento entre microservicios, sin embargo, conforme más microservicios participan en la transacción, se vuelve más complejo entender el flujo de la misma y determinar qué servicios escuchan qué eventos, es también necesario evitar dependencias cíclicas que puedan crear bucles infinitos de eventos. Además de esto, para ser confiable y utilizable, requiere ser acompañado de otros patrones de diseño, e.g., Event Sourcing y CQRS, lo cual puede complicar la implementación y traer otros problemas consigo. Por ejemplo, al utilizar Event Sourcing, el almacén de eventos representa un punto central de fallo. Así mismo, a pesar de la mejora de confiabilidad que Event Sourcing establece, sigue sin definirse una forma o recomendación para manejar casos en los que un evento se emite pero el servicio receptor se encuentra inaccesible; de no tratarse casos como este, la transacción queda incompleta y con datos inconsistentes en el almacén de eventos.

Muchos de los problemas mencionados de la aproximación coreografiada y sus patrones asociados se deben a que la transacción en si no existe como un concepto de alto nivel, la transacción queda oculta a bajo nivel en forma de eventos, lo que complica su tratamiento atómico. En la aproximación que se propone, se intentan remediar, o al menos mitigar, los problemas de aproximaciones existentes, mientras al mismo tiempo se simplifica el proceso al mantener una visión de alto nivel de las transacciones, explotando el nivel de abstracción elevado que los sistemas Multi-Agente permiten.

3. Propuesta

En esta sección se propone una aproximación para implementar transacciones distribuidas para la arquitectura de microservicios, dicha aproximación tiene las siguiente características:

- Basada en agentes. Se explotan conceptos de sistemas Multi-Agente tales como comunicación basada en actos de habla [10] y coordinación.
- Basado en consistencia eventual, dado que es el modelo de persistencia de datos que permite mayor escalabilidad y heterogeneidad.
- La aproximación propuesta puede verse como una posible implementación del patrón Saga introducido en la sección 2.1.
- Sigue un modelo semi-orquestado asíncrono. Es simi-orquestado porque no existe una unidad central de mando sino que cada agente le pide a otro que realice el siguiente paso; es asíncrona ya que los agentes son capaces de comunicar o pedir algo sin necesidad de esperar una respuesta, así mismo los agentes pueden exhibir comportamiento reactivo cuando un evento sucede.
- Define un modelo de transacción que permite describir flujos de trabajo así como configuraciones de transacciones.
- Propone estrategias de manejo de errores para la detección, tolerancia y recuperación de errores, dichas estrategias explotan el nivel de abstracción elevado de los sistemas Multi-Agente.

La idea general de la propuesta es que cada microservicio tenga asociado un agente particular, dicho agente puede estar o no en el mismo servidor. Cada transacción nace en algún microservicio, el cual cuenta con los medios para comunicar, de forma directa o indirecta, el inicio de la transacción a su agente asociado. La forma concreta de comunicación depende de la implementación, por ejemplo, si tanto el microservicio como el agente se encuentran en el mismo servidor, la comunicación puede establecerse de forma simple mediante un pipe, o podrían utilizarse tecnologías de agente concretas tales como CArtAgO [13] que le permiten al agente reaccionar ante señales emitidas por el microservicio. Una vez el inicio de la transacción le es informada al agente, la transacción es manejada por una capa Multi-Agente independiente, la cual se encuentra también a cargo del manejo de errores referentes a transacciones y a la propia capa de agentes. La figura 1 muestra la idea general de la aproximación propuesta.

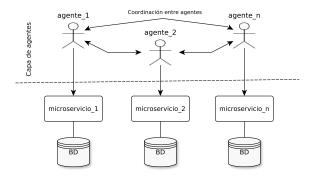


Fig. 1. Modelo general. Cada microservicio tiene asociada una base de datos BD y un agente. Los agentes, en la capa de agentes independiente, se coordinan entre si para realizar transacciones entre microservicios.

3.1. Modelo de transacción

En esta sección se presenta un modelo de transacción propuesto, el cual facilita la configuración y representación a nivel agente de las transacciones, dicho modelo se apoya en conceptos de teoría de grafos.

Definición 1 Una transacción $T = \{St_1, St_2, ..., St_n\}n >= 1$ es un conjunto de caminos simples, i.e., que no pasan más de una vez por el mismo vértice, llamados sub-transacciones.

Definición 2 Una sub-transacción $St = \{sc_1, sc_2, ..., sc_m\}m >= 1$ es un conjunto de caminos de longitud 1 llamados secciones.

De esta forma, las transacciones pueden verse como secuencias lineales que son descompuestas en secciones. En cada sección se ejecuta un paso de la transacción, para esto, es necesario contar con información asociada.

Definición 3 $IS = \langle tr_id, st_name, input_data, incoming_action, output_data, compensation_action, state, next \rangle es un tupla de información asociada a la sección, donde:$

- tr_id: identificador único de la transacción a la cual pertenece la sección.
- st_name: nombre lógico de la sub-transacción a la cual pertenece la sección.
- input_data: datos asociados para ser procesados por la sección.
- incoming_action: nombre de la acción del microservicio a ser ejecutada en la sección. La acción recibe input_data.
- output_data: datos producidos por incoming_action.
- compensation_action: nombre de la acción del microservicio a ser ejecutada si incoming_action ya ha sido ejecutada y algo salió mal en cualquier punto de la transacción. La acción recibe output_data.
- state: establece el estado de la ejecución de la sección, es un valor del conjunto {pendant, processing, processed, pendant_compensation, processing_compensation, processed_compensation}.
- next: agente que continuará con la sub-transacción st_name para la transacción actual tr_id. También puede establecer un enlace entre la sub-transacción actual st_name y una nueva new_st_name, esto en caso de que se requiera visitar nuevamente a una agente, esto es, crear un camino que no sea simple.

La aproximación propuesta no tiene lineamientos específicos en cuanto a cómo representar o crear <code>input_data</code> y <code>output_data</code>, esto depende de los desarrolladores de microservicios y de la implementación especifica del modelo. Por ejemplo, se puede preferir el uso de un formato de texto estándar como JSON donde los desarrolladores pueden elegir utilizar sólo algunos de los datos de entrada para sus operaciones.

Es también interesante notar que el modelo permite la creación de subtransacciones que trabajan de forma paralela, esto gracias a que es posible ligar una sub-transacción existente con una nueva a través de next, siempre y cuando no haya dependencia directa entre las sub-transacciones, en cuyo caso se puede plantear una sola sub-transacciones secuencial.

3.2. Configuración

La mayoría de información asociada a una sección, descrita en la definición 3, puede ser generada de forma dinámica por la capa de agentes, sin embargo, algunos datos deben ser configurados por los desarrolladores de microservicios: st_name , $incoming_action$, $compensation_action$ y next. Se propone un formato de configuración basado en lógica de primer orden el cual tiene la finalidad de representar las transacciones en las cuales los agentes participan. La configuración se realiza por agente, y es, junto con las acciones a ejecutar, los únicos aspectos referentes a transacciones por los cuales debe de preocuparse el desarrollador de microservicios. El formato propuesto contiene los siguientes predicados:

- incoming_action(st_name, action): establece la acción action que se ejecutará en el microservicio dada la sub-transacción st_name. La acción se ejecuta utilizando input_data y genera output_data como se mencionó en la sección 3.1.
- compensation_action(st_name, action): establece la acción action a ser ejecutada en caso de que haya algún problema en la transacción y state = processed para la sección correspondiente st_name. La acción se ejecuta utilizando output_data.
- $next(st_name, agent)$: establece al siguiente agente agent para continuar la sub-transacción st_name .
- next(st_name, new_st_name, agent): liga una sub-transacción st_name existente con una nueva new_st_name y establece el agente agent que empezará con dicha sub-transacción. Este predicado es de utilidad para crear caminos no simples en la transacción sin ambigüedad, i.e., cuando el siguiente agente de una sub-transacción ya había sido visitado.

Como ejemplo de configuración, considérese un caso donde un cliente quiere establecer una orden de compra sobre un producto. A continuación se muestra una posible configuración utilizando el formato propuesto.

```
//Agente order
incoming_action(orderCreation, createPendingOrder).
compensation_action(orderCreation, deleteOrder).
next(orderCreation, customer).
incoming_action(orderApproval, approveOrder).
compensation_action(orderApproval, cancelOrder).

//Agente customer
incoming_action(orderCreation, reserveCredit).
compensation_action(orderCreation, cancelReservation).
next(orderCreation, orderApproval, order).
```

El flujo de la transacción descrito en la configuración puede ser representado fácilmente mediante un diagrama similar al de secuencia de UML, el cual se muestra en la figura 2. Notar que la figura sólo muestra el flujo normal de la transacción, i.e., sin acciones de compensación. Así mismo, por legibilidad, en la figura no se muestran los datos de entrada y salida de cada acción. Esta facilidad de representación es gracias al manejo de alto nivel de las transacciones y muestra la posibilidad de una herramienta que genere automáticamente los archivos de configuración a través de una interfaz gráfica donde se modela el flujo principal y alterno (acciones de compensación) de la transacción.

3.3. Flujos de trabajo

Además de un modelo de transacción y una forma de configuración, es necesario que la capa de agente realice diversas labores que permitan dar soporte a las transacciones. En esta sección se presenta, desde la perspectiva de la capa de agente, el flujo de trabajo de una transacción, incluyendo los casos cuando la

Xavier Limón, Alejandro Guerra-Hernández, Angel J. Sánchez-García, et al.

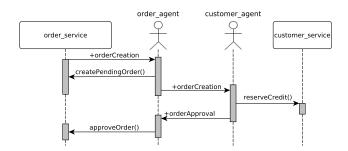


Fig. 2. Flujo normal de la transacción del ejemplo de ordenes y clientes. El símbolo "+" se utiliza para denotar una nueva creencia.

transacción se realiza de forma normal y cuando se genera algún fallo. El flujo normal de una transacción es el siguiente:

- 1. Un microservicio comunica el inicio de la transacción a su agente asociado, entregando datos de entrada *input_data* y el nombre de la sub-transacción inicial *st_name*.
- 2. El agente establece qué agentes participarán en la transacción y qué subtransacciones forman parte de la transacción. Esta información debe ser conocida por todos los agentes y dependiendo de la implementación, podría conocerse de antemano o ser descubierta.
- 3. Un identificador único tr_{-id} es creado para la transacción.
- 4. En cada sección el agente correspondiente ejecuta el *incoming_action* apropiado, pasando *input_data* como entrada y generando *output_data* como salida.
- 5. Utilizando la información de *next*, el agente de la sección pasa *output_data* y un *st_name* al siguiente agente.
- 6. Cuando una sub-transacción finaliza, el último agente en el extremo final se lo comunica a los demás agentes de la transacción. Una sub-transacción finaliza cuando ya no tiene asociado un next en la sección actual, o cuando en next se liga una sub-transacción nueva.
- 7. Cuando todas las sub-transacciones finalizan, la transacción finaliza.

El flujo alterno de compensación sólo considera problemas que pudieran darse a nivel de la acción incoming_action, por ejemplo, en el contexto de ordenes y clientes, un error de este tipo puede ser que el cliente no cuente con crédito suficiente, en dicho caso la acción falla y por lo tanto toda la transacción. Una posibilidad de manejo de errores derivados de perdida de conectividad y caída de agentes se específica en la sección siguiente. El flujo de compensación ante un fallo generado en una acción es el siguiente:

1. Un incoming_action de una sección falla y el agente asociado lo percibe.

- El agente envía un mensaje a los demás agentes de la transacción para que detengan su trabajo relacionado con la transacción, esto en caso de que haya sub-transacciones paralelas.
- 3. Los agentes consideran el estado state de sus secciones y ejecutan las acciones compensation_action en caso de ser necesario, utilizando el output_data apropiado para la sección. La ejecución de los compensation_action se realiza en el orden inverso a los de incoming_action.
- 4. Cuando un agente termina de ejecutar sus acciones de compensación, se lo comunica a los demás agentes de la transacción.
- 5. Cuando todos los agentes terminan, la transacción termina.

3.4. Estrategias de manejo de errores

En un sistema distribuido es importante considerar el fallo independiente de componentes y establecer estrategias de detección, tolerancia y recuperación de errores apropiadas [3]. En este apartado, se propone una serie de posibles estrategias a implementar para el manejo de errores, éstas sólo consideran problemas que pueden surgir en la capa de agente ya que los problemas derivados de microservicios pueden ser manejados a través del flujo alterno de compensación presentado en la sección 3.3. Estas estrategias son de vital importancia pues un mal manejo de éstas puede crear inconsistencias de datos y por consiguiente pérdida de integridad. Para el manejo de errores, se consideran los dos tipos de errores más generales que pueden ocurrir:

- 1. Pérdida de conectividad entre agentes. Se refiere a problemas de red, los cuales pueden ser temporales y afectan la coordinación entre agentes.
- Caída de algún agente. Agrupa problemas tales como la generación de una excepción no controlada hasta fallos de hardware que provocan la detención abrupta del proceso del agente.

De los dos tipos de errores generales mencionados, el más severo es la caída de agentes puesto que puede requerir de un reinicio manual y una recuperación de error más compleja, sin embargo, desde el punto de vista de los agentes, el efecto de cualquiera de los dos errores mencionados es no poder establecer alguna comunicación con los agentes afectados. Así mismo, la recuperación de errores de caída tiene elementos en común de la recuperación de error derivada de problemas de conectividad. De esta forma, la estrategia de detección de errores propuesta se basa sólo en pérdida de conectividad y se propone una estrategia general de recuperación de errores que aplica a errores de conectividad y parcialmente a errores de caída, posteriormente se propone una extensión para la recuperación de errores de caída. La estrategia de detención, tolerancia y recuperación de errores general es la siguiente:

Como parte de la capa de agentes, cada agente mantiene un intercambio periódico de información de conectividad con uno o dos agentes. Esta conectividad es lineal, de esta forma, el intercambio puede ser de izquierda a derecha y de derecha a izquierda. Los agentes de cada extremo sólo mantienen intercambio con un agente que le da información sobre la conectividad de todos los agentes en el extremo opuesto; mientras que los demás agentes mantienen intercambio con otros dos agentes, del agente de su izquierda reciben información de conectividad de los demás agentes del lado izquierdo, mientras a su vez le informan sobre la conectividad de los agentes que están hacía la derecha, esto mismo ocurre del lado derecho de forma inversa. El proceso descrito anteriormente se muestra en la figura 3.

- Con la información de conectividad, los agentes pueden determinar si tiene sentido o no iniciar una nueva transacción o continuar alguna en curso. Este aspecto puede verse como una forma de tolerancia a fallos.
- Si un agente pierde conectividad, este hecho es reportado a todos los agentes por uno de los agentes que esperaba un mensaje del agente afectado. Durante esta perdida, es posible que algún agente haya intentando contactar con el agente afectado, esto como parte del trabajo de una transacción, si este es el caso, entonces la transacción queda en pausa.
- Los agentes no afectados esperan un periodo de tiempo de gracia antes de realizar alguna recuperación de error, por si una reconexión ocurre.
- Durante el periodo de gracia, los mensajes de conectividad se intentan enviar de forma normal.
- Si la reconexión ocurre durante el periodo de gracia, entonces los agentes que tenían pendiente contactar con el agente afectado reinician su trabajo.
- Si la reconexión no ocurre durante el periodo de gracia, los agentes con conectividad ajustan su intercambio de información de conectividad con los agentes disponibles y continúan trabajando, todo esto como medida de tolerancia a errores, al mismo tiempo, siguen intentando contactar con los agentes no alcanzables. Los agentes afectados por transacciones incompletas inician un proceso similar al descrito en el flujo alterno de compensación mencionado en la sección 3.3, con la salvedad de que la transacción no se considera como terminada sino hasta que eventualmente haya una reconexión y se pueda realizar una limpieza completa.

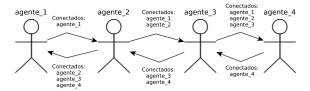


Fig. 3. Intercambio de información de conectividad entre agentes. Cada agente comparte su información con uno o dos agentes para mitigar carga de red.

Es importante notar que al ocurrir un problema de conectividad es posible que varios agentes en un mismo segmento de red pierdan conectividad con los

agentes de otro segmento, en la estrategia mencionada, cada segmento lleva a cabo los pasos antes descritos de forma independiente. La estrategia general de recuperación de error ante una reconexión que sucede después del tiempo de gracia es la siguiente:

- Cuando uno o varios agentes se reconectan se intenta recrear el intercambio de conectividad como estaba antes de la perdida de conexión.
- Si todos los agentes que forman parte de alguna transacción incompleta se reconectan, la transacción se da por finalizada, considerando que dichos agentes ya ejecutaron sus acciones de compensación.

La extensión de la estrategia antes mencionada para caídas es la siguiente:

- Cada agente mantiene una bitácora persistente del estado de cada una de sus transacciones.
- Cuando el agente se reinicia, éste verifica su bitácora y ejecuta las acciones de compensación adecuadas.
- El agente se reconecta con los demás agentes.

Para que el proceso de recuperación de errores derivado de caídas sea confiable, es necesario que existe atomicidad entre la escritura de la bitácora y la realización de las acciones que cambian el estado de la transacción, esto puede lograrse de diversas formas a nivel de implementación, por ejemplo, utilizando transacciones de código tradicionales de forma local.

4. Conclusiones y trabajo futuro

La aproximación presentada en este trabajo promueve un modelado de alto nivel de transacciones distribuidas escalables y heterogéneas, proveyendo una definición flexible de transacción, la cual es posible representar como conocimiento de agente a través del formato de configuración propuesto. El utilizar sistemas Multi-Agente permite la definición de alto nivel de flujos de trabajo y estrategias de manejo de errores, dejando a su vez abiertas oportunidades para mejorar aspectos del proceso a través de esquemas más sofisticados de negociación y razonamiento de agente que exploten técnicas de Inteligencia Artificial.

Desde el punto de vista de ingeniería de software, la aproximación propuesta promueve un bajo acoplamiento entre microservicios, ya que las coordinación de las transacciones se encuentra aislada en la capa de agentes. De igual forma, el acoplamiento entre el lado de microservicios y la capa de agentes se mantiene bajo. Desde el punto de vista de los microservicios, no existe la capa de agentes, salvo quizás para resolver cosas como el inicio de una transacción, lo cual es dependiente de la implementación concreta; a su vez, la capa de agentes es genérica, puede adaptarse a cualquier sistema basado en microservicios por lo que no existe una dependencia de dominio entre ambas partes, permitiendo de esta forma que cada parte evolucione de forma separada. Este acoplamiento bajo en todos los niveles tiene como beneficio facilitarle en gran medida a los desarrolladores el manejo de transacciones distribuidas, ya que éste no debe

preocuparse de establecer código especial, como por ejemplo para manejar eventos, sino que la gestión confiable y eficiente de la transacción la lleva a cabo la capa de agentes. Así mismo, no se tiene un sólo punto central de fallo o de concentración de tráfico, como en muchas otras aproximaciones existentes discutidas en la sección 2. El trabajo presentado es un esfuerzo para acercar cada vez más a los sistemas Multi-Agente al desarrollo de software tradicional, mostrando que una aproximación de este tipo tiene argumentos suficientes para ser considerada seriamente en Ingeniería de Software.

La propuesta, más que una guía de cómo llevar a cabo el proceso de transacciones distribuidas, como es el caso de los patrones de diseño presentados en la sección 2.1, da lineamientos más específicos de cómo realizar una implementación funcional que considera también aspectos de manejo de errores. Esta guía es de gran valor para la creación futura de una implementación específica, lo cual se planea hacer como trabajo futuro inmediato. En esta implementación futura será necesario escoger una pila de desarrollo Multi-Agente, teniéndose como principal opción a JaCaMo [2], el cual cuenta con los elementos suficientes para la creación de sistemas Multi-Agente distribuidos basados en el modelo BDI [12] de agencia.

El modelo BDI deja abiertas posibilidades para mejorar aspectos de razonamiento, negociación y coordinación entre agentes. Uno de los objetivos futuros es explorar esquemas más sofisticados en el manejo de transacciones, para que, por ejemplo, los agentes sean capaces de razonar si se debe ejecutar o no una operación sobre datos que están siendo manipulados en otra transacción, con el objetivo de mitigar posibles problemas derivados del uso de consistencia eventual.

Referencias

- Bailis, P., Ghodsi, A.: Eventual consistency today: Limitations, extensions, and beyond. Queue 11(3). 20 (2013)
- Boissier, O., Bordini, R.H., Hübner, J.F., Ricci, A., Santi, A.: Multi-agent oriented programming with jacamo. Science of Computer Programming 78(6), 747–761 (2013)
- 3. Coulouris, G.F., Dollimore, J., Kindberg, T.: Distributed systems: concepts and design. pearson education (2005)
- Dragoni, N., Giallorenzo, S., Lafuente, A.L., Mazzara, M., Montesi, F., Mustafin, R., Safina, L.: Microservices: yesterday, today, and tomorrow. In: Present and Ulterior Software Engineering, pp. 195–216. Springer (2017)
- 5. Garcia-Molina, H., Salem, K.: Sagas, vol. 16. ACM (1987)
- Josuttis, N.M.: SOA in practice: the art of distributed system design. .º'Reilly Media, Inc." (2007)
- 7. Limprecht, R.: Microsoft transaction server. In: Compcon'97. Proceedings, IEEE. pp. 14–18. IEEE (1997)
- 8. Moemeng, C., Gorodetsky, V., Zuo, Z., Yang, Y., Zhang, C.: Agent-based distributed data mining: A survey. In: Data mining and multi-agent integration, pp. 47–58. Springer (2009)
- 9. Mohan, C., Lindsay, B.: Efficient commit protocols for the tree of processes model of distributed transactions. In: Proceedings of the second annual ACM symposium on Principles of distributed computing. pp. 76–88. ACM (1983)

- 10. Moreira, Á.F., Vieira, R., Bordini, R.H., et al.: Extending the operational semantics of a bdi agent-oriented programming language for introducing speech-act based communication. Lecture notes in computer science pp. 135–154 (2004)
- 11. Newman, S.: Building microservices: designing fine-grained systems. .°'Reilly Media, Inc." (2015)
- 12. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI architecture. Tech. rep., Australia Artificial Intelligence Institute, Melbourne, Australia (1991)
- 13. Ricci, A., Viroli, M., Omicini, A.: Construenda est cartago: Toward an infrastructure for artifacts in MAS. Cybernetics and systems 2, 569–574 (2006)
- 14. Richardson, C.: Microservice architecture patterns and best practices. URL: http://microservices. io/index. html [accessed: 2018-03-17] (2016)
- 15. Young, G.: Cqrs and event sourcing. feb. 2010. URl: http://codebetter.com/gregyoung/2010/02/13/cqrs-and-event-sourcing (2010)

Máquina de estados finita con comportamientos de dirección para manipular agentes de juego RedCell

Abner Quiroz Clemente, Héctor Adrián Díaz Furlong, Verónica González Rivera, Juan Jesús Cuéllar González

Benemérita Universidad Autónoma de Puebla, Escuela de Artes Plásticas y Audiovisuales, Puebla, México

{abner.quiroz, hector.diazfurlong}@correo.buap.mx, {veronica.gonzalezri, juan.cuellarg}@alumno.buap.mx

Resumen. RedCell es un videojuego surgido del Global Game Jam 2018, creado por integrantes de la Escuela de Artes Plásticas y Audiovisuales (ARPA) de la Benemérita Universidad Autónoma de Puebla, en el cual se lleva a cabo una competencia dentro del cuerpo humano para contaminar células y destruir al oponente. Parte de este juego es la competencia contra un agente controlado por la computadora que busca también contaminar y vencer al jugador. Este trabajo se enfoca en la descripción de cómo se ha ido desarrollando el sistema que controla al agente autónomo, a través del diseño e implementación de una máquina de estados que manipula el flujo de ejecución y, además, permite que el agente utilice distintos comportamientos de dirección para la locomoción del jugador artificial. La combinación de máquinas de estados con comportamientos de dirección permite generar conductas complejas que brindan una sensación de competencia dentro del juego, que es el objetivo del desarrollo de esta inteligencia artificial, procurando siempre mantener un balance en la dificultad para evitar frustración por ser demasiado difícil o aburrimiento por ser demasiado sencilla.

Palabras clave: máquina de estados finita, comportamientos de dirección, agentes de juego, Raycast, RedCell.

Finite State Machine with Steering Behavior for RedCell Game Agent Manipulation

Abstract. RedCell is a videogame developed at the 2018 Global Game Jam by members of the Escuela de Artes Plásticas y Audiovisuales (ARPA) of the BUAP, in which a competition is carried out within the human body to contaminate cells and to destroy the opponent. Part of this game is the competition against a computer-controlled agent that seeks also to contaminate and to destroy the player. This work focuses on the description of how the system that controls the autonomous agent has been developed through the design and implementation of a state machine that manipulates the flow of execution and

that allows the agent to use different steering behaviors for the artificial player's locomotion. The combination of state machines with steering behaviors allows to generate complex behaviors that provide a sense of competition within the game, which is the objective of the development of this artificial intelligence, always trying to maintain a balance in the difficulty to avoid frustration because it is too difficult or boredom because it is too simple.

Keywords: finite state machine, steering behavior, game agents, Raycast, RedCell.

1. Introducción

1.1. Inteligencia Artificial en los videojuegos

La Inteligencia Artificial (IA) en los juegos ha estado presente desde juegos como Pacman, Asteroids y Space Invaders y esto es porque el uso de personajes no jugadores (NPC, por sus siglas en inglés), descritos en [5], brinda una mayor presencia dentro de la experiencia del juego. Los diseñadores de juegos deben tratar de proporcionar un balance dentro de la inteligencia de los NPC ya que el objetivo de la IA dentro de un videojuego no es sobrepasar al humano (jugador) sino brindar la ilusión de que no está solo dentro del mundo virtual, a través de compañeros, guías, enemigos, etcétera.

En este trabajo se usan agentes [1] de juego que percibirán su entorno a través de colisionadores circulares y raycasting, los cuales mandarán información para elegir sus movimientos y elegir el momento de aproximarse a otro jugador.

1.2. Máquinas de estados finitas para dirigir la IA

Las máquinas de estados finitas (FSM) son ampliamente utilizadas dentro del mundo de la IA de los videojuegos debido a que entre otras ventajas que tienen, es que son intuitivas, flexibles y fáciles de programar [2].

Esta parte del agente se encarga de escoger los objetivos o tareas a realizar, es decir que lleva a cabo la planeación de acciones.

La estructura que sigue cada estado de la FSM implementada consiste en las siguientes funciones, que son similares en [2] y [9]:

- —Enter. Las acciones que realiza el NPC una sola vez al inicio del cambio de estado;
- Reason. Esta función verifica cada frame si alguna condición se cumple para que la FSM realice un cambio de estado;
- —Act. Son las tareas del estado que se llevan a cabo durante el lapso que dure éste;
- —Exit. Las tareas de finalización del estado. Se llama una sola vez cuando hay un cambio de estado.

1.3. Comportamientos de dirección para manipular el movimiento de un agente

Para controlar la locomoción de un NPC dentro del juego se usan comportamientos de dirección (Steering behavior) que se encargan de calcular las trayectorias de movimiento que debe seguir el agente para satisfacer los objetivos planteados en la parte de planeación realizada en la FSM.

Existen distintos comportamientos para buscar, seguir, huir, perseguir, esconderse, etc. Todos ellos calculan la trayectoria que debe seguir el agente dejando la locomoción, es decir los aspectos mecánicos del movimiento, a otro componente del programa [4].

Los comportamientos se enfocan sólo en su objetivo como el objeto a perseguir o el jugador a evadir, y en el movimiento que deben realizar para cumplir su tarea.

1.4. RedCell

RedCell es un juego de competencia en el cual varios jugadores deben contaminar células a través del contacto o usando un proyectil. En ese juego existen dos etapas:

- 1. La primera consiste en la contaminación de todas las células en el escenario. El juego inicia con cierto número de células que cada jugador debe tratar de contaminar a través de tocarlas o impactar con ellas con un proyectil.
- 2. Una vez que todas las células han sido contaminadas entonces la segunda etapa se activa y los jugadores pueden hacerse daño entre ellos. Al disparase si un proyectil enemigo impacta entonces el jugador desaparece y es en este momento que entra en juego la cantidad de células que contaminaron, pues el jugador impactado puede revivir en alguna de las células que le pertenecen si es que tiene. Una vez que el jugador revivió, la célula desaparece.

Este juego surgió en el Global Game Jam 2018, un evento de tipo maratón de 48 horas, donde se forman equipos y se crean videojuegos en ese lapso. Una vez terminado el evento se retomó el juego para su desarrollo completo a partir del prototipo mostrado en la Fig. 1.

Es en el desarrollo posterior del juego que se empezó el diseño y construcción de NPC para que una persona pueda competir contra una IA.

2. Diseño e implementación de la FSM

Las acciones principales que debe desempeñar el NPC en este juego están ilustradas en la Fig. 2 y a continuación se describen:

Abner Quiroz Clemente, Héctor Adrián Díaz Furlong, Verónica González Rivera, et al.

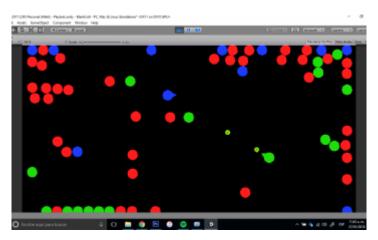


Fig. 1. Primer prototipo de RedCell en el Global Game Jam 2018. Era un juego para dos personas, aún no existía una IA.

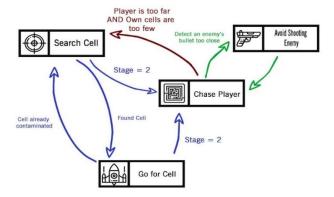


Fig. 2. Diagrama de la FSM que controla la toma de decisiones del NPC en el juego.

Buscar una célula. Durante la primera etapa del juego el agente deberá buscar células que contaminar, pues tener estas células significa tener vidas para participar en la segunda etapa. Esto se hace a través de una lista de objetos que almacena las células en el escenario.

Ir por una célula. El estado de Buscar solo se enfoca en localizar alguna célula para contaminar, pero en este estado es cuando se decide una vez ubicada la célula ir tras ella para contaminarla. El estado activará un comportamiento de persecución (Pursuit) que le permite al agente acercarse para contaminar a través de contacto o de un proyectil.

Perseguir al jugador. Cuando el juego entre en la segunda etapa el agente debe tomar como prioridad atacar al jugador para poder destruirlo.

3. Selección y construcción de los comportamientos de dirección

Para lograr que el agente realice las acciones que se planean en la FSM se usan comportamientos de dirección. En los estados de ir tras la célula y perseguir al jugador se utiliza el comportamiento de persecución.

3.1. Persecución

Este comportamiento busca calcular la posición donde se encontrará el objetivo para tratar de acercarse lo más posible a este. No sirve simplemente que el NPC se dirija a la posición actual de una célula o del jugador, pues estos tienen movimiento constantemente y jamás llegaría a alcanzar su destino, ilustrado en la Fig. 3.

Para poder calcular la posición donde se encontrará el objetivo (una célula o el jugador rival) se toma en cuenta la velocidad del agente y la del objeto perseguido y sus posiciones.

Primero calculamos un vector que nos indique la dirección desde el NPC hacia el objetivo, como se describe en [7, 8], usando la ecuación (1).

$$T = O - N, (1)$$

donde T es el vector resultado de 2 o 3 dimensiones que apunta hacia el jugador o célula destino, O es la posición del objetivo y N es la posición del NPC en el escenario.

Una vez encontrada esta dirección podemos calcular el tiempo de anticipación a tomar en cuenta usando las velocidades de los objetos en cuestión.

$$L = ||T|| / (NV + OV), \tag{2}$$

donde L es el tiempo de anticipación que calculamos, NV es la velocidad del NPC y OV es la velocidad del objetivo. Este tiempo encontrado en la ecuación (2) de anticipación es proporcional a la distancia entre el objetivo y el perseguidor; y es inversamente proporcional a la suma de las velocidades de los agentes [3].

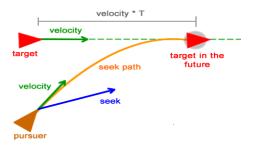


Fig. 3. Descripción gráfica del comportamiento de persecución [3]. En la imagen "pursuer" representa el NPC y "target" es una célula o un jugador rival.

Una vez calculado la cantidad de anticipación el nuevo destino del NPC se calcula como sigue.

$$S = O + OV * L, (3)$$

donde S es la posición futura del objetivo a la cual el agente ahora debe dirigirse. La ecuación (3) se usa en la función completa; en pseudo código queda de la siguiente forma.

```
void Pursuit()
{
    Vector T = O - N;
    float L = T.magnitude / (NV + OV);
// Hay que buscar la posicion calculada del evasor
    Vector S = O + OV * L;
    Vector DesiredVelocity = (S - N).normalized * NV;
// antes de aplicar, multiplicar por el peso
    SteeringForce += (DesiredVelocity - NV) * pursuitWeight;
}
```

Lo anterior es una síntesis de la función de pursuit desarrollada para RedCell. Se omiten algunos detalles.

3.2. Evasión de paredes

Como el escenario de competencia está confinado a un área rectangular para que las células en el juego puedan quedarse dentro de dicha área y estar en movimiento, es común que el NPC al calcular el movimiento que tiene que hacer al dirigirse hacia algún objetivo termine muy cerca de una pared o incluso colisionando con alguna pues los comportamientos de dirección trabajan añadiendo fuerza a los agentes y en ocasiones esto los lleva a situaciones de colisión.

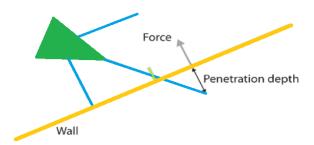


Fig. 4. En azul los sensores de colisión del agente. Para evadir paredes se toma en cuenta qué tan profundo los sensores entran en las paredes y el vector normal de dichas paredes para aplicar una fuerza que aleje al agente del choque. Imagen editada de [3].

Para evitar dichas colisiones se implementó un método de evasión, basado en la idea mostrada en la Fig. 4, de paredes que se encuentran activas todo el tiempo, el cual consiste en 3 pasos:

1. Creación de sensores. Utilizando una técnica de raycasting, se proyectan 3 vectores desde el agente. Uno hacia su frente (0 grados), otro a 45° y un último a -45°. Estos sensores trabajan como bigotes de un felino como se muestra en la Fig. 5, pues se usan para percibir si existe alguna pared cercana con la cual se pueda colisionar.

- 2. Examinación de cada sensor. Para cada uno de los 3 sensores se verifica si entraron en colisión con algún objeto etiquetado como pared. Si es así entonces se selecciona aquel cuya colisión sea más próxima a ocurrir, o sea que el punto de colisión del sensor sea el más cercano al agente.
- 3. Cálculo de fuerza. Una vez identificado el punto de colisión más próximo se procede a calcular una fuerza que lo aleje de la pared, esto a partir de que tanto el sensor penetró en la pared y el vector normal de dicha pared:

$$OS = N - C, (4)$$

donde OS es la cantidad que el sensor penetró en la pared. Esta información es importante pues mientras más cercano esté el punto de colisión quiere decir que se debe aplicar una fuerza más grande para alejar al agente lo antes posible de un choque con la pared. N es la posición del NPC y C es el punto de colisión del sensor con la pared.

Después de aplicar la ecuación (4) se procede a crear una fuerza en la dirección de la normal de la pared, con la magnitud de la cantidad de penetración encontrada:

$$F = W * \|OS\|, \tag{5}$$

donde F es el vector de fuerza aplicado al agente para alejarse de la pared, W es el vector normal de la pared y la ecuación (5) se usa para la función de evadir la pared.

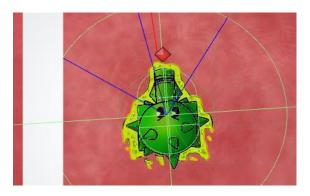


Fig. 5. Captura de RedCell en depuración. En azul, los sensores construidos con raycasting del agente. Estos sensores detectan posibles colisiones con paredes del escenario.

3.3. Evasión de proyectiles

Para darle oportunidad al NPC de competir contra un jugador humano, se diseñó y construyó un comportamiento que le da la oportunidad al agente de esquivar un proyectil lanzado por el rival.

Este comportamiento, que se bautizó como Evasión de proyectiles, tiene la siguiente lógica.

- Se crea un sensor de proyectiles circular alrededor del agente que registra cuando un proyectil del rival se acerca. Al detectar dicho proyectil se agrega a una lista de proyectiles aproximándose al NPC.
- —Se busca cuál es el proyectil más cercano al NPC.

— Calculamos una fuerza de dirección que nos aleje del proyectil. Esta fuerza depende de la cercanía del proyectil, ya que mientras más cerca se encuentre, con mayor rapidez debe reaccionar para poder intentar esquivar el disparo.

En el siguiente pseudocódigo se presenta el funcionamiento de este comportamiento:

```
void BulletAvoidance()
  //if we have found an intersecting obstacle
  if (EnemyCloseBullets.Count > 0)
    ClosestIntersectingObstacle = null;
    float DistToClosestIP = float.MaxValue;
    foreach (GameObject bullet in EnemyCloseBullets)
      // look for the closest bullet
      float dist =
        Distance(transform.position,
                   bullet.transform.position);
      if (dist < DistToClosestIP)</pre>
        ClosestIntersectingObstacle = bullet;
        DistToClosestIP = dist;
    }
    // calculate a steeringforce away from obstacle
    Vector SteeringForceLocal = Vector.zero;
    LocalPosOfClosestObstacle =
        LocalPos(ClosestIntersectingObstacle);
    float ClosestIntersectingObstacleRadius =
        ClosestIntersectingObstacle.radius;
    if (ClosestIntersectingObstacle)
      float FeelerLength = bulletFeelerRadius * 2;
      float multiplier = 1.0f +
         (FeelerLength - LocalPosOfClosestObstacle.X) /
        FeelerLength;
      //calculate the lateral force
      SteeringForceLocal.y =
         (ClosestIntersectingObstacleRadius -
ClosestIntersectingObstacle.y) * multiplier;
    }
    SteeringForce += SteeringForceLocal *
bulletAvoidanceWeight;
}
```

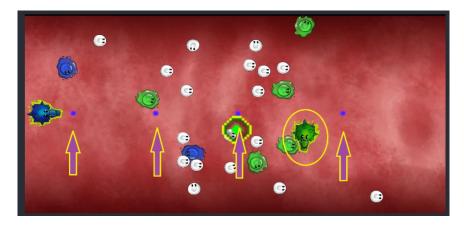


Fig. 6. NPC (verde) en movimiento tratando de esquivar los disparos del rival en azul. El jugador azul disparando proyectiles en dirección del NPC.

Este comportamiento se debe mantener activo todo el tiempo sin importar el estado en que se encuentre el NPC, pues en todo momento el agente debe estar pendiente de los disparos que haga el rival y poder tratar de reaccionar a tiempo para evitar su destrucción. En la Fig. 6 se ilustra el movimiento del agente para esquivar el ataque del rival.

3.4. Uso de raycasting para disparar

Una función importante de los jugadores es la habilidad de disparo, pues con esta es con la que se usan los proyectiles para eliminar al rival y contaminar células. Para que el agente puede disparar en la FSM se establece un estado para buscar al jugador y otro para buscar células, pero en estos estados sólo se manipula el movimiento de la IA, no se manda a disparar.

La función de disparo del NPC está activa en todo momento del juego como el comportamiento de evasión. Cuando el agente se dirige hacia su objetivo se va proyectando un rayo (raycasting) hacia el frente que ayuda a determinar si este se encuentra frente al NPC.

El raycast identifica que objetos toca y discrimina para sólo mandar a llamar la función de disparar cuando una célula se atraviese en su camino y también cuando un jugador rival lo haga durante la segunda etapa. El rayo se ilustra en la Fig. 7.

Para optimizar la función de disparo que generaba instancias de proyectiles se usó un patrón de programación llamado "Object pooling", el cual permite crear solo las instancias necesarias de un objeto y evita estar creando y destruyendo objetos en tiempo de ejecución [6].

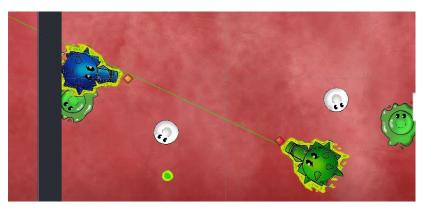


Fig. 7. Con una línea verde se aprecia el rayo (raycast) que detecta que objetos se encuentran delante del NPC. Este rayo solo se muestra para propósitos de depuración, no aparece en el juego final.

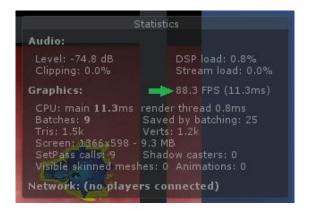


Fig. 8. La cantidad de cuadros por segundo en un juego de alta gamma se espera que sea de 60 para que el jugador no tenga problemas de interacción y sienta las animaciones del juego fluidas. En el caso de RedCell se cumple este número sin problemas.

4. Experimentación y resultados

Para poder probar el juego se realizaron partidas con jugadores humanos contra el NPC que se desarrolló. Vale la pena recordar que, para el caso de videojuegos, no es objetivo que la IA sea mejor que las personas, sino que sea retadora y genere una sensación de diversión y la necesidad de volver a jugar, es decir generar competencia.

El uso de IA no afecta el rendimiento del juego. Algunas estadísticas en la Fig. 8 muestran que el funcionamiento y "frame rate" son adecuados.

En cuanto a la experiencia de juego y el desempeño de la IA se realizaron pruebas de juego de humanos contra el NPC. En los siguientes gráficos se muestran los resultados obtenidos.

Tabla 1. ¿Cuántas veces ganaron los jugadores y cuántas el NPC?

Ganador	Porcentaje de usuarios
Jugador	31.7%
CPU	65.9%
Empate	2.4%



Fig. 9. Victorias de humanos contra CPU. El agente logró vencer en varias ocasiones a los jugadores que hicieron pruebas.

Tabla 2. ¿Qué tan difícil le pareció al jugador la IA?

Dificultad	Porcentaje de usuarios		
Muy fácil	0.0%		
Fácil	4.9%		
Medio	70.7%		
Difícil	24.4%		
Muy difícil	0.0%		

Tabla 3. Se les pidió a los jugadores que describieran en una palabra el desempeño general de la IA.

-	
Descripción del NPC	Porcentaje de usuarios
Divertido	29.3%
Frustrante	7.3%
Retador	39.0%
Competitivo	24.4%
Aburrido	0.0%

Aunque muchas veces los jugadores fueron derrotados por la IA como se aprecia en la Tabla 1, los resultados muestran que las personas quedaban entusiasmadas y con ganas de volver a jugar, ilustrado en las Figs. 9, 10 y 11. Consideran que la IA no es imposible de vencer y a pesar de perder contra ella, tenían ganas de volver a competir.

Algo importante que se aprecia en la Fig. 11 es que el elemento de frustración que se busca evitar en videojuegos fue mínimo, e indica que el diseño del juego va por buen camino como se ilustra en las Tablas 2 y 3.



Fig. 10. Que tan difícil es vencer al CPU. Los jugadores describieron que la dificultad de la IA era entre media y difícil sin llegar a ser imposible de vencer.

5. Conclusiones y trabajo actual

Basado en los resultados obtenidos se ha generado una IA competitiva que permite a los jugadores disfrutar de la experiencia y querer volver a jugar, aunque sin duda para las personas el momento en el que se genera mayor presencia en el juego es cuando se compite contra otra persona.

No se puede reemplazar la socialización que existe y es parte de la naturaleza de la competencia humana, sin embargo, el NPC creado es suficiente y cumple su propósito.

¿Cómo alterar la dificultad del NPC? Actualmente se está trabajando en este aspecto. Ideas como aumentar la velocidad de movimiento y reacción de la IA han surgido, pero no son viables porque los jugadores empiezan a sentir frustración al ver que la computadora tiene reacciones más rápidas que las de un humano. Sienten que el NPC está haciendo trampa.



Fig. 11. Gráfica de radar mostrando la descripción que los jugadores le daban al CPU.

La solución que se está explorando es reemplazar el sistema de toma de decisiones actual (FSM) por un sistema de Planeación de Acciones Basadas en Metas (GOAP),

que elimina la restricción que tienen las máquinas de estado de que su comportamiento está limitado y puede ser predicho con mayor facilidad.

También se está trabajando para integrar potenciadores (power-ups) al juego que lo hacen más interesante y desafiante. La IA deberá tomar en cuenta los tipos de potenciadores, cuando tomar alguno, y cuando usarlo. Diseñar al NPC con estas circunstancias presenta un reto interesante.

Referencias

- 1. Stuart, J., Peter, N.: Artificial Intelligence: A Modern Approach. 3rd ed. Prentice Hall, United States of America (2010)
- 2. Mat, B.: Programming game AI by example. 1st ed. Jones & Bartlett Learning, United States (2004)
- 3. Bevilacqua, F.: Understanding Steering Behaviors: Pursuit and Evade (2018)
- Craig, R.: Steering Behaviors for Autonomous Characters. In: Proceedings of Game Developers Conference 1999, pp. 763–782. Miller Freeman Game Group, San Francisco California (1999)
- Heri, A., Agung, H.: Kuspriyanto: Design and Implementation of Zoopedia: Behaviour of Non Playable Character (NPC) of Tiger Hunting the Prey. Procedia - Social and Behavioral Sciences 67, pp. 196–202 (2012)
- 6. Robert, N.: Game Programming Patterns. 1st ed. Genever Benning, United States (2014)
- Eric, L.: Mathematics for 3D Game Programming and Computer Graphics. 3rd ed. Cengage Learning Ptr, United States (2011)
- 8. Fletcher, D., Ian, P.: 3D Math Primer for Graphics and Game Development. 2nd ed. A K Peters Ltd, United States (2011)
- Ray, B., Aung, S., Thet, N.: Unity 2017 Game AI Programming. 3rd ed. Packt Publishing, United States (2018)

Propuesta de algoritmo GRASP para el problema de la tardanza total ponderada en una máquina

Rosmery-Caridad Acosta-Núñez, Guillermo-Airam Piloto-Cuellar, Ricardo Hernández-Zitlalpopoca, Santiago-Omar Caballero-Morales

UPAEP, Puebla, México

{rosmerycaridad.acosta, guillermoairam.piloto, ricardo.hernandez02, santiagoomar.caballero}@upaep.mx

Resumen. En este artículo se aborda el problema de la tardanza total ponderada en una máquina, conocido como Single Machine Total Weighted Tardiness (SMTWT, por sus siglas en inglés), reconocido en la literatura como un problema de tipo NP-duro debido a su complejidad computacional. Para su solución se hace una propuesta de algoritmo Greedy Randomized Adaptive Search Procedure (GRASP) con una estrategia de integración de los operadores de diversificación 2-opt e inversión, cuyo objetivo es la minimización de costos a través de encontrar un orden efectivo de las tareas abordadas en cada máquina. Se utilizó la biblioteca OR-Library con problemas de 40, 50 y 100 tareas con 125 instancias cada uno. El algoritmo propuesto obtuvo, para el total de casos analizados, valores que en promedio tenían un 6.5% de error respecto a los mejores valores conocidos y en aproximadamente el 15% se encontraron mejores soluciones que aquellas reportadas en la literatura.

Palabras clave: búsqueda local, GRASP, inversión, intercambio, SMTWT.

Grasp Algorithm Proposed for the Problem of the Single Machine Total Weighted Tardiness

Abstract. This article addresses the problem of total weighted delay in a machine, known as Single Machine Total Weighted Tardiness (SMTWT) for its acronym in English, recognized in the literature as a NP-hard problem due to its computational complexity. For its solution, a Greedy Randomized Adaptive Search Procedure (GRASP) algorithm is proposed with an integration strategy of 2-opt and inversion diversification operators, whose objective is to minimize costs through finding an effective order of tasks addressed in each machine. We used the OR-Library library with problems of 40, 50 and 100 tasks with 125 instances each. The proposed algorithm obtained for the total of analyzed cases values that on average had a 6.5% error with respect to the best known values and in approximately 15% of the cases better solutions were found than those reported in the literature.

Keywords: local search, GRASP, SMTWT, inversion, 2-Opt.

1. Introducción

La programación de trabajos comenzó a tomarse en serio en los procesos de producción a principios de este siglo con el trabajo de Henry Gantt y otros pioneros. Consiste en un proceso de toma de decisiones que se utiliza de forma regular en muchas industrias de producción y servicios, sobre el principio de la asignación de recursos a tareas en períodos de tiempo determinados y su propósito es optimizar uno o más objetivos, como mínimo tiempo, mayor beneficio, mayor utilización, entre otros. La programación, como proceso de toma de decisiones, juega un papel importante en la mayoría de los sistemas de fabricación y producción, así como en la mayoría de los entornos de procesamiento de la información. También es importante en entornos de transporte y distribución y en otros tipos de industrias de servicios [1].

Según Davoudpour [2], la planificación o diseño de la producción es un problema de decisiones a lo largo de la cadena de suministro. Esta planificación en la logística, implica la asignación de tareas o servicios para optimizar procesos en un área manufacturera.

La agrupación de estas tareas es necesaria dadas ciertas cuestiones tecnológicas y procesos de capacidad: una vez que se tenga la asignación de una familia de productos por máquina, se puede gestionar la asignación de tareas para cada una de las máquinas y obtener un sistema de producción homogenizado [3]. En otras palabras, el objetivo es completar una serie de trabajos en una sola máquina tan cercana como lo sea la fecha de entrega del cliente, la minimización de la tardanza es un concepto de suma importancia en la industria y sistemas de manufactura esbelta conocido como justo a tiempo, Just-in-Time (JIT, por sus siglas en inglés) [4].

En este artículo se aborda el problema de ordenamiento de tareas en una sola máquina, con el objetivo de minimizar la tardanza total ponderada. Para esto se analizaron tres tipos de problemas de secuenciación: de 40, 50 y 100 trabajos. Cada tipo de problema está compuesto de 125 instancias (diferentes ejemplos de problemas con el número de trabajos especificado) y fueron obtenidos de la biblioteca digital OR-Library. Con el fin de encontrar la secuencia que minimice costos, se implementó el algoritmo de búsqueda local GRASP con una estrategia de aplicación de los operadores 2-opt e inversión.

Los avances del artículo se presentan de la siguiente manera: en la Sección 2 se presenta una revisión bibliográfica sobre los antecedentes del problema abordado; en la Sección 3 se desarrolla un análisis del método de resolución aplicado, sucedida por la Sección 4, donde se muestran los resultados obtenidos; finalmente, en la Sección 5 se presentan las conclusiones referentes al análisis de los aportes de la investigación.

2. Estado de arte

El problema de una máquina ha sido el más popular de todos los problemas de programación. Existe una amplia variedad de resultados para un amplio conjunto de especificaciones de problemas. Dentro de estos resultados se encuentran los procedimientos para minimizar el tiempo medio de flujo [5] y para minimizar la

tardanza máxima [6]; ambos procedimientos determinan la secuencia de tareas óptima por medio de un orden simple de las tareas. Un procedimiento un poco más complejo es el de Moore [7], para minimizar el número de tareas tardías; aquí las tareas se ordenan, primero, de acuerdo con los tiempos de finalización deseados y, luego, esta secuencia se modifica mediante la eliminación secuencial de tareas tardías. Si bien los problemas antes mencionados han demostrado ser bastante fáciles, el problema de minimizar la tardanza ponderada ha sido considerablemente más difícil [8].

Pinedo [1] expone varias razones de por qué los modelos de una sola máquina son importantes. Estos modelos a menudo tienen propiedades que ni las máquinas en paralelo ni las máquinas en serie tienen, por lo cual, los resultados que se pueden obtener no sólo proporcionan información sobre su entorno, sino que, además, suministran una base para la heurística que se aplica a entornos de máquinas más complicados. En la práctica, los problemas de programación para estos entornos a menudo se descomponen en sub-problemas que se ocupan de máquinas individuales.

La programación de producción se puede definir como la asignación de recursos de producción disponibles a lo largo del tiempo para satisfacer mejor un conjunto de criterios. Típicamente, el problema de programación implica un conjunto de tareas a realizar, y los criterios pueden involucrar compensaciones entre la finalización temprana y tardía de una tarea, y mantener el inventario para la tarea y frecuentes cambios de producción [8].

Diversos estudios han abordado el problema de la tardanza. Karp [9] demostró que este problema es NP-completo. Srinivasan [10] determinó relaciones de precedencia entre pares de trabajos, lo cual planteó una serie de proposiciones que redujeron considerablemente el cálculo total involucrado en la fase de programación dinámica del método de solución. Igualmente, Fisher [11] presentó un algoritmo Branch & Bound para el problema de la programación de trabajos en una sola máquina para minimizar las tardanzas. El principal resultado teórico de su trabajo fue un algoritmo que resuelve el problema de Lagrange en una serie de pasos proporcionales al producto de n2 y el tiempo promedio de procesamiento del trabajo.

Asimismo, Rinnooy Kan *et al.* [12], Picard y Queyranne [13] y Baker y Schrage [14] estudiaron el problema de la tardanza de una máquina o una versión generalizada de la misma. Schrage y Baker [15] implementaron un algoritmo de programación dinámica para resolver problemas de secuenciación de una máquina con restricciones de precedencia. Dicho algoritmo fue mucho más eficiente que los anteriores para ciertos problemas de secuenciación de una máquina. El trabajo de Graves [8] ofrece un grupo de observaciones sobre el estado de la práctica de la programación de producción, estableciendo una perspectiva con la cual contrasta la teoría de programación.

El Single Machine Total Weighted Tardiness o SMTWT, que se conoce como $1||\sum w_j T_j$ según Graham *et al.* [16], se puede identificar como NP-duro. Lo que significa que se requiere de un alto esfuerzo computacional para ejecutar un algoritmo de optimización combinatoria, mientras que otros problemas, como la programación dinámica o la ramificación, tienen una complejidad cuadrática [21].

Bozejko, Grabowski y Wodecki [18] también han abordado el problema de la tardanza ponderada, donde proponen un algoritmo de búsqueda tabú con un

71

vecindario específico que emplea bloques de trabajos y una técnica de movimientos compuestos a partir de algunas propiedades nuevas del problema asociado con los bloques.

Otra revisión donde se aborda la tardanza de una sola máquina a partir de un algoritmo de búsqueda tabú junto a otros meta-heurísticos es Liao y Cheng [19]. Asimismo, Navarrete *et al.* [20], Caballeros y Alvarado [21], Vega y Caballeros [22], abordan la minimización de la tardanza total ponderada en un entorno de producción mediante el meta-heurístico de Procedimientos de Búsqueda Voraces Aleatorizados y Adaptativos o Greedy Randomize Adaptive Search Procedure (GRASP, por sus siglas en ingles). Este meta-heurístico es constructivo y determinístico dado que genera una solución completa calculando prioridades para cada uno de los trabajos a ser asignados.

Al añadir aleatoriedad en la selección de las tareas para su asignación se da la oportunidad de asignar trabajos diferentes a aquellos determinados mediante el proceso determinístico, lo cual puede conllevar a una mejor asignación global [23].

Otras de las posibles metodologías estudiadas para la solución del problema de la tardanza total ponderada en una máquina son, el procedimiento de la búsqueda vecina cercana [24], la secuencia inicial en dos pasos [24], el algoritmo de recocido simulado [25], algoritmos voraces [26], y algoritmos genéticos [27].

Una vez seleccionado el método de solución a utilizar en un problema de optimización, vale la pena comprobar su comportamiento en diferentes escenarios. La prueba *t* de dos muestras se puede utilizar para comparar los promedios entre grupos y determinar si existe alguna diferencia estadística significativa entre ellos. Esto para descartar diferencias causadas por eventos aleatorios y no por la eficiencia de un método con respecto a otro.

La prueba de hipótesis, es una prueba de análisis multi-variable en dos vectores de medias μ_1 y μ_2 , basado en dos muestras aleatorias independientes, una con distribución de medias μ_1 y otra con una distribución de medias μ_2 . Esta prueba es utilizada en diferentes aplicaciones científicas [28]. La clásica formulación de hipótesis se expresa como:

```
Hipótesis Nula Ho: \mu_1 - \mu_2 = 0,
Hipótesis Alternativa Ha: \mu_1 - \mu_2 \neq 0.
```

Para determinar si la diferencia entre las muestras de medias es estadísticamente significativa, es posible utilizar la comparativa de valor p con respecto al nivel de significancia α que nos indica el nivel de riesgo que existe en que realmente no haya diferencia. Es posible hacer las siguientes aseveraciones para el análisis de las muestras en una prueba de hipótesis [28]:

- $p \le \alpha$: La diferencia entre las medias es estadísticamente significativa Si p es menor o igual que el nivel de significancia, se rechaza la hipótesis nula, y se puede concluir que existen diferencias entre las medias de la población.
- $p > \alpha$: La diferencia entre las medias no es estadísticamente significativa Si p es mayor al nivel de significancia, la decisión es no rechazar la hipótesis nula, pues no hay evidencia suficiente para concluir que la diferencia entre las medias de las muestras, sean estadísticamente significantes.

3. Metodología aplicada

Búsqueda local es uno de los algoritmos más utilizados en el problema SMTWT encontrados en la literatura. A continuación, se explica en qué consiste el problema de SMTWT, el algoritmo GRASP en su forma estándar y el seudocódigo del GRASP propuesto en el presente artículo.

3.1. SMTWT

Según Abdul-Razaq *et al.* [29] el problema de SMTWT puede expresarse formalmente de la siguiente forma: cada uno de los n trabajos (numerados 1, ..., n) debe procesarse sin interrupción en una sola máquina que pueda manejar sólo un trabajo a la vez. Para un mejor entendimiento del problema, se expone un ejemplo dónde n=6, en la Tabla 1 se muestra la secuencia inicial de las tareas para el análisis.

Tabla 1. Secuencia de tareas para una máquina.

1	2	3	4	5	6

El trabajo i (i = 1, ..., n) queda disponible para el procesamiento en el tiempo cero. Esto significa que todas las tareas pueden ser programadas para comenzar la secuencia y por lo tanto no hay dependencia entre ellas. Cada tarea requiere un tiempo de procesamiento entero p_i , un peso positivo w_i y una fecha de vencimiento o de entrega d_i . Estos datos se presentan en la Tabla 2 para el análisis del ejemplo.

Tabla 2. Datos para el ejemplo analizado.

Tarea (i)	p_i	W_i	d_i
1	5	1	5
2	6	4	10
3	8	3	15
4	4	4	9
5	7	2	20
6	9	1	30

Para un orden de procesamiento de los trabajos, para el ejemplo se utilizará la secuencia expuesta en la Tabla 1, el tiempo de finalización (más temprano) se puede calcular como $C_i = P_i + C_{i-1}$ y la tardanza $T_i = max\{C_i - d_i, 0\}$ de la tarea $i(i=1,\ldots,n)$. En la Tabla 3 se muestran los resultados para C_i y T_i .

Tabla 3. Tiempo de finalización más temprano y tardanza del ejemplo.

Tarea (i)	p_i	C_i	d_i	T_i
1	5	5	5	0
2	6	11	10	1
3	8	19	15	4
4	4	23	9	14
5	7	30	20	10
6	9	39	30	9

El objetivo de este problema es encontrar una secuencia de programación de las tareas que minimize la tardanza total $\sum_{i=1}^{n} w_i * T_i$, que en este ejemplo sería de 101 unidades, en la Tabla 4 se muestran los cálculos realizados para determinar dicho valor.

Tabla 4. Cálculo de la tardanza total	ponderada para el	ejemplo.
---------------------------------------	-------------------	----------

Tarea (i)	W_i	T_i	$w_i \times T_i$
1	1	0	0
2	4	1	4
3	3	4	12
4	4	14	56
5	2	10	20
6	1	9	9
n=6			101
$\sum_{i=1}^{w}$	$i \times T_i$		

Como se observa en la Tabla 4 para el ejemplo se tienen cinco tareas con tardanza, representando el 84% del total de las tareas a programar, por lo que es necesario encontrar otra secuencia de tareas que minimice la cantidad de trabajos con entrega tardía y a la vez la tardanza total ponderada.

3.2. GRASP

GRASP es una meta-heurística de búsqueda local de inicio múltiple, conformada por dos fases: a) en la primera se desarrolla un algoritmo GREEDY para encontrar aleatoriamente una solución no óptima, pero si factible al problema; b) en la segunda fase de realiza una búsqueda local y se comparan los beneficios de las nuevas soluciones encontradas con la solución histórica, y en caso de que sea mejor que esta última se procede a actualizar la histórica. A continuación, se observa el seudocódigo de un procedimiento general del GRASP propuesto por Resende y Ribeiro [30].

Procedure GRASP;

```
f^* \leftarrow \infty;
1
2
       for k = 1, ..., Max. Iterations do
3
                  Construct a greedy randomized x \in X;
                  Find y by applying local search to x;
5
                  if f(y) < f^* do;
                    x^* \leftarrow y;
6
                     f^* \leftarrow f(x^*);
8
                  end if
9
         end for
       return x*
end GRASP;
```

3.3. GRASP propuesto

En el artículo se evalúa un conjunto de instancias mediante el meta-heurístico GRASP, en el cual se integraron de manera estratégica los heurísticos 2-opt e inversión. A continuación, se presenta el seudocódigo del algoritmo desarrollado para la solución de los problemas de SMTWT abordados:

Procedure GRASP modified;

```
Read Input;
2
     x^* \leftarrow Construct \ a \ greedy \ randomized \ x \in X;
3
     f^* \leftarrow f(x^*);
4
     for k = 1, ..., Max. Iterations do
5
         Construct a 2 – opt randomized y \in Y;
         Construct an reverse randomized z \in Z;
6
         Find S by applying local search to y and z;
7
8
         if f(S) < f^* do;
            x^* \leftarrow S;
9
            f^* \leftarrow f(x^*);
10
         end if
11
12
     end for
13 return x^*
end GRASP modified;
```

Operadores de diversificación 2-opt e inversión. El operador de intercambio o 2-opt es un algoritmo iterativo de mejora. Este procede bajo una prueba sistemática, dependiendo si la secuencia actual puede ser mejorada intercambiando 2 tareas; para el 2-opt se requiere un tiempo de O(n2), por lo que se vuelve muy complejo cuando el número de tareas crece, por lo que el esfuerzo computacional es muy alto. En la Figura 1 se muestra cómo funciona el 2-opt tomando el ejemplo anterior para el problema de SMTWT en donde se seleccionan aleatoriamente dos posiciones.



Fig. 1. Aplicación del 2-opt: a) Secuencia actual, b) Secuencia resultante.

Adicionalmente, se utilizó el operador de inversión, el cual es un algoritmo de optimización discreto continuo para coordinar los valores de tareas o nodos asignados en un problema, se toma un número de tareas, y se invierten de adelante hacia atrás, con el objetivo de encontrar un resultado mejor al anterior [31]. En la Figura 2 se muestra cómo funciona el operador de inversión tomando el ejemplo anterior para el problema de SMTWT en donde se seleccionan aleatoriamente dos posiciones para determinar el rango de tareas a las que se va a aplicar dicho operador. En este caso las

posiciones seleccionadas son 2 y 5 por lo que se realiza la inversión de las tareas comprendidas entre estas dos posiciones.



Fig. 2. Aplicación del operador inversión: a) Secuencia actual, b) Secuencia resultante

En el algoritmo diseñado en el presente artículo se utilizan estos dos operadores para realizar la generación de nuevas soluciones a partir de una solución inicial, con la diferencia de que, al aplicar el operador de inversión a una serie de tareas, estas se extraen de la secuencia actual, se invierten y se insertan al final de la cadena no en el lugar dónde se encontraba inicialmente, para el ejemplo analizado anteriormente quedaría como se muestra en la Figura 3.



Fig. 3. Aplicación del operador inversión con inserción: a) Secuencia actual, b) Secuencia resultante

Estos dos heurísticos se gestionaron a través del algoritmo GRASP, dónde en cada iteración se ejecutan ambos operadores y se selecciona la solución que mejor costo arroje una vez evaluada la función objetivo y en caso de que la seleccionada sea mejor que la histórica, se procede a su actualización, repitiendo dicho proceso hasta el máximo de iteraciones definidas.

4. Análisis de resultados

Se aplicó el algoritmo GRASP desarrollado en la sección anterior para la solución del problema SMTWT utilizando las instancias de J. E. Beasley de 40, 50 y 100 trabajos, conformado cada uno por 125 instancias. Estas se obtuvieron a través de la base de datos OR Library [32] la cual contiene un amplio número de problemas de investigación de operaciones que sirven de referencia para la comparación de algoritmos y técnicas de optimización a nivel mundial. Los datos para los problemas seleccionados se generaron aleatoriamente basándose en una distribución uniforme con valores entre 1 y 100 para los tiempos de procesamiento p_i , entre 1 y 10 para los pesos w_i y con una combinación de dos distribuciones uniformes para las fechas de entrega d_i . Dicha biblioteca cuenta actualmente con las soluciones óptimas para 124 de 125 instancias de 40 trabajos y 115 de 125 para 50 trabajos, así como las mejores soluciones encontradas para la instancia 19 de 40 trabajos, para los 10 restantes de 50 trabajos y para las 125 instancias de 100 trabajos.

Para la implementación de la meta-heurística seleccionada se trabajó en el software Matlab versión 2015a. Una vez obtenidos los resultados se compararon con los

óptimos y mejores conocidos reportados en la biblioteca, calculándose el error por cada problema, y se determinó que los mejores resultados fueron obtenidos mediante la combinación de los operadores de inversión aleatoria con inserción e intercambio aleatorio, además el tiempo de ejecución computacional para los tres tipos de problema fue en promedio de 33.39 minutos.

En las Figs. 4, 5 y 6 se observa la desviación porcentual de los valores obtenidos respecto a los valores óptimos o mejores conocidos de las tardanzas ponderadas para las 125 instancias en cada tipo de problema.

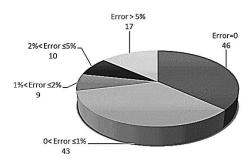


Fig. 4. Error obtenido por el algoritmo GRASP para las 125 instancias de 40 trabajos.

Como se observa en la Figura 4 el algoritmo desarrollado encontró los valores óptimos en 46 instancias que representan el 36.8% de las 125 analizadas, y en el 86.4% la desviación con respecto a los mejores valores reportados en la librería es menor al 5%. Por ende, de las 125 instancias de 40 trabajos analizadas inicialmente sólo el 13.6% tuvo una desviación superior al 5%. Siendo la desviación promedio de las 125 instancias de un 3.07% aproximadamente.

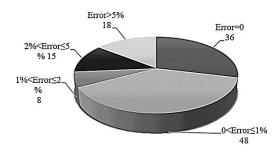


Fig. 5. Error obtenido por el algoritmo GRASP para las 125 instancias de 50 trabajos.

En la Fig. 5 se muestra que para las 125 instancias de 50 trabajos el método GRASP encontró 36 valores óptimos que representan el 28.8% de las 125 analizadas y que el 85.6% de los resultados obtenidos tienen una desviación menor al 5%, siendo la desviación promedio de las 125 instancias de un 2.84% aproximadamente.

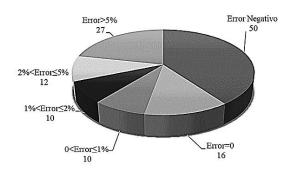


Fig. 6. Error obtenido por el algoritmo GRASP para las 125 instancias de 100 trabajos.

En el caso de las instancias de 100 trabajos no se tiene referencia de los valores óptimos, por lo que la desviación se corresponde con los mejores valores reportados hasta la actualidad en la base de datos. como se muestra en la Figura 6 de las 125 instancias analizadas con el método GRASP se encontró 50 soluciones para las cuales la evaluación en la función objetivo es mejor que el mejor conocido, representando un 40% de las 125 analizadas. Además, un 12.8% se corresponde con los mejores valores conocidos; y el 78.4% de los resultados obtenidos tienen una desviación menor al 5%, siendo la desviación promedio de las 125 instancias de un 13.74% aproximadamente.

Dados los resultados obtenidos, se trabajó con un comparativo de errores para identificar si éstos se comportan de forma similar. Para ello, se separaron los errores en tres poblaciones diferentes, errores de los problemas de 40, 50 y 100 tareas; el objetivo es aceptar o rechazar la idea que el meta-heurístico tiene la misma eficiencia en el total de instancias. De esta manera, con base a los datos de la población de los errores, se trabajará inicialmente con una prueba de hipótesis de igualdad y diferencia, analizando los datos opuestos sobre una población, una hipótesis nula y una hipótesis alternativa, la primera verificará si los datos de la población son iguales, y la alterna si los datos de la población son diferentes, el estadístico utilizado es la prueba "t" en un análisis de dos medias, aplicado para comparar si el promedio de dos poblaciones son significativamente diferentes. En base a ello, determinaremos la respuesta utilizando p value y se aceptará o rechazara la hipótesis con respecto al nivel de significancia: si p value es menor que el nivel de significancia $\alpha = 95\%$ entonces se rechazará la hipótesis nula. A continuación, se muestran las poblaciones utilizadas para el análisis de significancia estadística:

- Errores de las instancias de 40 tareas = μ_1 ,
- Errores de las instancias de 50 tareas = μ_2 ,
- Errores de las instancias de 100 tareas = μ_3 .

4.1. Pruebas de Significancia Estadística: Comparación de Medias

Primera prueba:

Ho: $\mu_1 - \mu_2 = 0$,

Ho: $\mu_1 - \mu_2 \neq 0$.

Resultados estadísticos de la primera prueba de hipótesis (resultados obtenidos mediante Minitab).

```
Two-Sample T-Test and CI: 40 tareas, 50 tareas

Two-sample T for 40 tareas vs 50 tareas

N Mean StDev SE Mean

40 tareas 125 0.0307 0.0858 0.0077

50 tareas 125 0.0284 0.0700 0.0063

Difference = \mu (40 tareas) - \mu (50 tareas)

Estimate for difference: 0.00238

95% CI for difference: (-0.01713, 0.02188)

T-Test of difference = 0 (vs \neq): T-Value = 0.24 P-Value = 0.810

DF = 238
```

Segunda prueba:

```
Ho: \mu_1 - \mu_3 = 0,
Ho: \mu_1 - \mu_3 \neq 0.
```

Resultados estadísticos de la segunda prueba de hipótesis (resultados obtenidos mediante Minitab).

```
Two-Sample T-Test and CI: 40 tareas, 100 tareas
Two-sample T for 40 tareas vs 100 tareas
             Ν
                  Mean
                         StDev SE Mean
          125 0.0307 0.0858
40 tareas
                                0.0077
100 tareas 125
                 0.137
                        0.826
                                  0.074
Difference = \mu (40 tareas) - \mu (100 tareas)
Estimate for difference:
                         -0.1067
95% CI for difference: (-0.2536, 0.0403)
T-Test of difference = 0 (vs \neq): T-Value = -1.44 P-Value =0.153
```

En ambos casos, podemos observar que el *p-value* es mayor al nivel de significancia, por lo que estadísticamente podemos asegurar que los errores se comportan de manera similar entre los diferentes problemas: 40, 50 y 100 tareas, con mucha más similitud entre las instancias de 40 vs. 50 tareas, y con una menor similitud entre las instancias de 40 vs. 100. De esta manera se puede concluir que no hay evidencia suficiente para afirmar que existe una diferencia en el comportamiento de los resultados obtenidos por el algoritmo propuesto para los diferentes tipos de problema. Por lo tanto, el desempeño del algoritmo GRASP propuesto es consistente y robusto a través de problemas de asignaciones de diferentes tamaños.

5. Conclusiones y trabajo futuro

En este artículo se trabajó con una técnica de rápida implementación para encontrar una buena solución al problema de SMTWT. Considerando que este tipo de problemas se le considera NP-duro, se utilizó la integración de un meta-heurístico GRASP con los operadores 2-opt e inversión, y se compararon los resultados con los mejores valores conocidos de las diferentes instancias de los problemas. Este análisis y desarrollo de la solución presenta una alternativa competitiva y ofrece resultados rápidos para la secuenciación de tareas en máquinas de producción en línea. La principal contribución del trabajo se puede ver como el beneficio de sólo integrar dos heurísticos (2-opt e inversión) en un proceso de selección cerrada.

La calidad de los resultados es competitiva, se trabajó con 125 instancias para cada tipo de problema de 40, 50 y 100 tareas, haciendo un total de 375 instancias. En promedio, se obtuvo resultados con un porcentaje de error menor al 5% para alrededor del 83.47% (313 de 375). El algoritmo propuesto reportó resultados eficientes dado el error promedio de 6.5%, y, además, para 50 instancias de 100 tareas reportó mejores resultados que los conocidos.

El tiempo de procesamiento del algoritmo fue corto (33.39 minutos en promedio), en el cual se encontraron estos resultados, por lo que es posible encontrar valores óptimos y cercanos al óptimo en un tiempo razonable. Es importante mencionar que el tiempo de ejecución para los 3 problemas; 40, 50 y 100 tareas no es lineal.

El comportamiento del meta-heurístico desarrollado es adecuado. Como se pudo observar en el estadístico de prueba, no hay evidencia suficiente para determinar que los errores tienen una dispersión no-consistente en los problemas de 40, 50 y 100 tarea. Si bien, *p-value* no identificó comportamiento de dispersión, si se pudo observar que en el problema de 100 tareas la variación en los errores fue mayor, unos por arriba de lo esperado y otros mejor que los actuales conocidos.

Para trabajos futuros se sugiere comprobar con otros meta-heurísticos y operadores de diversificación, pues no se descarta la posibilidad de poder encontrar mejores resultados que los registrados por el algoritmo de búsqueda local propuesto para el problema SMTWT.

Referencias

- 1. Pinedo, M. L.: Scheduling: theory, algorithms, and systems. 3rd ed. Springer, NY (2008)
- Davoudpour, G. R.: Coordinated Location, Distribution and Inventory Decisions in Supply Chain. Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran (2015)
- 3. Schaller, J.: Scheduling a permutation flow shop with family setups to minimize total tardiness. International Journal of Production Research 50(8), pp. 2204–2217 (2012)
- Rabadi, G., Moraga, R., Al-Salem, A.: Heuristics for the unrelated parallel machine scheduling problem with setup times. Journal of Intelligent Manufacturing 17(1), pp. 85–97 (2006)
- Smith, W. E.: Various optimizers for single stage production. Naval Research Logistics (NRL) 3(1-2), pp. 59–66 (1956)

- Jackson, J. R.: Scheduling a Production Line to Minimize Maximum Tardiness. Research Report 43, Management Sciences Research Project, University of California, Los Angeles (1955)
- 7. Moore, J. M.: An n-Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs. Management Science 15(1), pp. 102–109 (1968)
- 8. Graves, S. C.: A review of production scheduling. Operations research 29(4), pp. 646–675 (1981)
- 9. Karp, R. M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W., Bohlinger J.D. (eds) Complexity of Computer Computations. The IBM Research Symposia Series, pp 85–103. Springer, Boston, MA (1972)
- 10. Srinivasan, V.: A hybrid algorithm for the one machine sequencing problem to minimize total tardiness. Naval Research Logistics (NRL) 18(3), pp. 317–327 (1971)
- 11. Fisher, M. L.: A dual algorithm for the one-machine scheduling problem. Mathematical programming 11(1), pp. 229–251 (1976)
- 12. Rinnooy Kan, A. H. G., Lageweg, B. J., Lenstra, J. K.: Minimizing total costs in one-machine scheduling. Operations Research 23(5), pp. 908–927 (1975)
- 13. Picard, J. C., Queyranne, M.: The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. Operations Research 26(1), pp. 86–110 (1978)
- 14. Baker, K. R., Schrage, L. E.: Finding an optimal sequence by dynamic programming: an extension to precedence-related tasks. Operations Research 26(1), pp. 111–120 (1978)
- 15. Schrage, L., Baker, K. R.: Dynamic programming solution of sequencing problems with precedence constraints. Operations research 26(3), pp. 444–449 (1978)
- Graham, R. L., Lawler, E. L., Lenstra, J. K., Kan, A. R.: Optimization and approximation in deterministic sequencing and scheduling: a survey. In: P.L., Hammer, E.L. Johnson, B.H. Korte, (eds) Annals of Discrete Mathematics, vol. 5, pp. 287–326. Elsevier (1979)
- 17. Bożejko, W., Grabowski, J., Wodecki, M.: Block approach tabu search algorithm for single machine total weighted tardiness problem. Computers & Industrial Engineering 50(1–2), pp. 1–14 (2006)
- 18. Liao, C. J., Cheng, C. C.: A variable neighborhood search for minimizing single machine weighted earliness and tardiness with common due date. Computers & Industrial Engineering 52(4), pp. 404–413 (2007)
- 19. Niño-Navarrete, Á. M., Caballero-Villalobos, J. P.: Evaluación de funciones de utilidad de GRASP en la programación de producción para minimizar la tardanza total ponderada en una máquina. Ingeniería 14(2), pp. 51–58 (2009)
- 20. Caballero-Villalobos, J.P., Alvarado-Valencia, J.A.: Greedy Randomized Adaptive Search Procedure (GRASP), una alternativa valiosa en la minimización de la tardanza total ponderada en una máquina. Ingeniería y Universidad 14(2), 275–295 (2010)
- 21. Vega-Mejía, C. A., Caballero-Villalobos, J. P.: Uso combinado de GRASP y Path-Relinking en la programación de producción para minimizar la tardanza total ponderada en una máquina. Ingeniería y Universidad 14(1), 79–96 (2010)
- 22. Moreira, J. M.: Greedy randomized dispatching heuristics for the single machine scheduling problem with quadratic earliness and tardiness penalties. The International Journal of Advanced Manufacturing Technology 44(9–10), pp. 995–1009 (2009)
- 23. Kim, Y.: A new branch and bound algorithm for minimizing mean tardiness in two-machine flowshops. Computers & Operations Research 20(4), pp. 391–401 (1993)
- 24. Parthasarathy, S., Rajendran, C.: A simulated annealing heuristic for scheduling to minimize mean weighted tardiness in a flowshop with sequence-dependent setup times of jobs – A case study. Production Planning and Control 8, pp. 475–483 (1997)
- 25. Framinan, J. M., Leisten, R.: Total tardiness minimization in permutation flowshops: a simple approach based on a variable greedy algorithm. International Journal of Production Research 46(22), pp. 6479–6498 (2008)

- 26. Vallada, E., Ruiz, R., Minella, G.: Minimizing total tardiness in the m-machine flowshop problem: A review and evaluation of heuristics and metaheuristics. Computers & Operations Research 35(4), p. 1350 (2008)
- 27. Liu, W., Tony, T., Xia, Y.: Two-sample test of high dimensional means under dependence. Journal of the Royal Statistical Society 76(2), pp. 349–372 (2014)
- 28. Abdul-Razaq, T. S., Potts, C. N., Van Wassenhove, L. N.: A survey of algorithms for the single machine total weighted tardiness scheduling problem. Discrete Applied Mathematics 26(2–3), pp. 235–253 (1990)
- 29. Resende, M. G., Ribeiro, C. C.: A GRASP with path relinking for private virtual circuit routing. Networks 41(2), pp. 104–114 (2003)
- 30. Mahinthakumar, K., Sayeed, M.: Hybrid Genetic Algorithm Local Search Methods for Solving Groundwater Source Identification Inverse Problem. Journal of Water Resources Planning and Management 131(1) (2005)
- 31. OR-Library, http://people.brunel.ac.uk/~mastjjb/jeb/orlib/wtinfo.html (2018)

Una comparativa de diferentes parámetros en la construcción evolutiva de descriptores para la clasificación de imágenes de texturas de piezas arqueológicas

Julio Antonio Jiménez Falcón¹, Manuel Ornelas-Rodríguez¹,

Marco Aurelio Sotelo-Figueroa², Pedro Antonio López García³, Alfonso Rojas Domínguez¹, Héctor José Puga Soberanes¹, Velino Esteves-Jiménez²

- $^{1}\,$ Tecnológico Nacional de México Instituto Tecnológico de León, León, G
to., México
- ² DCEA-Universidad Guanajuato, Departamento de Estudios Organizacionales, Guanajuato, Gto., México
 - ³ Escuela Nacional de Antropología e Historia, Ciudad de México, México {julio.jimenez, manuel.ornelas}@itleon.edu.mx

Resumen. En el presente trabajo se muestran los resultados de los descriptores construidos mediante una metodología basada en Gramática Evolutiva. Se hace la comparación de dos parámetros de la propuesta, uno es el uso de los canales de color y la escala de grises en el proceso de construcción de los descriptores, lo cual se llevó a cabo utilizando dos gramáticas tipo Backus-Naur en el proceso evolutivo. La primera gramática usa los canales de color rojo, azul, verde y la escala de grises, mientras que la segunda gramática solo usa la escala de grises. El otro parámetro es la forma de representar los vectores de características de las imágenes procesadas con los descriptores construidos, comparando dos alternativas: el histograma y los estadísticos del histograma. Los descriptores construidos fueron aplicados a la clasificación de imágenes de texturas de piezas arqueológicas procedentes de la cultura egipcia. Se muestran los resultados obtenidos, los cuales fueron analizados estadísticamente.

Palabras clave: clasificación de imágenes, construcción de descriptores, gramática evolutiva.

A Comparison between Different Parameters in the Evolutionary Construction of Descriptors for the Classification of Textures Images of Archaeological Pieces

Abstract. This work shows results about descriptors constructed by a methodology based in Grammatical Evolution. A comparison of two

parameters of this methodology was made. The first parameter is the use of the color and gray scale channels in the construction process of image descriptors. This was done through a comparison of two Backus-Naur grammars used in the evolutionary process, the first grammar used the red, blue, green and grayscale channels, and the second grammar only used the gray scale. The other parameter was the form to represent the feature vectors of the processed images, two alternatives was compared: the histogram and the histogram statistics. The descriptors constructed were applied to the classification of textures images of archaeological pieces from Egyptian culture. The results obtained are shown, which were analyzed statistically.

Keywords: image classification, feature construction, grammatical evolution.

1. Introducción

El análisis y clasificación de texturas representan importantes tareas en el área de visión por computadora. Para obtener resultados eficientes en el proceso de clasificación, es necesario tener descriptores que proporcionen un conjunto de características que representen adecuadamente a la base de datos de imágenes a tratar [1]. Comúnmente esta representación es hecha a mano por expertos en visión por computadora, [2,3,4,5,6], sin embargo, cuando no existe información a priori de la base de datos, la selección de los descriptores adecuados representa un problema. Para solucionar esta dificultad, se han propuesto diferentes alternativas basadas en algoritmos evolutivos, las cuales trabajan sobre la construcción o selección de descriptores de características [7,8,9,10].

En este trabajo, se presenta un método evolutivo para la construcción de descriptores de texturas usando Gramáticas Evolutivas (GE), a partir de la combinación de filtros pasa-baja, pasa-altas y direccionales, y operadores aritméticos. GE requiere una gramática formal, un motor de búsqueda y un proceso de mapeo. También se llevó a cabo una comparativa de dos parámetros, uno es el uso de los canales de color y escala de grises en el proceso de construcción del descriptor contra el uso de únicamente la escala de grises. Otro parámetro es la forma de representar los vectores de características en el proceso de la GE, donde se hizo la comparación entre el histograma y los estadísticos del histograma.

El trabajo está organizado de la siguiente manera: la sección 2 muestra un acercamiento a la GE; la sección 3 describe la metodología que se llevó a cabo; los resultados de la experimentación son reportados en la sección 4 y, por último, la sección 5 presenta las conclusiones.

2. Gramática evolutiva

La Gramática Evolutiva (GE) [11] está basada en el conocimiento de la Biología Molecular, como la Programación Genética (PG) [12], pero GE usa una gramática formal relacionada con el problema, un motor de búsqueda que se encarga del proceso evolutivo, y un algoritmo que realiza el proceso de mapeo [13]. GE usa una gramática libre de contexto tipo Backus-Naur (BNF) como gramática formal, la cual es expresada por la tupla {N,T,P,S}, donde N representa los símbolos no terminales, T son los símbolos terminales, P es el conjunto de reglas de producción y S es el símbolo de inicio.

El algoritmo Depth-First (DF) fue usado como proceso de mapeo. Un ejemplo de este proceso es mostrado en la figura 1. DF comienza con el símbolo de inicio y enseguida procede a sustituir el símbolo no terminal ubicado más a la izquierda con un regla de producción seleccionada por la ecuación (1),

$$Regla = c \% r, \tag{1}$$

donde c es un codon del genotipo, r es el número de reglas de producción válidas para el actual no terminal y % es el operador módulo. Este proceso continua hasta que el fenotipo generado no contenga más símbolos no terminales.

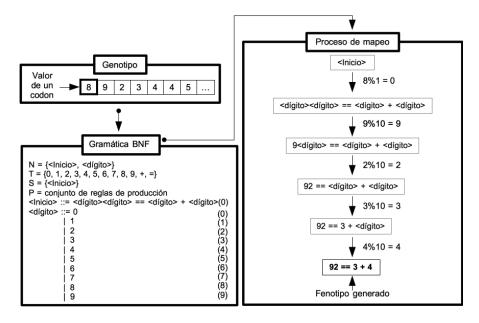


Fig. 1. Un ejemplo del proceso de mapeo de genotipo a fenotipo usando una gramática BNF y el algoritmo DF.

Para este ejemplo, el primer codon es c=8 y el símbolo inicial es <Inicio>, el cual tiene un número de regla de producción r=1. Aplicando la ecuación 1

se tiene 8%1=0, lo que conduce a seleccionar la regla de producción número 0, dada por <dígito><dígito>==<dígito>+<dígito>. Luego, el siguiente codon es c = 9 y el símbolo no terminal más a la izquierda es <dígito> con r = 10 reglas de producción válidas. De la ecuación 1 se obtiene 9%10=9, lo que nos lleva a sustituir el símbolo no terminal <dígito>por la regla de producción 9, que consiste en el símbolo terminal 9. El proceso de mapeo continua hasta que todos los símbolos no terminales han sido sustituidos por símbolos terminales.

Una vez que la población de genotipos es mapeada a su forma fenotípica, cada solución es evaluada por medio de la función objetivo. Estos valores de rendimiento son usados para evolucionar la población a través de un motor de búsqueda.

3. Metodología

Los experimentos realizados en esta investigación se basaron en el modelo de GE usado en [14], el cual es mostrado en la figura 2. En este trabajo, los módulos del modelo consisten en lo siguiente:

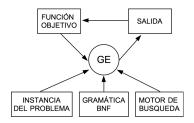
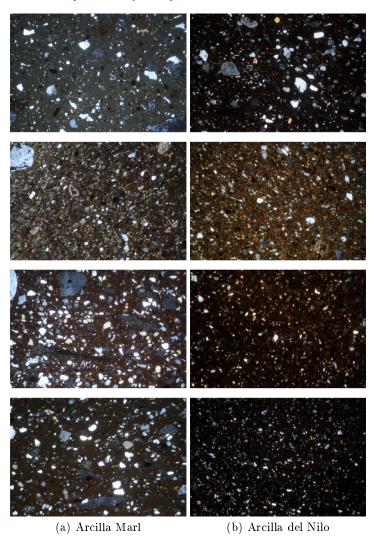


Fig. 2. Diagrama del modelo de GE utilizado en este trabajo [14].

Instancia del problema. Se usó la base de datos utilizada en el trabajo [15], la cual está formada por imágenes digitales de texturas de piezas arqueológicas de la cultura egipcia. Esta base de datos está dividida en dos clases: Arcilla del Nilo con 22 imágenes y Arcilla Marl con 47 imágenes. Cada imagen tiene una resolución original de 3900 x 2900 píxeles, sin embargo, para reducir el costo computacional se redimensionaron a 128 x 128 píxeles mediante interpolación bilineal. La figura 3 muestra ejemplos de las imágenes utilizadas en este experimento.

Gramática BNF. La gramática base utilizada en este trabajo es mostrada en la tabla 1, y sus símbolos terminales son descritos en la tabla 2. Dos variantes de la gramática base fueron usadas (BNF1 y BNF2), donde la diferencia entre ellas es el número de símbolos terminales validos para el símbolo no terminal <I>. BNF1 contiene los canales rojo, verde, azul y la escala de grises (los símbolos terminales Ir, Ig, Ib e Igray), y BNF2 usa únicamente la escala de grises. El trabajo [7] fue usado como base para el diseño de los símbolos terminales.

Una comparativa de diferentes parámetros en la construcción evolutiva de descriptores...



 ${f Fig. 3.}$ Ejemplos de las imágenes de texturas de las piezas arqueológicas usadas en este trabajo.

Motor de búsqueda. Para proporcionar significancia estadística a los resultados, 33 experimentos independientes fueron realizados con cada configuración. El algoritmo genético (GA por sus siglas en inglés) fue usado como motor de búsqueda, y su forma básica es mostrada en la figura 4. Los parámetros de GA fueron fijados con base a nuestra experiencia y sus valores fueron los siguientes: tamaño de población: 50 individuos genotípicos, longitud del cromosoma: 50 codones generados aleatoriamente entre [0,255], número de generaciones: 200, método de selección: 2 padres con torneo binario, método de

cruza: aleatorio a dos puntos, y método de mutación: mutación de un punto aleatorio de un sólo individuo por generación seleccionado aleatoriamente. En cada generación, la nueva población fue formada con 10 individuos seleccionados por elitismo, 15 nuevos individuos creados aleatoriamente y 25 generados por GA.

Tabla 1. Gramática BNF usada por GE.

```
N = \{ \langle Start \rangle, \langle MaxPooling \rangle, \langle Image \rangle, \langle Filtering 1 \rangle, \langle Filtering 2 \rangle, \langle I \rangle \}
T = {Maxpooling2, Maxpooling4, Maxpooling6, Maxpooling8, Maxpooling10,
        Gau1, Gau2, GauX, GauY, LoG1, LoG2, Lap, GBO-0, GBO-45, GBO-90,
       GB0-135, Aver, Med, EHIS, ABS, SQR, SQRT, LOG2, TIME-0.5, ADD,
       SUB, ABSsub, MULTI, DIV, Ir, Ig, Ib, Igray}
S = \{\langle Start \rangle\}
P = set of production rules
\langle \text{Start} \rangle ::= \langle \text{MaxPooling} \rangle
<MaxPooling> ::= Maxpooling2(<Filtering>) | Maxpooling4(<Filtering>)
                       | Maxpooling6(<Filtering>) | Maxpooling8(<Filtering>)
                       | Maxpooling10(<Filtering>)
<Filtering> ::= <Filtering1 > | <Filtering2 >
GauY(\langle Image \rangle) \mid LoG1(\langle Image \rangle) \mid LoG2(\langle Image \rangle)
                      Lap(\langle Image \rangle) \mid GBO-0(\langle Image \rangle) \mid GBO-45(\langle Image \rangle)
                      GBO-90(\langle Image \rangle) \mid GB0-135(\langle Image \rangle) \mid Aver(\langle Image \rangle)
                      Med(<Image>) \mid EHIS(<Image>) \mid ABS(<Image>)
                      SQR(\langle Image \rangle) \mid SQRT(\langle Image \rangle) \mid LOG2(\langle Image \rangle)
                     TIME-0.5(\langle Image \rangle)
\langle Filtering2 \rangle ::= ADD(\langle Image \rangle, \langle Image \rangle) \mid SUB(\langle Image \rangle, \langle Image \rangle)
                     ABSsub(<Image>,<Image>) | MULTI(<Image>,<Image>)
                     DIV(\langle Image \rangle, \langle Image \rangle)
<Image> ::= <Filtering1> \mid <Filtering2> \mid <I>
\langle I \rangle ::= Ir | Ig | Ib | Igray
```

Salida. Es la representación de cada una de las imágenes de la base de datos y consiste en un vector de características obtenido mediante alguna de las dos representaciones utilizadas en este trabajo: histograma (H) y estadísticos del histograma (EH), esté último formado por máxima probabilidad, energía, media, entropía, varianza, asimetría y r normalizada. Estas representaciones fueron obtenidas al aplicarse sobre las imágenes finales filtradas por los descriptores construidos por la GE. En total se obtuvieron 4 configuraciones, la gramática

Una comparativa de diferentes parámetros en la construcción evolutiva de descriptores...

```
Requiere: NúmeroDeIndividuos, NúmeroDeGeneraciones,
ContadorDeGeneraciones = 0;
Población = GenerarIndividuosAleatorios(NúmeroDeIndividuos);
CalcularAptitudPoblación(Población);
MIENTRAS ContadorDeGeneraciones < NúmeroDeGeneraciones HACER
    NuevaPoblación(NúmeroDeIndividuos); //Instanciar nuevo arreglo
    PARA i = 0, i < NúmeroDeIndividuos, i ++
        Padres = SeleccionarPadres(Población);
        NuevoIndividuo = MétodoDeCruza(Padres);
        NuevoIndividuo = MétodoDeMuta(NuevoIndividuo);
        NuevaPoblación(i) = NuevoIndividuo;
CalcularAptitudPoblación(NuevaPoblación);
Población = NuevaPoblación;
ContadorDeGeneraciones++;
```

Fig. 4. Algoritmo Genético básico.

BNF1 con H y EH (BNF1-H y BNF1-EH respectivamente), y la gramática BNF2 con la aplicación de las mismas dos representaciones (BNF2-H y BNF2-EH).

Función objetivo. Para evaluar la aptitud de cada solución generada, la ecuación (2) [16] fue usada como función objetivo:

$$error = 1 - exactitud,$$
 (2)

donde exactitud es el número de imágenes de prueba correctamente clasificadas dividido entre el total de imágenes de prueba. Se utilizó validación cruzada con 3 pliegues. Para cada uno de los 33 experimentos realizados con cada configuración, se hizo un muestreo aleatorio para formar los tres conjuntos de imágenes. La clasificación fue hecha por el algoritmo kNN, con $\mathbf{k}=3$. Finalmente se promedió el error de clasificación de cada uno de los pliegues para obtener la aptitud de cada individuo evaluado. El fenotipo con el mínimo valor de la función objetivo corresponde a la mejor solución generada.

4. Resultados

La tabla 3 muestra la mediana y la varianza del porcentaje de clasificación de los 33 experimentos independientes para cada configuración, así como los porcentajes obtenidos por descriptores del estado del arte [15]. De los resultados obtenidos, se puede observar que en este experimento, los descriptores generados por GE obtuvieron un mejor desempeño.

En la tabla 4 se muestra el mejor descriptor generado por cada configuración y el porcentaje de clasificación obtenido.

La tabla 5 muestra el porcentaje del uso de cada canal de color, la escala de grises y la mezcla de éstos, lo cual fue calculado con los mejores descriptores

 ${\bf Tabla~2.~Descripci\'on~de~los~s\'imbolos~terminales~usados~por~la~gram\'atica~BNF.}$

Símbolos	
terminales	Descripción
Maxpooling2	Max-pooling 2D con una ventana de agrupado de tamaño 2X2
Maxpooling4	Max-pooling 2D con una ventana de agrupado de tamaño 4X4
Maxpooling6	Max-pooling 2D con una ventana de agrupado de tamaño 6X6
Maxpooling8	Max-pooling 2D con una ventana de agrupado de tamaño 8X8
Maxpooling10	Max-pooling 2D con una ventana de agrupado de tamaño 10X10
Ir	Canal rojo de la imagen de entrada
Ig	Canal verde de la imagen de entrada
Ib	Canal azul de la imagen de entrada
Igray	Escala de grises de la imagen de entrada
Gau1	Filtro de suavizado Gaussiano con una $\sigma=1$
Gau2	Filtro de suavizado Gaussiano con una $\sigma=2$
GauX	Derivada a lo largo del eje X del filtro gaussiano
GauY	Derivada a lo largo del eje Y del filtro gaussiano
LoG1	Laplaciano de filtro gaussiano con $\sigma=1$
LoG2	Laplaciano de filtro gaussiano con $\sigma=2$
Lap	Filtro Laplaciano
GBO-0	Filtro de Gabor con orientación de 0 grados
GBO-45	Filtro de Gabor con orientación de 45 grados
GBO-90	Filtro de Gabor con orientación de 90 grados
GB0-135	Filtro de Gabor con orientación de 135 grados
Aver	Filtro de promediado con una ventana de agrupado de tamaño 5x5
Med	Mediana en una ventana de agrupado de tamaño 5x5
EHIS	Ecualización del histograma
ABS	Valor absoluto de una imagen píxel por píxel
ADD	Suma dos imágenes píxel por píxel
SUB	Resta dos imágenes píxel por píxel
ABSsub	Resta absoluta dos imágenes píxel por píxel
MULTI	Multiplicación de dos imágenes píxel por píxel
DIV	División de dos imágenes píxel por píxel pero regresa 0 sí el divisor es 0
SQR	Cuadrado de una imagen píxel por píxel
SQRT	Raíz cuadrada de una imagen píxel por píxel
LOG2	Logaritmo cuadrado de una imagen píxel por píxel
$\mathrm{TIME}\text{-}0.5$	Multiplicación de una imagen por 0.5 píxel por píxel

Una comparativa de diferentes parámetros en la construcción evolutiva de descriptores...

Tabla 3. Comparación de los porcentajes de clasificación obtenidos por nuestra propuesta y por otros descriptores del estado del arte.

	Datos	
Configuraciones	Mediana	Varianza
BNF1-H	0.8997	0.000091
BNF2- H	0.8852	0.000253
BNF1- EH	0.8681	0.000365
BNF2-EH	0.8726	0.000181
Descriptores de Law [15]	0.7391	*
Filtros de Gabor [15]	0.6812	*
Descriptores de Haralick [15]	0.7101	*

^{*}El trabajo no reporta varianza

 ${\bf Tabla~4.}~{\bf Ejemplos~de~los~mejores~descriptores~generados~por~las~cuatro~configuraciones~de~{\bf GE}~y~sus~respectivos~desempe\~nos.$

Configuración	Descriptor	Porcentaje de clasificación
BNF1-H	Maxpooling10(GBO-	92.68%
	90(MULTI(Igray,DIV(LoG1(Igray),EHIS(
	Igray)))))	
BNF2-H	Maxpooling4(TIME-	92.68%
	0.5(MULTI(SUB(Igray,Igray),DIV(LoG1(GBO-	
	135(DIV(DIV(SQR(SQRT(GBO-	
	0(Igray))),DIV(LOG2(ABS(Igray)),	
	[gray)), [gray))), SQR(Gau1([gray))))))	
BNF1-EH	${\tt Maxpooling2}({\tt SQRT}({\tt MULTI}({\tt Igray}, {\tt DIV}({\tt Igray}, $	90.22%
	Ig,LoG2(ADD(Igray,SUB(SQRT(Ir),Igray)))),	
	Igray))))	
BNF2-EH	${ m Maxpooling 10 (LoG1(LoG1(SUB(Igray, ADD(Igray, AD$	91.48%
	MULTI(Igray,Igray),Igray)))))	

generados en los 33 experimentos de las configuraciones de BNF1. De los resultados mostrados, se observa que BNF1-H utiliza más el canal verde mientras que BNF1-EH lo hace con la mezcla de canales.

 ${\bf Tabla~5.}~ Porcentaje~ del~ uso~ de~ los~ canales~ de~ color,~ escala~ de~ gris~y~ la~ mezcla~ de~ canales~ por~ las~ configuraciones~ de~ BNF1.$

		Canales de color			
Configuraci	$\mathbf{\acute{o}n}$ $Rojo$	Verde	Azul	Escala de grises	$Mezcla\ de\ canales$
BNF1-H	0 %	90.91%	0 %	3.3%	6.06%
BNF1-EH	6.06%	15.15%	6.06%	24.24%	48.48~%

Finalmente, la tabla 6 muestra la mediana, varianza, mínimo y máximo del número de símbolos terminales usados por los 33 mejores descriptores de cada configuración. Los resultados muestran que la configuración BNF1-H tiene una media de símbolos terminales usados menor que las otras configuraciones.

Tabla 6. Datos estadísticos del número de símbolos terminales usados por cada configuración.

	Datos estadísticos			
Configuración	Mediana	Varianza	$M\'inimo$	$M\'{a}ximo$
BNF1-H	3	12.309	3	18
BNF2-H	5	30.195	3	24
BNF1-EH	6	11.797	3	15
BNF2-EH	5	10.627	4	20

5. Conclusiones

En el presente trabajo se mostró una metodología, basada en GE, la cual permite generar descriptores aplicados a la clasificación de imágenes de texturas de piezas arqueológicas.

En base a los resultados obtenidos, se puede afirmar que nuestra propuesta es viable de utilizar cuando no se tiene la suficiente información a priori de la base de datos del problema a tratar, pues para este experimento se obtuvieron desempeños mayores a los obtenidos por otros descriptores del estado del arte. Así también esta metodología puede ser usada en la solución de otros tipos de problemas de clasificación modificando el diseño de la gramática.

Con los datos que nos proporciona la tabla 3, podemos identificar que las configuraciones que utilizaron el histograma como parámetro de representación del vector de características, obtuvieron un mejor desempeño que las que utilizaron los estadísticos del histograma. También se observa que la configuración BNF1-H tiene una menor varianza que las demás configuraciones.

El presente trabajo busca servir de apoyo al arqueólogo para su toma de decisiones.

Como trabajo futuro, los autores proponen el diseño de gramáticas utilizando otros descriptores básicos, los cuales se pueden aplicar a diferentes bases de datos de imágenes de texturas. También se propone utilizar otros algoritmos de clasificación y motores de búsqueda en el proceso evolutivo.

Agradecimientos. Los autores agradecen el apoyo recibido del Tecnológico Nacional de México - Instituto Tecnológico de León y del Consejo Nacional de Ciencia y Tecnología (CONACyT) por medio de la beca para estudiantes de posgrado: 778561 (J. Jiménez) y CÁTEDRAS-2598 (A. Rojas). Así también se agradece a la Dra. Irmgard Hein el suministro de la base de datos de imágenes utilizada en este experimento.

Referencias

- Lu, D., Weng, Q.: A survey of image classification methods and techniques for improving classification performance. International Journal of Remote Sensing 28(5), 823—870 (2007)
- Mehta, R., Eguiazarian , K. E.: Texture Classification Using Dense Micro-Block Difference. IEEE Transactions on Image Processing 25(4), 1604-1616 (2016)
- 3. Sá Junior, J., Backes, A. R.: ELM based signature for texture classification. Pattern Recognition 51, 395—401 (2016)
- Ledoux, A., Losson, O., Macaire, L.: Texture classification with fuzzy color co-occurrence matrices. In: 2015 IEEE International Conference on Image Processing (ICIP), pp. 1429–1433. IEEE, Quebec City (2015)
- Zhang, Z., Liu, S., Mei, X., Xiao, B., Zheng, L.: Learning completed discriminative local features for texture classification. Pattern Recognition 67, 263-275 (2017)
- 6. Zhao, Y., Wang, R. G., Wang, W. M., Gao, W.: Local Quantization Code histogram for texture classification. Neurocomputing 207, 354–364 (2016)
- Shao, L., Liu, L., Li, X.: Feature Learning for Image Classification Via Multiobjective Genetic Programming. IEEE Transactions on Neural Networks and Learning Systems 25(7), 1359–1371 (2014)
- 8. Tran, B., Zhang, M., Xue, B.: Multiple feature construction in classification on high-dimensional data using GP. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8. IEEE, Athens (2016)
- Lensen, A., Al-Sahaf, H., Zhang, M., Xue, B.: Genetic Programming for Region Detection, Feature Extraction, Feature Construction and Classification in Image Data. Heywood, M., McDermott, J., Castelli, M., Costa, E., Sim, K. (eds) EuroGP 2016, LNCS, vol. 9594, pp. 51-67. Springer, Cham (2016).
- 10. Gavrilis, D., Tsoulos, I. G., Dermatas, E.: Selecting and constructing features using grammatical evolution. Pattern Recognition Letters 29(9), 1358–1365 (2008)
- Ryan, C., Collins, JJ, O Neill, M.: Grammatical evolution: Evolving programs for an arbitrary language. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds) EuroGP 1998, LNCS, vol. 1391, pp. 83-96. Springer, Heidelberg (1998)
- 12. Koza, J. R., Poli, R.: Genetic Programming. 1st edn. Springer US, Boston (2005)
- Dempsey, I., O'Neill, M., Brabazon, A.: Foundations in Grammatical Evolution for Dynamic Environments. 1st edn. Springer Berlin Heidelber, Heidelber (2009)
- Sotelo-Figueroa, M. A., Soberanes, H. J., Carpio, J. M., Fraire, H. J., Cruz, L., Soria-Alcaraz, J. A.: Improving the bin packing heuristic through grammatical evolution based on swarm intelligence. Mathematical Problems in Engineering 2014, 1–12, (2014)
- Hein, I., Rojas-Domínguez, A., Ornelas, M., D'Ercole, G., Peloschek, L.: Automated classification of archaeological ceramic materials by means of texture measures. Journal of Archaeological Science: Reports 2017, Elsevier Press (2017)
- Bradley, A. P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30(7), 1145-1159 (1997)

Diseño y desarrollo de una arquitectura IoT en contexto con la plataforma FIWARE

Alicia Martínez¹, Haidée Onofre¹, Hugo Estrada², Daniel Torres¹, Olivia Maquinay¹

¹Tecnológico Nacional de México/CENIDET, Cuernavaca, Morelos, México

²Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación
{amartinez, olivia}@cenidet.edu.mx, {haidee.onofreramirez,
torresestradadaniel}@gmail.com, hugo.estrada@infotec.mx

Resumen. En los últimos años, el surgimiento del concepto de Internet de las cosas ha impulsado la creación y el desarrollo de plataformas que permiten procesar grandes volúmenes de información de contexto, producida por redes de sensores que monitorean espacios o ambientes inteligentes. Estas plataformas procesan la información de contexto que producen las aplicaciones inteligentes a través de diferentes dispositivos y sensores conectados a Internet. FIWARE es una plataforma de código abierto que impulsa la creación de estándares para el desarrollo de aplicaciones y servicios inteligentes de diferentes dominios, en la nube. En este artículo se presenta el diseño y desarrollo de una arquitectura de Internet de las cosas, que hace uso de los estándares, componentes genéricos y APIs que ofrece la plataforma FIWARE. El artículo presenta también, la implementación de la arquitectura propuesta en una aplicación para el monitoreo y sensado de la conducción de automóviles en estacionamientos de instituciones y empresas.

Palabras clave: FIWARE, internet de las cosas, arquitecturas de internet de las cosas.

Design and Development of an Internet of Things Architecture in Context with the FIWARE Platform

Abstract. In the last few years, the emergence of Internet of Thing concept has boosted the creation and development of platforms that allows to process bigger volumes of context information produced by sensor networks monitoring smart environments. These platforms processing the context information that the smart applications produce through of different devices and sensors connected to Internet. FIWARE is an open source platform that impulses the standards creation for the development of smart applications and services of different domains, in the cloud. This paper presents the design and development of an Internet of Things Architecture, making use of the standards, generic enablers and APIs that offers the FIWARE platform. This paper also presents, the implementation of the architecture proposed, with an application for the monitoring of cars driving in the parking lots of institutions and companies.

Keywords: FIWARE, internet of things, internet of things architectures.

1. Introducción

Hoy en día, el Internet de las Cosas (IoT) y la Inteligencia Artificial (AI) se encontrarán entre las tecnologías transformadoras más importantes a lo largo de este año 2018 [1]. Estas tecnologías están aportando un gran valor a las empresas y las personas que utilizan estos nuevos servicios.

Actualmente, la automatización de procesos mediante técnicas de inteligencia artificial es una prioridad en las empresas que desean mejorar sus niveles de eficiencia. El IoT, por otro lado, ha sido un tema recurrente en los últimos años y su función es llevar el análisis en tiempo real a los procesos automatizados. El IoT genera un flujo de trabajo sofisticado para personalizar las experiencias que las empresas ofrecen a sus clientes y usuarios [2], utilizando sistemas que conectan el mundo físico con el mundo digital. En este sentido, IoT se ha definido como "un concepto que propone la conexión de los objetos cotidianos y de uso común a Internet" [3].

Mediante el Internet de las cosas, los objetos se reconocen a sí mismos y obtienen un comportamiento de inteligencia, al tomar decisiones relacionadas sobre la información que pueden comunicar sobre sí mismos [4]. De esta manera, los objetos inteligentes que intervienen dentro de una arquitectura IoT pueden producir información de contexto en grandes cantidades, es por esto que, actualmente, se requiere del uso de plataformas especializadas que cuenten con estándares y protocolos para el manejo y procesamiento de información de IoT a gran escala, con el objetivo de crear y desplegar aplicaciones de IoT inteligentes y administrables.

La plataforma FIWARE, desarrollada como resultado de una asociación públicaprivada en Europa, permite desarrollar aplicaciones de software relacionadas con Internet del Futuro. Esta plataforma ofrece componentes para realizar análisis de grandes volúmenes de datos y la provisión de métricas en tiempo real, manipulación de información de contexto, análisis de eventos en tiempo real, recopilación de información de sensores y acción sobre actuadores. Para lograr esto, FIWARE presenta un conjunto de APIs genéricas estandarizadas que, facilitan la comunicación con Internet de las cosas y manipulan la información de contexto. Estos componentes pretenden impulsar la creación de estándares necesarios para desarrollar aplicaciones inteligentes en diferentes dominios.

En el presente artículo se presenta el diseño y desarrollo de una arquitectura para IoT que utiliza los estándares, componentes genéricos y APIs de FIWARE, y se combina con capas de servicio y de sensado para considerar un escenario completo de Internet de las cosas. Esta arquitectura fue utilizada para la implementación de una aplicación inteligente que monitorea la conducción de un usuario a través de su teléfono móvil. La aplicación realiza la detección de eventos de conducción de manera automática, los cuales pueden ser: conducción a una velocidad no autorizada, conducción en sentido contrario y paradas repentinas de un vehículo. Además, la aplicación envía alertas a los usuarios que se encuentran cerca del usuario que emitió la alerta y al guardia de seguridad de la empresa.

El artículo está organizado de la siguiente manera: la Sección 2 muestra los fundamentos teóricos utilizados en nuestra propuesta. La Sección 3 detalla el diseño y desarrollo de nuestra arquitectura de IoT utilizando la plataforma FIWARE. La Sección 4 muestra la implementación de la arquitectura propuesta y la aplicación IoT

generada a partir de esta arquitectura. Finalmente, la Sección 5 presenta las conclusiones y trabajos futuros.

2. Fundamentos teóricos y trabajos relacionados

2.1. Plataformas IoT

Las plataformas IoT pueden definirse como: "servicios diseñados para facilitar la sociedad de la información, a través de la interconexión (física y digital) de cosas basadas en las tecnologías de la información, sobre un modelo de cómputo en la nube" [5]. Esta definición señala el vínculo entre IoT y el concepto de cómputo en la nube; como plataformas ofrecidas por un proveedor de servicios, para el despliegue de aplicaciones de software que manejan a gran escala información de contexto proveniente de "dispositivos o cosas".

Actualmente, el mercado de las plataformas IoT está en auge y en continua expansión Hay encuestas que hacen mención de que más del 80% de las empresas cree que el campo del Internet de las Cosas es el más interesante para sus negocios. Algunas plataformas IoT son las Plataformas de conectividad/M2M, que se centran principalmente en establecer las condiciones que permitan a un dispositivo el intercambio de información de manera bidireccional con otro dispositivo, o con una aplicación de negocios, usando una red de telecomunicaciones [6].

Existen otras plataformas de software específicas de hardware. Una de estas plataformas es: PAVENET [7], plataforma para redes de sensores a partir del diseño de un nodo inalámbrico basado en Dual-CPU, denominado U3, como alternativa a los dispositivos disponibles, ya que proponen las dificultades de utilizar nodos basados en un solo núcleo de procesamiento para el desarrollo de aplicaciones que necesiten grandes recursos de procesamiento disponibles. Otro trabajo es LiteOS [8], plataforma que ha sido especialmente creada para programadores que deseen desarrollar aplicaciones para redes de sensores inalámbricas sin tener mayores conocimientos de la plataforma hardware, ya que se basa en la filosofía de trabajo UNIX, así como programación enfocada a hilos de ejecución, y programación orientada a eventos.

Otra plataforma es FIWARE, la cual proporciona capacidades en la nube basadas en OpenStack [9], y mejoradas junto con un conjunto de herramientas y librerías de valor añadido denominadas Generic Enablers (GEs). Estos GEs, ofrecen interfaces (APIs) estándar y abiertas que facilitan tareas como la integración de dispositivos de Internet de las Cosas (Internet of Things, IoT), analizar y procesar datos a media y gran escala (Big Data), o incorporar interfaces avanzadas para interactuar con los usuarios.

2.2. Ecosistema FIWARE

FIWARE es una plataforma de código abierto que impulsa la creación de estándares para el desarrollo de aplicaciones y servicios inteligentes de Internet del Futuro en diferentes dominios [10], que proporciona un conjunto de componentes genéricos e interfaces de programación de aplicaciones (APIs, por sus siglas en

inglés: *Application Programming Interface*), con el propósito de facilitar el desarrollo y despliegue de aplicaciones inteligentes en múltiples sectores y escenarios [11].

Algunos de los componentes genéricos y APIs que conforman el ecosistema FIWARE se describen a continuación.

Orion Context Broker. El Orion Context Broker es un componente genérico de FIWARE que permite manipular información de contexto y habilitar la suscripción de aplicaciones a estos datos de contexto, otorgando la oportunidad de disponer de estos datos de contexto en todo momento [12]. La información de contexto puede provenir de fuentes diferentes como: sistemas ya existentes, usuarios (a través de aplicaciones móviles) y redes de sensores.

El Orion ContextBroker (Fig. 1) permite modelar y obtener acceso a la información de contexto independientemente de la fuente de donde proviene esa información. Este componente está habilitado para manejar información de contexto a gran escala mediante la implementación del estándar API REST [13]. La manipulación de información de contexto en el Orion ContextBroker de FIWARE se realiza siguiendo la especificación de la API estándar: FIWARE NGSI.



Fig. 1. Esquema del Orion ContextBroker de FIWARE.

API FIWARE NGSI. La API estándar que propone FIWARE para recopilar, gestionar, publicar e informar sobre cambios en la información de contexto se denomina FIWARE NGSI (*Next Generation Service Interface*, por sus siglas en inglés). Esta API está diseñada para administrar todo el ciclo de vida de la información de contexto, incluyendo actualizaciones, consultas, registros y suscripciones [14].

Los elementos principales en el modelo de datos NGSI son: entidades de contexto, atributos y metadatos, como se muestran en la Fig. 2.



Fig. 2. Elementos del modelo de datos FIWARE NGSI.

Las entidades de contexto, o simplemente entidades, son el centro del modelo de datos NGSI de FIWARE. Una entidad representa una cosa. Por ejemplo, cualquier objeto físico o lógico (un sensor, una persona, una habitación, etc.). Los atributos de contexto son propiedades que describen las entidades de contexto y los metadatos son piezas de información que se encargan de describir características adicionales al atributo de contexto.

La API FIWARE NGSI es utilizada dentro del Orion ContextBroker para la manipulación de información de contexto de los modelos de datos NGSI. Actualmente la versión utilizada de esta API es FIWARE NGSI en su versión 2 (FIWARE NGSIv2).

API QuantumLeap. QuantumLeap es un API de FIWARE que soporta el almacenamiento de modelos de datos del estándar NGSI de FIWARE en una base de datos de series de tiempo [15]. Orion ContextBroker es el encargado de notificar a la API QuantumLeap los cambios en un modelo de datos, esto se realiza a través de la creación de una suscripción de un modelo de datos en el Orion ContextBroker, la cual tendrá como función notificar los cambios en el modelo de datos a la API QuantumLeap.

La API QuantumLeap recibe los modelos de datos en formato NGSI y los transporta al motor de base de datos correspondiente para su almacenamiento. Previo a esto, la API QuantumLeap realiza la conversión de cada atributo de los modelos de datos en formato NGSI, al que corresponda en el motor de base de datos, según el tipo de dato establecido para cada atributo en los modelos de datos.

3. Diseño y desarrollo de la arquitectura IoT

La arquitectura propuesta en este artículo tiene como objetivo utilizar los componentes de la plataforma FIWARE para diseñar e implementar aplicaciones inteligentes que interactúen con capas de servicio y de sensado de información de contexto. Con la finalidad de instanciar esta arquitectura en un caso específico, se diseñó e implementó una aplicación inteligente, denominada *DrivingApp*, que permite monitorear dentro de una institución o empresa el manejo de los vehículos que acceden a los estacionamientos de la empresa.

En caso de que un vehículo sea conducido de manera errónea, la aplicación enviará una alerta al guardia de seguridad, así como a los usuarios de la aplicación que se encuentren cerca de donde se generó la alerta.

Durante la etapa de desarrollo de la aplicación inteligente de IoT (*DrivingApp*), fueron identificados diversos inconvenientes en la manipulación del sensor GPS, dada la inexactitud que presenta en los valores de las coordenadas, lo cual depende directamente del tipo de sensor con el que cuenta el dispositivo, teniendo una eficacia aproximadamente del 70% en la mayoría de los casos; como respuesta a esta anomalía se adecuó una alternativa: que el dispositivo obtuviera su ubicación por medio de la red (wi-fi o de datos móviles), si es que el dispositivo estuviera conectado a una, en caso contrario esta se obtendría del GPS.

El comportamiento inteligente de las aplicaciones y servicios que integran la arquitectura de IoT propuesta, radica en la detección de eventos de conducción de manera automática, los cuales pueden ser: conducción a una velocidad no autorizada,

conducción en sentido contrario y paradas repentinas de un vehículo; además de la búsqueda y elección de los dispositivos móviles a los que se envía la alerta, según la configuración de envío de alertas establecida en la aplicación móvil por medio del usuario. La arquitectura desarrollada (ver Fig. 3) está compuesta por tres módulos: capa de aplicación y sensado, capa de servicios y el ecosistema FIWARE.

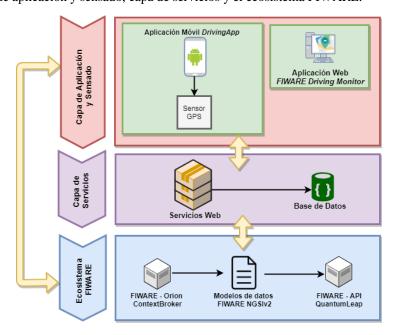


Fig. 3. Arquitectura IoT en contexto con el ecosistema FIWARE.

3.1. Capa de aplicación y sensado

La capa de aplicación y sensado es la encargada de recolectar y dar tratamiento a la información de contexto que se obtiene a través la aplicación movil (*Driving App*) y la aplicación web (*FIWARE Driving Monitor*). La aplicación móvil (*DrivingApp*) procesa y manipula los datos recolectados del teléfono móvil, de las lecturas realizadas del sensor GPS. Una vez que la información recolectada es procesada, se detecta si sucede algún evento que pueda generar alguna alerta, en caso de que esto ocurra, la alerta es presentada en la interfaz gráfica del dispositivo móvil, así como también en la aplicación web.

La aplicación móvil consume los servicios web de la capa de servicios para extraer y modificar información de la base de datos. Además, la aplicación móvil establece comunicación directa con el Orion ContextBroker para el envío de modelos de datos FIWARE NGSI. El Orion ContextBroker almacena estos modelos de datos y los replica, ya sea a la API QuantumLeap o a la capa de servicios, según sea el caso.

La aplicación web de monitoreo (FIWARE Driving Monitor) es la encargada de mostrar la información de contexto recopilada por la aplicación móvil de IoT. Esta información de contexto generada por la aplicación móvil de IoT, la obtiene a través

de la capa de servicios, y la clasifica y representa a través diferentes elementos gráficos representativos.

3.2. Capa de servicios

La capa de servicios es un módulo que tiene como funcionalidad principal ser intermediario de comunicación entre el ecosistema FIWARE y la capa de aplicación.

La información de contexto generada por la aplicación móvil IoT es enviada a la capa de servicios, donde es manipulada a través de los módulos de servicios web, para su almacenamiento en la base de datos y en el Orion ContextBroker de FIWARE.

Los servicios web de la capa de servicios están divididos en 3 módulos, estos módulos proveen las siguientes funcionalidades:

- API RESTFul: es un servicio que actúa como interfaz para el intercambio de información (inserción, actualización, eliminación y consulta de datos) entre las aplicaciones y la base de datos.
- Servidor de Notificaciones: es un servicio que tiene como objetivo replicar notificaciones de alertas emitidas por los usuarios de la aplicación móvil. Cada alerta emitida en la aplicación móvil es manipulada en este servicio, para ser replicada como notificación a los dispositivos móviles que hagan uso de la aplicación móvil DrivingApp. Adicional a la notificación de la alerta a los dispositivos móviles, el servidor de notificaciones se encarga de enviar las notificaciones de estas alertas a la aplicación web.
- Estas funcionalidades son posibles mediante a una conexión directa del servidor de notificaciones con el Orion ContextBroker de FIWARE, a través de la suscripción a la información de contexto del modelo de datos de alerta FIWARE NGSIv2.
- Servicio de consultas de contexto: es un servicio que tiene la capacidad de responder a las consultas de información de contexto específicas provenientes de la capa de aplicación. Por ejemplo, las solicitadas por la aplicación móvil o web.

3.3. Ecosistema FIWARE

El ecosistema FIWARE es la plataforma en la nube dentro de la arquitectura IoT propuesta en este trabajo. Este ecosistema se encarga de manipular y persistir la información de contexto proveniente de las aplicaciones, a través de los componentes que integran la plataforma. Se utilizan tres componentes claves del ecosistema FIWARE en este proyecto:

- Orion ContextBroker: este componente de FIWARE se utiliza para capturar los datos de contexto producidos por los sensores del teléfono inteligente, y representa estos datos en modelos de datos definidos que siguen el estándar FIWARE NGSIv2.
- Modelos de datos FIWARE NGSIv2: los modelos de datos de FIWARE utilizados son: Alert, Device, DeviceModel, Road, RoadSegment

OffStreetParking y Building categoría tipo "Zone" [16]. Estos modelos de datos sirven para estandarizar la información de contexto referente a: la detección de eventos de alerta emitidos a través de la aplicación móvil, las propiedades de los dispositivos móviles, los valores obtenidos por el sensor GPS del teléfono inteligente y, la información de las calles y estacionamientos que se encuentran dentro de la zona comprendida por la empresa o institución.

— API QuantumLeap: este componente se utiliza para almacenar los datos obtenidos del sensor GPS como series de tiempo, de esta forma será posible realizar visualizaciones de los datos a lo largo de cierto periodo de tiempo.

El aporte principal de este trabajo en comparación con otras arquitecturas y plataformas de IoT, es el diseño y desarrollo de la arquitectura descrita, en la que se hace uso del ecosistema FIWARE como plataforma de IoT, el cual interactúa con capas de servicio y de sensado de datos de contexto. Además, esta arquitectura no sólo se ha definido en lo abstracto, sino que ha sido implementada en una aplicación inteligente de IoT, lo cual ha permitido verificar el funcionamiento y operación de la arquitectura propuesta.

Al utilizar FIWARE como plataforma de IoT se facilita la conexión de dispositivos y almacenamiento de datos, integrando los protocolos de dispositivos y métodos de conectividad, para dar interpretación a la información relevante; asimismo permite aislar el procesamiento de datos, de las capas de servicio y de aplicación y sensado de la arquitectura IoT desarrollada.

4. Implementación de la arquitectura IoT propuesta

La implementación de la arquitectura propuesta se llevó a cabo a través del desarrollo de dos aplicaciones: *DrivingApp y FIWARE Driving Monitor*. Estas aplicaciones desempeñan sus funcionalidades en la capa de aplicación y sensado de la arquitectura IoT, y se comunican con la capa de servicios y con el ecosistema FIWARE respectivamente. Actualmente las aplicaciones se encuentran en la etapa final del desarrollo para posteriormente iniciar con la etapa de pruebas de software.

La implementación de la arquitectura IoT propuesta tiene como ventajas: la integración de servicios web que pueden adaptarse para ser consumidos por aplicaciones inteligentes de IoT con propósitos similares a la que se desarrolla en el presente trabajo; facilita el despliegue e implementación de una aplicación de monitoreo de conducción de automóviles que puede ser utilizada por empresas o instituciones que requieran de estos servicios, y que, además, busquen un entorno de desarrollo que puedan operar con un mínimo esfuerzo de administración, configuración y mantenimiento. La utilización de la plataforma FIWARE permitió el desarrollo de un sistema modular basado en componentes reusables, lo cual acelera el tiempo de construcción de sistemas. Si todos los componentes del sistema presentado en este artículo hubiesen sido desarrollados desde cero, se habría invertido una gran cantidad de tiempo únicamente para desarrollar el componente de manejo de datos de contexto, el cual ya es proporcionado por la plataforma FIWARE. Estas

características de modularidad también se presentan en la aplicación generada en este trabajo, lo cual facilita su reutilización.

En los resultados derivados de la experimentación e implementación de la arquitectura IoT (a través de las aplicaciones desarrolladas), se logró demostrar que el uso de la plataforma FIWARE facilita la conexión de dispositivos y manejo información de contexto; sin embargo la integración de FIWARE como la plataforma IoT dentro la arquitectura propuesta, fue uno de los principales retos en la etapa de diseño, ya que esto requirió de buscar la forma óptima de comunicación de los componentes de la plataforma FIWARE y el resto de las capas de arquitectura del sistema propuesto. Es importante destacar que la arquitectura es resultado de un uso novedoso de los componentes FIWARE unidos a capas de servicio y de sensado.

4.1. Aplicación móvil de IoT

DrivingApp es una aplicación móvil de IoT para dispositivos móviles Android. La Fig. 4 muestra una vista de la aplicación. La aplicación móvil tiene como objetivo detectar de manera automática la errónea conducción de un automóvil.

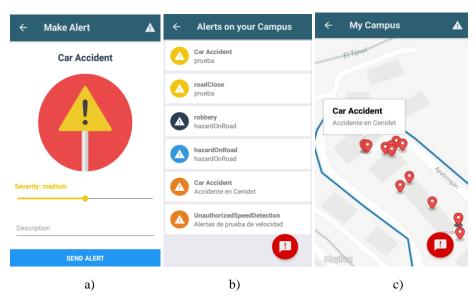


Fig. 4. Aplicación Móvil de IoT: *DrivingApp* a) Generación manual de alerta de accidente automovilístico, b) Listado de las alertas suscitadas más recientes, c) Mapa de la ubicación de las alertas suscitadas dentro de la institución.

La aplicación es configurada para monitorear a los usuarios de la aplicación dentro de los estacionamientos de una empresa o institución. Los eventos que detecta la aplicación móvil son: conducción a una velocidad no autorizada, conducción en sentido contrario y paradas repentinas de un vehículo.

El sensor utilizado para la detección automática de los eventos es el GPS, y además, se hace uso de los modelos de datos de FIWARE NGSIv2 mencionados con anterioridad dentro del ecosistema FIWARE, con el objetivo de estandarizar la

información recolectada. El teléfono se encuentra en constante recolección de la información del contexto; es decir, cada segundo se obtiene la posición GPS del teléfono y la velocidad a la que se encuentra. Esta información es procesada en el teléfono para determinar si sucede algún evento que pueda generar alguna alerta.

Cuando un evento de conducción errónea es detectado, se genera una alerta y se envía a todos los usuarios de la aplicación que se encuentren cerca del lugar en donde se generó dicha alerta. Además, también se envía la alerta a la aplicación web, que será utilizada por el guardia de seguridad de la empresa para notificarle que un usuario se encuentra conduciendo de manera errónea.

El sistema permite también generar alertas de manera manual, con el objetivo de alertar a los usuarios de eventos relacionados a tráfico, accidentes y/o seguridad, lo cuales sean observados en un área o lugar específico dentro de las instalaciones de la institución.

4.2. Aplicación web de monitoreo

FIWARE Driving Monitor es una aplicación web que tiene como objetivo gestionar la información de los usuarios de la aplicación móvil DrivingApp y la información de contexto generada por dicha aplicación. La Fig. 5 muestra una vista de la aplicación en la cual se muestra delimitado el mapa de la ubicación de la empresa.

Además, se muestran en el mapa las alertas que se han generado en diferentes puntos de la empresa. La aplicación provee las herramientas gráficas necesarias para realizar la delimitación geográfica en mapas, de las sucursales o campus de la empresa y de las zonas que comprenden los estacionamientos de la empresa.

El usuario de esta aplicación puede ser el guardia de seguridad de la empresa. Por lo que, este usuario podrá consultar el control estadístico de la información de contexto referente a alertas de eventos ocurridos dentro de los campus y zonas de la institución.

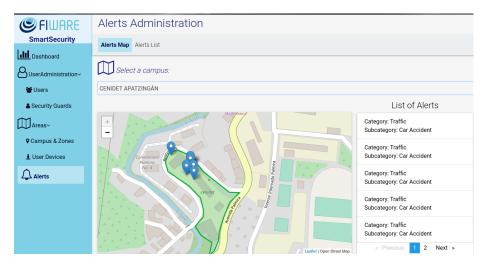


Fig. 5. Aplicación Web de Monitoreo: FIWARE Driving Monitor.

La información antes mencionada se muestra a través de gráficas y tablas en donde se visualizan los tipos de eventos ocurridos y el número de incidencias de cada uno. Esta aplicación web es la encargada de administrar el registro de información de contexto de los usuarios, como su localización. Esto con el fin de determinar si algún usuario se encuentra actualmente dentro de la empresa o bien, si estuvo en determinado momento dentro de esta, y esta información del usuario se muestra a través de tablas y marcadores en el mapa.

5. Conclusiones

La arquitectura de IoT propuesta en este artículo permite demostrar el uso de FIWARE como una plataforma que permite el desarrollo de aplicaciones inteligentes para diferentes dominios de aplicación.

La arquitectura desarrollada permite el despliegue de componentes y módulos de cada capa de forma independiente; esto demuestra el bajo acoplamiento de los módulos entre cada de capa y ofrece la posibilidad de implementar las funcionalidades de una capa sin necesidad de desplegar las demás capas.

La implementación de la arquitectura propuesta utiliza las capacidades de la plataforma FIWARE, para desarrollar una aplicación móvil que permite sensar la conducción de automóviles en estacionamientos de instituciones o empresas, a fin de detectar eventos de alerta durante la conducción del vehículo.

Es importante destacar que la arquitectura propuesta es novedosa porque permite unir componentes de propósito general de la plataforma FIWARE, con capas diseñadas con servicios web y con capas para el sensado de información de contexto. Una de las consecuencias directas del uso de los componentes FIWARE es la modularidad del sistema, lo cual facilita su adaptación y reuso, tanto a nivel de arquitectura para diversas aplicaciones IoT, como de la implementación específica que se realizó para validar la arquitectura, la cual puede ser reusada en entornos donde se requiera monitorear la actividad de los usuarios dentro de los estacionamientos y de las instalaciones de la empresa en general.

Como trabajos futuros, podría explorarse el diseño de una arquitectura de IoT configurable, de manera que esta pueda capturar información de diferentes tipos de sensores o dispositivos sin necesidad de adaptaciones al código. Por otra parte, sería útil analizar las posibilidades de desplegar otros componentes genéricos y servicios de la plataforma FIWARE, para su integración dentro de la arquitectura de IoT propuesta, con el propósito de enriquecer y ampliar las funcionalidades de la arquitectura actual; de manera que se pueda extender el uso de la arquitectura, implementándose en diversos escenarios de soluciones IoT con enfoques o dominios de aplicación distintos, y aplicables para un mismo espacio inteligente.

Agradecimientos. Esta investigación ha sido financiada por la Comisión Europea y CONACYT a través del proyecto SmartSDK.

Referencias

 Portaltic Europa Press, http://www.europapress.es/portaltic/sector/noticia-internet-cosasinteligencia-artificial-pilares-transformacion-digital-2018-fujitsu-20180201124739.html, (2018)

- Diario del profesional Ti, https://diarioti.com/cuando-la-inteligencia-artificial-el-internetde-las-cosas-y-el-blockchain-convergen/105992 (2018)
- 3. Gubbi, J., Buyya, R., Marusic, S., Palaniswamia, M.: Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. Future Generation Computer Systems Journal 29(7), pp. 1645–1660 (2013)
- 4. Aldein Mohammed, Z., Ali Ahmed, E.: Internet of Things Applications, Challenges and Related Future Technologies. World Scientific News Journal 67(2), pp. 126–148 (2017)
- 5. Pratim Ray, P.: A survey of IoT cloud platforms. Future Computing and Informatics Journal 1(1-2), pp. 35–46 (2016)
- 6. Boswarthick, D., Elloumi, O., Hersent, O.: M2M Communications: A Systems Approach. John Wiley & Sons, Editor. West Sussex, United Kingdom (2012)
- Saruwatari, S., Kashima, T., Minami, T., Morikawa, H., Aoyama, T.: PAVENET: A
 Hardware and Software Framework for Wireless Sensor Networks. Trans. of the Society
 of Instrument and Control Engineers Vol. E-S-1, no.1, pp. 74–84 (2005)
- 8. Cao, Q., Abdelzaher, T., Stankovic, J., He, T.: The LiteOS Operating System: Towards Unix Like Abstraction for Wireless Sensor Networks. In: Proceedings of the 7th International Conference on Information Processing in Sensor Networks, pp. 233. St. Louis, MO, USA (2008)
- 9. Kumar, R. *et al.*: Open Source Solution for Cloud Computing Platform Using OpenStack. International Journal of Computer Science and Mobile Computing 3(5), pp. 89 98 (2014)
- 10. Telefónica. FIWARE, el estándar que necesita el IoT (2018)
- 11. FIWARE: European Commission. About Us (2018)
- 12. FIWARE ORION Documentation. http://fiware-orion.readthedocs.io/en/stable/ (2018)
- 13. FIWARE IOT Stack. Context Broker (Orion). http://fiware-iot-stack.readthedocs.io/en/latest/context_broker/#development-of-context-aware-applications (2018)
- Telefónica I+D FIWARE-NGSI v2 Specification, http://fiware.github.io/specifications/ ngsiv2/stable/, último acceso 2018/03/22
- QuantumLeap Home, https://smartsdk.github.io/ngsi-timeseries-api/, último acceso 2018/03/30
- FIWARE Data Models repository. https://github.com/Fiware/dataModels, último acceso 2018/05/16

Sistema de monitoreo de crecimiento de lunares para la detección temprana de cáncer de piel tipo melanoma basado en dispositivos móviles Android

Leonel Rosas Arias, Jesús Daniel Rojas Cid, Daniel Benito Román Ocampo, José António López Tello, Waldemar Pérez Bailón

Instituto Tecnológico de Lázaro Cárdenas, Departamento de Sistemas Computacionales, Cd. Lázaro Cárdenas, Michoacán, México {rosas.leonel95,danielbenito.roman}@gmail.com, drcid99@yahoo.com,{ltello021, waldemar1953}@hotmail.com

Resumen. El cáncer de piel de tipo melanoma es altamente curable si se detecta en una etapa temprana. El melanoma puede surgir en lunares expuestos a radiación UV por periodos prolongados de tiempo, cambiando su color, forma y tamaño. En este trabajo se presenta un sistema para el monitoreo del crecimiento y la apariencia de lunares en la piel como alerta de posible melanoma maligno. Primero, se realiza la segmentación de lunares en la imagen obtenida por el dispositivo movil haciendo uso de filtros en el espacio de color YCbCr. Después, se elimina el ruido restante en la imagen y se calcula el área del lunar. Finalmente, la fotografía, fecha de captura y área son almacenados en una base de datos SQL local para su posterior consulta. Los resultados muestran que al usar una referencia de longitud dada por el usuario se obtiene una exactitud promedio del 95.5 % respecto al área real del lunar.

Palabras clave: procesamiento de imágenes, dispositivos móviles, segmentación de color, cáncer de piel de tipo melanoma, software de seguimiento médico.

An Android Mobile Device-based Mole Growth Monitoring System for Early Detection of Melanoma Skin Cancer

Abstract. Melanoma skin cancer is highly curable if it is detected in early stages. Melanoma can arise in moles exposed to UV radiation for extended periods of time, changing its color, shape and size. In this paper, we present a monitoring system for mole's growth and appearance as an alert for possible malignant melanoma. First, we segment moles in the image taken by the mobile device by using filters in the YCbCr color space. Then, we remove residual noise in the image and calculate the mole area. Finally, the photograph, as well as the capture date and the area

are stored in a local SQL database for future consultation. Experimental results show that if a reference length is given by the user, the proposed system reaches an average accuracy of $95.5\,\%$ respect to the real mole area.

Keywords: image processing, mobile devices, color segmentation, melanoma skin cancer, medical follow-Up software.

1. Introducción

El cáncer de piel afecta a personas de todas las razas y grupos étnicos. El grado de detección de cáncer de piel en hombres hispanos se da en menor medida que en los hombres blancos no-hispanos. Por esta razón, los hombres hispanos son más propensos a ser diagnosticados con etapas avanzadas de cáncer [1,2]. El riesgo de desarrollar cáncer de piel se debe a la combinación de factores genéticos y ambientales [3]. La causa más común es la exposición prolongada a la luz ultravioleta [4].

El cáncer de piel se divide en melanoma maligno y cáncer de piel no-melanoma (NMSC por sus siglas en inglés) [3,4,5]. El melanoma representa el 4% de los casos de cáncer y es responsable directo del 75% de todas las muertes causadas por cáncer de piel [6].

El melanoma puede aparecer en un lunar pre-existente en la piel que se ha vuelto más obscuro o bien ha cambiado en apariencia, aunque es más común encontrarlo como un lunar nuevo o una peca inusual. Es altamente curable si se detecta en una etapa temprana [7] y se desarrolla en dos fases: Fase de crecimiento radial (sobre la epidermis) y fase de crecimiento vertical (hacia la dermis) [8].

Existe un factor genético conocido como el síndrome familiar de melanoma y lunares atípicos (FAMM por sus siglas en ingles). Pacientes con este síndrome son propensos a sufrir la mutación del gen CDKN2A, causante de melanoma [9]. Los niños que presentan esta condición son mucho más propensos a desarrollar este tipo de cáncer en edades tempranas [5] y por esta razón se recomiendan examinaciones caseras de manera regular, así como tener buenas prácticas de protección solar [10].

Existen diversos parámetros para diagnosticar melanoma. Algunos de estos son: Asimetría, irregularidad en los bordes, variación de color y diámetro, este último es controversial puesto que existen melanomas que son menores a 6 mm de diámetro [4]. Sin embargo, estudios recientes demuestran que sigue siendo un parámetro útil para diferenciarlos [11].

El cuidado y prevención del melanoma en México está por debajo de lo óptimo. A pesar de que el tratamiento es similar al de países desarrollados, se cree que la razón principal de un diagnóstico en etapas avanzadas es debido a la baja cultura de salud en la población. La mayoría de pacientes con melanoma no acuden a consulta médica hasta que los tumores son grandes, generan incomodidad o ya están causando un problema para su salud [12].

Detectar melanoma en pacientes con diversos lunares o lesiones melanocíticas benignas puede ser algo complicado incluso para los más experimentados dermatólogos [13,14]. El mapeo de lunares (del inglés *Mole Mapping*) es una técnica que involucra tecnología de imagen no invasiva para mejorar el monitoreo de lesiones melanocíticas nuevas o cambiantes. Tecnologías no invasivas de imagen han mostrado detectar melanoma en etapa temprana. El seguimiento médico digital es una tendencia que está revolucionando la manera en la que los pacientes participan en el cuidado y monitoreo de la salud [13].

Las aplicaciones para smartphone (apps) ofrecen la posibilidad de llevar un monitoreo dermatológico periódico. Existen diversas apps para la detección del cáncer de piel pero muy pocas han sido evaluadas como clínicamente eficientes. El 33 % de las apps menciona explícitamente el uso de dermatólogos certificados, el 31 % mencionan la intervención de un equipo médico, mientras que el 36 % no proporciona ningún tipo de información [15]. La mayoría de usuarios que descargan las apps no las usan regularmente, sin embargo, existen esfuerzos para generar conciencia en la población [16]. También se han realizado esfuerzos para enseñar a fisioterapeutas a identificar indicios de cáncer de piel en sus pacientes [17].

Algunas apps desarrolladas para llevar un seguimiento de lunares mediante fotografias son: UMSkin-Check, LoveMySkinMole, FotoSkin y Embarrassing Bodies -- My MoleChecker [15].

Trabajos recientes en sistemas para la detección de melanoma se describen en [18,19,20] obteniéndose resultados alentadores. En [20] se presenta el desarrollo de una app móvil capaz de identificar cáncer de piel en tiempo real utilizando algoritmos automatizados, sin embargo, los resultados de efectividad oscilan entre el $75\,\%$ y el diagnóstico toma alrededor de 15 segundos.

Trabajos en el área tele-dermatológica también han utilizado teléfonos móviles con cámara y acceso a internet para facilitar el diagnóstico de un dermatólogo de manera remota [21].

2. Segmentación de lunares en el espacio de color YCbCr

La segmentación de objetos mediante su color ha sido uno de los enfoques más comúnmente utilizados para la detección de objetos, trabajos relacionados se muestran en [22,23,24,25]. Para realizar esta tarea es posible utilizar diferentes espacios de color, tales como el espacio RGB, HSV, YCbCr, LAB, entre otros.

En este trabajo se ha decidido utilizar el espacio de color YCbCr debido a que, en base a experimentos realizados, este fue el espacio de color con el que se ha obtenido una segmentación exitosa de manera rápida e intuitiva.

Una imagen representada en el espacio de color YCbCr se expresa como la composición de tres canales:

$$I_{YCbCr} = (I_Y, I_{Cb}, I_{Cr}), \tag{1}$$

Leonel Rosas Arias, Jesús Daniel Rojas Cid, Daniel Benito Román Ocampo, et al.

donde I_Y es el canal correspondiente a la luma y I_{Cb} y I_{Cr} son los componentes de crominancia azul y crominancia roja, respectivamente. Esto se ilustra en la Figura 1.

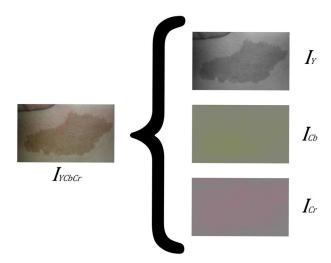


Fig. 1. Componentes de la imagen en el espacio de color YCbCr.

Las imágenes obtenidas por la cámara del dispositivo, haciendo uso de la librería OpenCV, son obtenidas en formato RGBA, donde R indica el canal rojo; G, el canal verde; B, el canal azul; y A, el canal alfa, correspondiente a la transparencia de la imagen. Para realizar la operación de segmentación en el espacio de color YCbCr, se realiza la conversión de los tres primeros canales de la imagen, es decir de RGB a YCbCr. Esto se obtiene mediante el siguiente modelo:

$$Y = 0.299R + 0.587G + 0.114B, (2)$$

$$Cb = 0.564(B - Y) + \Delta,\tag{3}$$

$$Cr = 0.713(R - Y) + \Delta,\tag{4}$$

donde

$$\Delta = \begin{cases} 128 & \text{para imágenes de 8 bits} \\ 32768 & \text{para imágenes de 16 bits} \\ 0,5 & \text{para imágenes de punto flotante} \end{cases} \tag{5}$$

Sistema de monitoreo de crecimiento de lunares para la detección temprana de cáncer de piel...

La segmentación de los lunares en el espacio de color YCbCr se realiza mediante una imagen máscara. Esta imagen contiene la localización de todos los pixeles en la imagen original que pertenecen al lunar. La imagen máscara es una imagen de tipo binaria que está dada por la siguiente condición:

$$I_{Mask}(x,y) = \begin{cases} 1 & C_Y \wedge C_{Cb} \wedge C_{Cr} \\ 0 & \text{En cualquier otro caso} \end{cases}$$
 (6)

donde

$$C_Y: Y_{min} \le I_{Mask_Y}(x, y) \le Y_{max},\tag{7}$$

$$C_{Cb}: Cb_{min} \le I_{Mask_{Cb}}(x, y) \le Cb_{max}, \tag{8}$$

$$C_{Cr}: Cr_{min} \le I_{Mask_{Cr}}(x, y) \le Cr_{max}. \tag{9}$$

 Y_{min} y Y_{max} son los valores mínimo y máximo del filtro para el canal Y. De la misma manera, los parámetros Cb_{min} , Cb_{max} , Cr_{min} y Cr_{max} representan los valores mínimos y máximos del filtro en sus respectivos canales.

La Figura 2 muestra el resultado de la segmentación de un lunar dorsal perteneciente a uno de los sujetos de prueba. Debido a que la tonalidad de color del lunar es muy similar a algunas otras secciones de la imagen, se producen falsos positivos. En la siguiente sección se aborda la metodología utilizada para eliminar estos falsos positivos con la intervención del usuario.

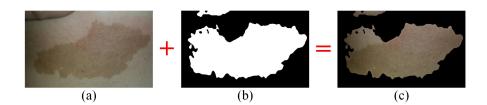


Fig. 2. Resultado de la segmentación del lunar. a) Imagen original. b) Imagen máscara.
c) Imagen segmentada.

3. Post-procesamiento

En esta sección se presenta la metodología utilizada para eliminar los pixeles falsos positivos en la segmentación del lunar, es decir, aquellos pixeles que, a pesar de haber pasado el filtro descrito en la sección anterior, no pertenecen al lunar. Este post-procesamiento es opcional, ya que, en algunos casos, el lunar

puede tener una tonalidad de color muy distinta en comparación a la piel, lo que permite una rápida segmentación sin la necesidad de llevarlo a cabo.

Se presentan dos métodos para resolver este problema: La eliminación de pixeles falsos positivos de manera selectiva y el rellenado dinámico de huecos en los lunares.

3.1. Eliminación de ruido en la imagen mediante gestos táctiles

El ruido generado por los pixeles falsos positivos puede ser eliminado selectivamente con ayuda del usuario. Para ello se implementa una estrategia de borrado similar al de los editores de imágenes. Se detecta la posición donde el usuario ha tocado la imagen y se dibuja un circulo de radio:

$$r_{circle} = \frac{I_{width}}{rel},\tag{10}$$

donde I_{width} es el ancho de la imagen en pixeles, y rel es un factor de escala. En este caso se ha hecho uso de rel = 20, para una relación de radio veinte veces menor al ancho máximo de la imagen.

El radio del círculo se establece de manera dinámica debido a que la resolución de la imagen mostrada en pantalla varía en relación a la resolución de la pantalla del dispositivo. Un valor de radio fijo podría causar alguna de a las siguientes situaciones:

- En un dispositivo con pantalla de mayor resolución a la del dispositivo de prueba, se borrará una región circular muy pequeña.
- En un dispositivo con pantalla de menor resolución a la del dispositivo de prueba, se borrará una región circular muy grande.

Este círculo coloca todos los pixeles dentro sí mismo a cero. Al colocar todos los pixeles a cero en los tres canales de la imagen, se elimina el ruido en la imagen. Este proceso se ilustra en la Figura 3.

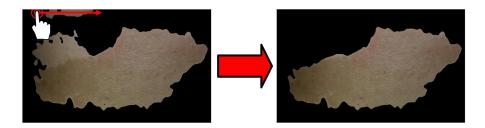


Fig. 3. Segmentación final del lunar con asistencia del usuario.

3.2. Rellenado dinámico de huecos en la imagen

Frecuentemente, al realizar el proceso de segmentación por color existen regiones o huecos dentro del objeto segmentado que no son detectados como parte del mismo. Esto se debe principalmente al reflejo de la iluminación o bien porque existen otros elementos de diferente color dentro del elemento de interés. Estos huecos producidos en el proceso de segmentación pueden ser rellenados para segmentar el objeto de interés en su totalidad. Para ello se utiliza una operación de cerradura morfológica entre la imagen binaria A (correspondiente a la imagen segmentada) y un elemento estructural B, es decir:

$$A \bullet B = (A \oplus B) \ominus B, \tag{11}$$

donde \oplus y \ominus indican un proceso de dilatación y erosión, respectivamente.

En base a resultados experimentales, se propone realizar diez procesos de dilatación seguido de diez procesos de erosión. En cada uno de estos procesos se utiliza un elemento estructural de tipo rectangular de tamaño $K \times K$, en donde K se ha establecido de la siguiente manera:

$$K = \begin{cases} 3 & I_{width} \le 300, \\ roundToNextOdd(\frac{I_{width}}{100}) & \text{En cualquier otro caso.} \end{cases}$$
 (12)

A su vez, la función roundToNextOdd evita que el elemento estructural posea un tamaño par o racional. Un kernel de tamaño par o racional produce errores al usar las funciones de dilatación y erosión de la librería OpenCV. La función roundToNextOdd redondea el valor del argumento y posteriormente aproxima el resultado al siguiente número impar en caso de no serlo. Este proceso se ilustra en la Figura 4.

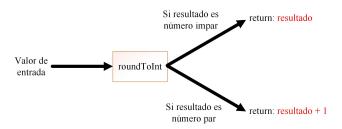


Fig. 4. Estructura interna de la funcion roundToNextOdd.

El proceso de llenado de huecos se ilustra en la Figura 5. Se han añadido diversos huecos de diferentes tamaños en la imagen segmentada del proceso anterior para comprobar la efectividad del método propuesto.

Leonel Rosas Arias, Jesús Daniel Rojas Cid, Daniel Benito Román Ocampo, et al.

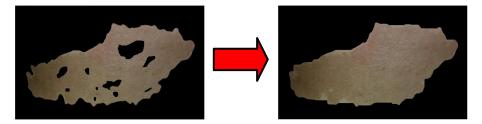


Fig. 5. Rellenado de huecos en la segmentación del lunar.

4. Determinación del área del lunar

La determinación del área del lunar se realiza contando todos los pixeles con valor diferente de cero en la imagen segmentada. Este método, aunque bastante fácil de aplicar, presenta un problema significativo. La distancia a la que se toma la fotografía del lunar debe ser consistente para evitar errores en la medición del área. No tomar la fotografía a una distancia consistente en todas las muestras supone alguno de los siguientes problemas:

- Al tomar la fotografía a mayor distancia, menor área es detectada (Un número menor de pixeles representa la misma área del lunar).
- Al tomar la fotografía a menor distancia, mayor área es detectada (Un número mayor de pixeles representa la misma área del lunar).

Para resolver este problema se pide al usuario proporcionar una medida de referencia entre dos puntos de la imagen. Estos dos puntos se alinean con dos líneas de guía verticales visibles durante el proceso de la captura de la imagen. Este proceso se ilustra en la Figura 6, donde la posición sobre el eje x de las líneas L_1 y L_2 son establecidas mediante:

$$L_1 = \frac{3(I_{width})}{8},\tag{13}$$

$$L_2 = \frac{5(I_{width})}{8}. (14)$$

La posición de las líneas L_1 y L_2 puede ser modificada por el usuario durante el proceso de captura de la imagen. Esto se describe en la siguiente sección.

Una vez establecida una medida de referencia y conociendo las posiciones de L_1 y L_2 se calcula el área del lunar como:

$$A = N_{pixels} \left(\frac{ref}{L_2 - L_1}\right)^2, \tag{15}$$

donde ref es la medida de referencia en milímetros y N_{pixels} es el número de pixeles diferentes de cero en la imagen.

Sistema de monitoreo de crecimiento de lunares para la detección temprana de cáncer de piel...

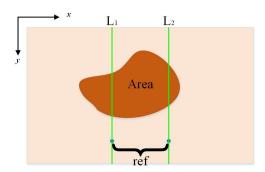


Fig. 6. Determinación del área del lunar con medida de referencia conocida.

5. Crecimiento del lunar

Al realizar un chequeo periódico del crecimiento de lunares es deseable establecer una métrica que indique el grado de crecimiento de los mismos. Para este fin, se propone utilizar la pendiente de la recta que mejor se ajusta a los datos históricos del área del lunar. Esta recta es obtenida mediante regresión lineal por el método de mínimos cuadrados. Los parámetros de la recta y=mx+b son determinados mediante:

$$m = \frac{\sum_{i=0}^{N-1} (\mathbf{x}_i - \bar{x})(\mathbf{y}_i - \bar{y})}{\sum_{i=0}^{N-1} (\mathbf{x}_i - \bar{x})^2},$$
(16)

$$b = \bar{y} - m(\bar{x}),\tag{17}$$

donde ${\bf x}$ es el vector correspondiente al área del lunar, ${\bf y}$ el vector correspondiente al número de muestra, $\bar x$ y $\bar y$ son la media cada uno de los vectores y N es el número de muestras.

El valor de la pendiente de la recta proporciona una referencia del crecimiento del lunar. Adicionalmente, el signo de la pendiente también proporciona información. Un signo positivo, indicaría un incremento del área lunar, mientras que, un signo negativo indicaría la reducción del mismo.

6. Resultados

En esta sección se presenta la aplicación desarrollada y se da una descripción de los elementos que la componen. También se presenta el desempeño del sistema en la medición de área de los lunares ante diversas condiciones. Las pruebas de desempeño del sistema se realizaron en un teléfono celular móvil Samsung Galaxy S5. Por otra parte, las pruebas de compatibilidad de software, se realizaron adicionalmente en los siguientes dispositivos: Samsung Galaxy S7, ZTE Blade

Leonel Rosas Arias, Jesús Daniel Rojas Cid, Daniel Benito Román Ocampo, et al.





Fig. 7. Módulo principal. a) Visualización general de las muestras. b) Visualización individual.

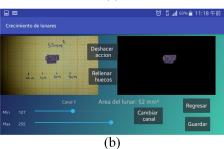
v6, Alcatel Pixi 4, Samsung Galaxy Grand Prime y Samsung Galaxy Grand Prime Plus.

6.1. Aplicación móvil

La aplicación está dividida en tres módulos:

- Módulo principal. Este módulo, mostrado en la Figura 7, despliega un listado de las muestras obtenidas que pueden ser ordenadas por fecha, de la más reciente a la más antigua y viceversa. Se muestra la fecha de captura, el área del lunar y la vista previa de la fotografía del lunar para cada una de las muestras. Al presionar sobre alguno de los elementos se muestra la fotografía en gran tamaño junto con sus datos correspondientes.
- Módulo de obtención de nueva muestra. En este módulo se realiza la adquisición, procesamiento y post-procesamiento de la imagen del lunar. Proporciona los controles necesarios para modificar los parámetros del filtro en el espacio de color YCbCr y la posición de las líneas de guía. Además, una vez adquirida la imagen, proporciona opciones para rellenar los posibles huecos de la segmentación y para deshacer el último cambio realizado. También se proporciona al usuario la posibilidad de eliminar el ruido de la imagen segmentada mediante gestos táctiles. Este módulo se muestra en la Figura 8.
- Módulo del historial gráfico. Este módulo muestra el gráfico del comportamiento del crecimiento del lunar. El gráfico muestra el número de muestra en el eje x y el área del lunar en el eje y. También se muestra la recta y = mx + b obtenida mediante regresión lineal que indica el crecimiento del lunar. Al tocar sobre alguna de las muestras del gráfico, se presenta un mensaje con su información respectiva. Esto se aprecia en la Figura 9.

Sistema de monitoreo de crecimiento de lunares para la detección temprana de cáncer de piel...



 ${\bf Fig.\,8.}$ Módulo de obtención de nueva muestra. a) Adquisición de la imagen. b) Segmentación y guardado.

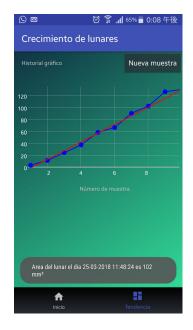


Fig. 9. Módulo del historial gráfico.

6.2. Desempeño del sistema en la medición de área

Para probar la efectividad de la medición de área del sistema se han utilizado hojas milimétricas. Estas hojas cuadriculadas con cuadros de 1 mm² son ideales para conocer de antemano el área de las superficies de prueba al contar los cuadros individuales. En las hojas se han dibujado superficies con un área aleatoria de entre 0 y 150 mm². Se han realizado 30 pruebas en 10 diferentes medidas de superficie. Las pruebas se realizaron con tres medidas diferentes de referencia previamente establecidas. Se ha determinado mediante el promedio del error de todas las muestras que la exactitud del sistema es de aproximadamente el 95,5 %. El resultado de estas pruebas se muestra en el gráfico de la Figura 10.

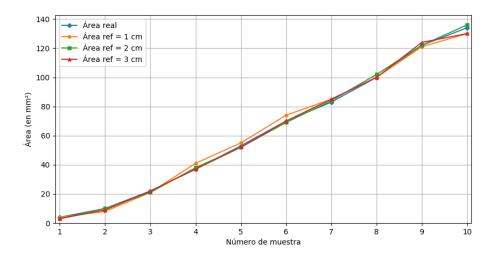


Fig. 10. Desempeño del sistema con diferentes medidas de referencia.

7. Conclusiones y trabajos futuros

En este trabajo se ha presentado una herramienta para el monitoreo y detección temprana de cáncer de piel del tipo melanoma a partir del crecimiento de lunares. Se ha presentado un método de segmentación rápida de lunares mediante su color haciendo uso de filtros en el espacio de color YCbCr.

Existen casos en donde el color de la piel es muy similar al color del lunar, por lo que, es necesario un mayor nivel de interacción con el usuario para eliminar las regiones de la imagen que no pertenecen al lunar. Este proceso de corrección manual, aunque reduce el grado de automatización del sistema, puede realizarse sin mayor dificultad de manera rápida. La determinación del área ha sido realizada exitosamente habiendo obtenido resultados de hasta el 100 % de exactitud respecto al área real del lunar. El margen de error por medición ha sido

de un par de pixeles, por lo que, en lunares con superficies pequeñas (menores a 10 mm^2) los errores porcentuales tienden a incrementar en comparación con lunares de superficies mayores. La determinación del área del lunar se ha visto afectada por diversos factores. Los dos factores principales son: La distorsión en la imagen causada por el lente de la cámara del dispositivo y el ángulo de captura de la imagen. Este último se presenta cuando la fotografía no es tomada de manera paralela a la superficie de la piel. A pesar de estos factores de error, el porcentaje de efectividad promedio del sistema ha sido del 95,5 %. Por esta razón, creemos que el sistema ofrece una medida confiable tratándose de un equipo no especializado.

En trabajos futuros se proyecta la adición de nuevas funcionalidades al sistema con el fin de realizar un diagnóstico oportuno de cáncer de piel utilizando técnicas de Machine Learning. Estas nuevas funcionalidades serán capaces de analizar el color, la forma y la textura del lunar de manera local y/o global para diferenciar entre melanoma maligno y otras lesiones melanocíticas benignas.

Agradecimientos. Trabajo de investigación apoyado y financiado por el Tecnológico Nacional de México/Instituto Tecnológico de Lázaro Cárdenas.

Referencias

- Zhou, J., Enewold, L., Peoples, G. E., McLeod, D. G., Potter, J. F., Steele, S. R., Clive, K. S., Stojadinovic, A., Zhu, K.: Colorectal Prostate Skin Cancer Screening Among Hispanic White Non-Hispanic Men 2000–2005. Journal of the National Medical Association, 103(4), pp. 343–350 (2011)
- 2. Agbai, O. N., Buster, K., Sanchez, M., Hernandez, C., Kundu, R. V., Chiu, M., Roberts, W. E., Draelos, Z. D., Bhushan, R., Taylor, S. C., Lim, H. W.: Skin cancer and photoprotection in people of color: A review and recommendations for physicians and the public. Journal of the American Academy of Dermatology, 70(4), pp. 748–762 (2014)
- 3. Chummun, S., McLean, N. R.: The management of malignant skin cancers. Surgery (Oxford), 35(9), pp. 519–524 (2017)
- 4. Craythorne E., Al-Niami F.: Skin cancer. Medicine, 45(7), pp. 431–434 (2017)
- 5. Maguire-Eisen, M.: Skin Cancer: A Growing Health Problem for Children. Seminars in Oncology Nursing, 29(3), pp. 206–213 (2013)
- Godoy, S. E., Ramirez, D. A., Myers, S. A., Winckel, G., Krishna, S., Berwick, M., Padilla, R. S., Sen, P., Krishna S.: Dynamic infrared imaging for skin cancer screening. Infrared Physics & Technology, 70, pp. 147–152 (2015)
- Reshma, M., Priestly Shan, B.: Review of Image Acquisition and Classification Methods on Early Detection of Skin Cancer. Proceedings of Fifth International Conference on Soft Computing for Problem Solving, pp. 1015–1025 (2016)
- 8. Holte, K., Biswas, A.: Pathology of malignant skin tumours. Surgery (Oxford), 35(9), pp. 478–483 (2017)
- Soura, E., Eliades, P. J., Shannon, K., Stratigos, A. J., Tsao H.: Hereditary melanoma: Update on syndromes and management: Genetics of familial atypical multiple mole melanoma syndrome. Journal of the American Academy of Dermatology, 74(3), pp. 395–407 (2016)

- 10. Hawryluk, E. B., Liang, M. G.: Pediatric Melanoma, Moles, and Sun Safety. Pediatric Clinics of North America, 61(2), pp. 279–291 (2014)
- 11. Glazer, A. M., Rigel, D. S., Winkelmann, R. R., Farberg, A. S.: Clinical Diagnosis of Skin Cancer: Enhancing Inspection and Early Recognition. Dermatologic Clinics, 35(4), pp. 409–416 (2017)
- Lino-Silva, L. S., Domínguez-Rodríguez, J. A., Aguilar-Romero, J. M., Martínez-Said, H., Salcedo-Hernández, R. A., García-Pérez, L., Herrera-Gómez, A., Cuellar-Hubbe, M.: Melanoma in Mexico: Clinicopathologic Features in a Population with Predominance of Acral Lentiginous Subtype. Annals of Surgical Oncology, 23(13), pp. 4189–4194 (2016)
- Berk-Krauss, J., Polsky, D., Stein, J. A.: Mole Mapping for Management of Pigmented Skin Lesions. Dermatologic Clinics, 35(4), pp. 439

 –445 (2017)
- Bresler, S. C., Wanat, K., Seykora, J.: Benign Melanocytic Lesions and Melanoma. Pathobiology of Human Disease, pp. 1182–1192 (2014)
- Chao, E., Meenan, C. K., Ferris, L. K.: Smartphone-Based Applications for Skin Monitoring and Melanoma Detection. Dermatologic Clinics, 35(4), pp. 551–557 (2017)
- 16. Noar, S. M., Leas, E., Althouse, B. M., Dredze, M., Kelley, Ayers, J. W.: Can a selfie promote public engagement with skin cancer? Preventive Medicine (2017)
- 17. Trotter, S. C., Louie-Gao, Q., Hession, M. T., Cummins, D.: Skin Cancer Education for Massage Therapists: A Novel Approach to the Early Detection of Suspicious Lesions. Journal of Cancer Education, 29(2), pp. 266–269 (2014)
- Kostopoulos, S. A., Asvestas, P. A., Kalatzis, I. K., Sakellaropoulos, G. C., Sakkis, T. H., Cavouras, D. A., Glotsos, D. T.: Adaptable pattern recognition system for discriminating Melanocytic Nevi from Malignant Melanomas using plain photography images from different image databases. International Journal of Medical Informatics, 105, pp. 1–10 (2017)
- 19. Dalila, F., Zohra, A., Reda, K., Hocine, C.: Segmentation and classification of melanoma and benign skin lesions. Optik International Journal for Light and Electron Optics, 140, pp. 749–761 (2017)
- Taufiq, M. A., Hameed, N., Anjum, A., Hameed, F.: m-Skin Doctor: A Mobile Enabled System for Early Melanoma Skin Cancer Detection Using Support Vector Machine. eHealth 360°, pp. 468–475 (2017)
- Walocko, F. M., Tejasvi, T.: Teledermatology Applications in Skin Cancer Diagnosis. Dermatologic Clinics, 35(4), pp. 559–563 (2017)
- 22. Garcia-Lamont, F., Cervantes, J., López, A., Rodriguez, L.: Segmentation of images by color features: A survey. Neurocomputing, 292, pp. 1–27 (2018)
- Hassanat, A. B. A., Alkasassbeh, M., Al-awadi, M., Alhasanat, E. A. A.: Color-based object segmentation method using artificial neural network. Simulation Modelling Practice and Theory, 64, pp. 3–17 (2016)
- Sikha, O. K., Kumar, S. S., Soman, K. P.: Salient region detection and object segmentation in color images using dynamic mode decomposition. Journal of Computational Science (In Press) (2017)
- Cernadas, E., Fernández-Delgado, M., González-Rufino, E., Carrión, P.: Influence of normalization and color space to color texture classification. Pattern Recognition, 61, pp. 120–138 (2017)

Algoritmos para el control de indicadores clave de desempeño para Smart Cities

Elsa Estrada, Adriana Peña Pérez Negrón, Graciela Lara López, Luis Casillas

Universidad de Guadalajara, CUCEI, Departamento de Ciencias Computacionales,
Guadalajara, Jalisco, México
{elsa.estrada, ciela.larea}@academicos.udg.mx,
{adriana.pena, luis.casillas}@cucei.udg.mx

Resumen. En cada ciudad emergen diversos y diferentes tipos de situaciones a las que la ciudad debe adaptarse, lo que implica un cambio continuo durante su proceso de evolución, mismo que involucra un gran número de variables. Este esquema dificulta la integración de metodologías y herramientas para medir los indicadores que permitan la toma de decisiones que lleven al mejoramiento de las condiciones de habitabilidad. La selección de indicadores clave de desempeño de una ciudad conforma un modelo. En este trabajo se presenta el diseño y un prototipo que implementa los algoritmos de un esquema general para el control de indicadores clave de desempeño para las Smart Cities. Este diseño permite además llevar un método sistemático en el quehacer de monitoreo y actualización, así como la comparación de diferentes reportes entre distintos modelos. El método aplicado se desarrolla en tres fases: estudio de modelos de métricas de ciudad, análisis del formato en el Open Data, y el diseño e implementación de algoritmos para procesamiento de objetos JSON en mongodb.

Palabras clave: smart cities, open data para evaluación de ciudades, documentos JSON de indicadores clave de desempeño, NOSQL.

Key Performance Indicators Algorithms for Smart Cities

Abstract. In each city diverse and different type of situations emerge, to which the city has to adapt itself. This implicates a continuous change on its evolution process that involves a number of variables. This scheme difficult methodologies and tools integration to measure the indicators that allows the decision making for the improvement of the living conditions. The selection of the key performance indicators of a city represents a model. In this work are presented the design and a prototype that implement the algorithms of a general scheme for key performance indicators for Smart Cities. Moreover, this design allows a systematic approach for monitoring and updating the key performance indicators, as well as comparing different reports of different models. The

applied approach is developed in three faces: analysis of the metrics model of a city, Open Data format analysis, and the design and implementation of algorithms for processing JSON objects in mongodb.

Keywords: smart cities, open data for cities evaluation, JSON documents of key performance indicators, NOSQL.

1. Introducción

Aunado al incremento de la población en las ciudades va el aumento en la demanda de recursos y servicios, y aparecen fenómenos como la falta de inclusión social e inequidad. Para mitigar estos problemas, las Smart Cities proponen el desarrollo de estrategias de mediciones que soporten la toma de decisiones, lo que supone el manejo de una cantidad indefinida de indicadores.

Chourabi, et al. [1], en su estudio: "Understanding Smart Cities: An Integrative Framework", definen el concepto de Smart City como una estrategia que emerge para mitigar los problemas a consecuencia del crecimiento de la población urbana y la rápida urbanización, cuyos indicadores clave de desempeño o kpi (por sus siglas del inglés: key performance indicator) que impactan en el bienestar son: la administración y organización, la tecnología, el gobierno, política, gente y comunidades, economía, infraestructura construida y medio ambiente natural. Mientras que el trabajo de Chourabi, et al. [1] se centra en la caracterización por kpi, Townsend [2] se enfoca en las operaciones y sus beneficios, de tal manera que se refiere a una Smart City como aquella que resuelve sus problemas a través del monitoreo de eventos y consumo de recursos mediante el uso de sensores.

Otro concepto muy relacionado con Smart City es el de "Ciudad Sustentable" [3], que enfatiza como factor preponderante a las Tecnologías de la Información y de la Comunicación (TIC) para la mejora, principalmente, de tres aspectos urbanos: 1) mejora de la calidad de vida, 2) la eficiencia de operación urbana y de servicios, y 3) el abastecimiento de las demandas económicas, sociales y ambientales.

Las Smart Cities surgen como medio estratégico para resolver problemas que afectan tanto a la sustentabilidad como a la calidad de vida, impulsando el desarrollo de instrumentos para monitoreo y evaluación de fenómenos urbanos. Ejemplo de dichos instrumentos son: los modelos de métricas para evaluar ciudades, así como los sistemas autónomos que implementan diversos tipos de componentes que incluyen al internet de las cosas (IoT por sus siglas del inglés: *internet of things*), algoritmos *data science* y sistemas inteligentes para la toma de decisiones, infraestructura SDN (Software Definido por la Red) o los estándares de transferencia 5G.

1.1. Modelos de métricas de ciudades

Las métricas son entonces las herramientas para facilitar la toma de decisiones y mejorar el desempeño en las ciudades de por ejemplo, la calidad de vida y el bienestar. Sin embargo estos son factores abstractos e intangibles de tal manera que no es posible medirlos directamente. Por lo que se apoyan en mediciones que se

expresan en indicadores, por ejemplo el porcentaje de población de 15 años de edad con primaria terminada, o bien el porcentaje de emisiones de gas efecto invernadero.

De acuerdo con Jones Lang Lasalle [4], las mediciones son una de las vías para evaluar la complejidad del cambio urbano y juzgar qué enfoques tienen éxito y cuáles no. Los índices que nos ayudan a evaluar son herramientas útiles para los gobiernos, negocios y organizaciones. Nos ofrecen oportunidades tanto para aprender sobre las fortalezas y deficiencias no detectadas, como para persuadir a la ciudadanía en realizar cambios para incrementar el bienestar. De tal manera que la utilidad de los índices se resume en 5 objetivos: 1) evaluación y diagnóstico, 2) comparación de condiciones, 3) apoyo y persuasión, 4) pedagogía, y 5) generar confianza en el vivir. Las Smart Cities por lo tanto están interesadas en medir la calidad de vida de una ciudad para identificar áreas de mejora y así dar soporte a la toma de decisiones. Con este fin se han presentado por un lado modelos de estructuración de las métricas o kpi, y, por otro lado, los gobiernos han puesto atención en instrumentos para conocer la opinión del ciudadano sobre los factores que producen su bienestar.

Entre los modelos más importantes están el de Boyd Cohen que en 2012 formó un comité compuesto por representantes de IBM, Smart City Expo, Smart Cities Council y el Center for Innovation in Cities, para medir a las ciudades, generando en el año 2014 un modelo de métricas con 6 kpi principales, con 18 indicadores en 62 subdivisiones [5]. De los 62 indicadores del modelo Cohen, 16 de ellos son mapeados a ISO 37120 [6] que abarca 17 kpis con 100 indicadores. El Global Power City evalúa y clasifica las mejores ciudades de acuerdo a 6 kpi que contienen 26 indicadores subdivididos en 70 [7]. Se han propuesto muchos otros modelos como el Índice de Desarrollo Humano [8] o el método de la huella ecológica que sirve para calcular los recursos que una ciudad consume y los residuos que produce.

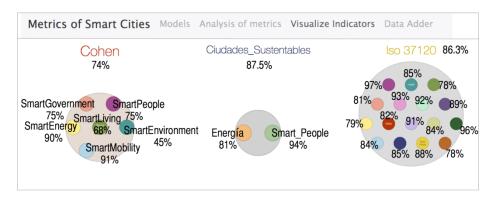


Fig. 1. Estructura general de la plataforma "Visualización de la herramienta Smart City".

En este documento se describe una herramienta que incorpora funciones de carga de datos (*Data Adder*), análisis y visualización de patrones (*Analisys of Metrics*) y un modelo (*Model*) que realiza el desplegado en modo árbol de cada indicador clave de desempeño o kpi con su fuente en la web. En este estudio se presentan los algoritmos que contemplan el almacenamiento de las mediciones para obtener una vista global de las condiciones de calidad de vida que asocian internamente cada kpi, lo que permite

la evaluación de una ciudad desde la perspectiva de diferentes modelos, conformando un modelo extendido. Este trabajo es parte de un proyecto global para medir la calidad de vida de la ciudad de Guadalajara, para el cual se ha desarrollado una herramienta de identificación de patrones entre los sectores territoriales basándose en el análisis de datos del Open Data y en una visualización geo-referenciada [9]. En la Fig. 1 se puede ver la parte gráfica de esta propuesta, aquí se muestra la visualización de kpis y sus valores para tres modelos: Cohen, Ciudades Sustentables e ISO 37120.

En las siguientes secciones se expone el método, su diseño e implementación con resultados desde la perspectiva del funcionamiento.

1.2. Método para el control de kpi (indicadores clave de desempeño)

Debido a la gran cantidad de variables que cada modelo implica, además de su tendencia al crecimiento, se observa la necesidad de una herramienta capaz de administrar sus subdivisiones o capas. Con ello posteriormente se podrá relacionar cada kpi hijo a un conjunto de datos, de tal manera que puedan ser analizados para obtener una medición tanto individual como general. El objetivo de este trabajo es por lo tanto, el de presentar un diseño completo con el desarrollo de algoritmos para la organización de kpis.

El proceso se realizó en tres fases: 1) primero se analizaron los diferentes modelos de métricas de ciudad; 2) posteriormente se examinaron las fuentes de datos locales en Open Data con el fin de identificar los formatos de registro en los archivos, para determinar el tipo de manejador de bases de datos apropiado (SQL o NoSQL), se seleccionó el Open Data por su formalidad y la veracidad en la información que se publica; 3) finalmente, en la tercera etapa se hizo el diseño y la implementación de algoritmos.

1.3. Formato JSON y mongodb

El formato JSON (*JavaScript Object Notation*) sirve para el intercambio y almacenamiento de datos y permite optimizar la velocidad de actualización y hacer archivos ligeros. Este formato utiliza convenciones familiares para los programadores, tales como, por ejemplo, llaves "{}" para agrupación y símbolos para anidar elementos u objetos que a su vez pueden contener a otros objetos [11]. Uno de los campos que más utiliza estos esquemas es el de los negocios, dado que muchas organizaciones buscan obtener ventaja del más mínimo dato. Aunque grandes volúmenes de datos se encuentran almacenados en sistemas diferentes, éstos pueden ser intercambiados integral y velozmente.

Comparado con XML, otro estándar común para el intercambio, el esquema JSON es más fácil de analizar gramáticamente, su formato para la definición de pares llave/valor requiere menor carga de bytes tanto para la transferencia entre sistemas, como para el almacenamiento y su indexación, JSON además posee mejor desempeño, incluye tipos de datos y arreglos, y es más veloz [12].

Aunque aún prevalece la necesidad de manejadores de bases de datos relacionales (RDB por sus siglas del inglés: *relational data base*) en los que existe una única llave para cada registro, con la normalización se decremento la redundancia reduciendo el

costo de almacenamiento [13]. Hoy en día se reconoce información sin una estructura definida en la que es más relevante el análisis científico, su velocidad de transferencia y facilidad de procesamiento, es decir manejadores de documentos JSON.

La aparición del NoSQL (originalmente referido como "non SQL" o "non relational") lenguaje de consulta para bases de datos no estructuradas, comenzó en el 2007 [14], período en el cual las exigencias de las organizaciones se dirigen al tratamiento y análisis de grandes cantidades de datos y no a las relaciones normalizadas, estos datos son escasamente definidos en sus tipos, lo que dio a lugar al desarrollo de plataformas orientadas a documentos como mongodo que maneja el esquema JSON, tomando en cuenta la escalabilidad resolviendo el problema del aumento continuo de datos e incompatibilidad de formatos entre sistemas [15]. Es por esto, que en nuestro caso de aplicación, en el que se hace intercambio de documentos entre el sistema local y la web, y el volumen de kpis tiende al Big Data, se seleccionó el formato JSON.

2. Modelo en capas

Una vez hecho el análisis de los modelos aplicados para medir la sustentabilidad de las ciudades, la calidad de vida, el desarrollo humano o la capacidad de producir lo que consume, se observó que estos suelen ser subdivididos para expresar mejor precisión en la medida, se identificó una estructura anidada y la formación de grupos en niveles, que se denominaron capas.

También se encontró que en Open Data no existe un estándar en la definición de los tipos de kpi, ni en sus rangos ni en las unidades de medida, esto es, sin estructura con la forma key-value.

2.1. Interfaz

Un primer prototipo para la implementación de navegación en capas se presentó en [1], en donde la interfaz fue inicialmente orientada al almacenamiento de fuentes de los datos (Fig. 2).



Fig. 2. Interfaz básica para la captura de capas (kpis) extraída de [9].

Dicha interfaz es aquí extendida en cuanto a su funcionalidad, mediante los elementos que se describen en la Tabla 1.

Tabla 1. Funciones de elementos de pantalla.

Tipo	Acción		
Caja de	Captura del nombre con el que se identifica al		
Texto	modelo de métricas urbanas		
Botón	Guarda en estructura temporal el Nombre del		
	Modelo capturado		
Caja de	Nombre del kpi que se desea anidar		
Texto			
	Guarda en estructura temporal el Nombre del kpi		
Botón	anidado en la capa activa con el padre y valor		
	inicializado en null		
Botón	Regresar a un nivel del modelo		
Botón	Navegar a un nivel superior		
Dotón	Eliminar una capa y sus kpis hijos		
DOION			
Botón	Convertir a JSON y enviar a servidor mongodb		
	Caja de Texto Botón Caja de Texto Botón Botón Botón Botón Botón		

2.2. Navegador en capas

Para un monitoreo inteligente, por ejemplo, en consumo de energía para determinar la demanda de recursos, el proceso a seguir en esta herramienta es mediante los siguientes pasos: a) identificar el modelo de métricas de ciudades sustentables y agregarlo, b) si existen, agregar las capas o kpis, y c) agregar los indicadores asociados y anidarlos al nivel o capa correspondiente. En la Fig. 3 se hace una comparación de dos modelos: uno internacional, el ISO37120, y uno local nacional mexicano. Se debe hacer un mapeo o selección de los kpis que se desean mantener de cada modelo.

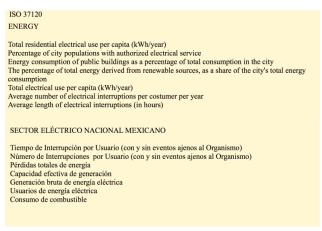


Fig. 3. Listados para kpis de Energía del estándar internacional ISO37120 y del modelo nacional mexicano [10].

En la Fig. 4 se puede ver la pantalla de captura. En este ejemplo se capturó el nombre de un kpi de cada modelo, es por esto que aparecen ambos en el mismo nivel. Al navegar por la Capa Anterior aparecerá el nombre de la capa en la que se han anidado.

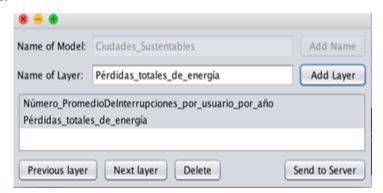


Fig. 4. Dos kpi's añadidos a la rama padre Pérdidas_totales_de_energía.

2.3. Estructura de documentos con kpis

Al ser enviadas las capas al Servidor se generan cuatro documentos: uno para la identificación del nombre del modelo, el cual tiene una variable value con valor null y que queda disponible para almacenar en su momento la medida de porcentaje de Sustentabilidad de una ciudad, a medida que se actualicen los valores de sus diferentes capas (Fig. 5).

En este caso el id Energía tiene un documento padre Ciudades_Sustentables, ver Fig. 6. Al igual que el anterior, para cada documento la variable value inicializada en null, variable servirá para la colocación de la medida de evaluación de Energía cuando sus subcapas o documentos hijos se hayan evaluado. En la Fig. 7 se muestran dos documentos de kpis pertenecientes a Energía, pero ya dentro del modelo Ciudades_Sustentables.

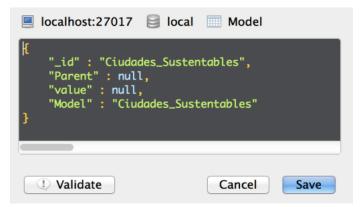


Fig. 5. Objeto JSON para la primera capa del modelo de Ciudades_Sustentables.

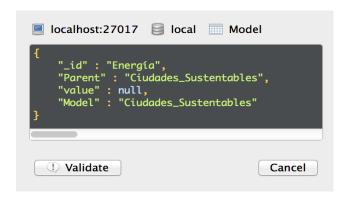


Fig. 6. Objeto JSON para la segunda capa, kpi Energía asociada a la rama padre Ciudades_Sustentables.

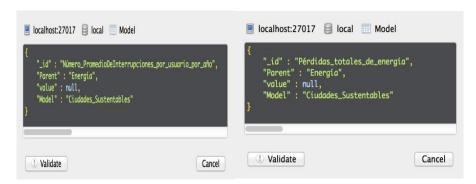


Fig. 7. Dos objetos JSON para la tercer capa, cada indicador asociado a la rama padre Energía.

Veamos otro ejemplo con la captura de la capa Smart People e indicadores clave de desempeño de la subcapa Educación, en la que previamente se fusionaron varios modelos de métricas entre ellos Cohen e ISO37120.

Las métricas de la calidad de la educación PISA aportan indicadores clave de desempeño internacional; además los indicadores clave de desempeño nacionales incluyen la infraestructura y cobertura.

De modo que Educación es evaluada en torno a cuatro aspectos: 1) alumnos en nivel licenciatura y la calidad de universidades a las que asisten, 2) alumnos en nivel de preparatoria y calidad de escuelas preparatorias, 3) alumnos en nivel de escuelas secundaria y calidad de escuelas secundarias, y 4) alumnos en nivel de escuelas primarias y calidad de escuelas primarias. En la Fig. 8 se muestra un árbol con los kpsi considerados, cada uno en sus ramas.

En la Fig. 9 se muestran dos pantallas, de una capa y la subsiguiente, con la interacción de la interfaz para captura del árbol de indicadores de Educación en la rama Smart People, Ciudades_Sustentables, un modelo híbrido ya que mezcla diversos índices, de Educación desde el modelo Cohen, ISO37120, PISA e indicadores de desempeño nacionales.

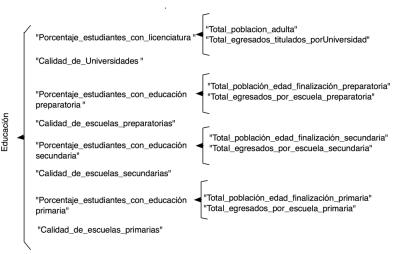


Fig. 8. Kpis para la medida de evaluación de Educación combinando indicadores de Educación de ISO37120 e indicadores de Educación de INEGI.

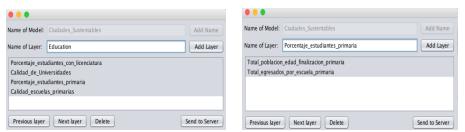


Fig. 9. Captura de kpis para la capa Educación y para su rama Porcentaje_estudiantes_primaria combinando indicadores de ISO37120 e INEGI.

Los documentos JSON resultantes almacenados en mongodo se muestran en la Fig. 10 en tres capas subsecuentes.

2.4. Algoritmos del navegador de capas en el árbol de kpis

El algoritmo para la función Capa Siguiente asigna como padre al kpi seleccionado en la interfaz de la pantalla y actualiza los datos sobre esta (Fig. 11).

```
private void btnNextActionPerformed(java.awt.event.ActionEvent evt) {
    if (listKpiTree.getSelectedIndex() != -1) {
        //Get the actual Kpi
        actualParentKpi = listKpiTree.getSelectedValue();
        //Update the gui
        updateGui();
    }
}
```

Fig. 10. Algoritmo de la función Capa Siguiente.

El botón Capa Anterior busca el padre del kpi seleccionado y extrae el id actividades correspondiente, dichas se ejecutan por el btnBackActionPerformed (Fig. 12), que realiza dos acciones retornar al padre (returnTree) y actualizar la interfaz (updateGui). El código de returnTree dispara a getParent con parámetro actualParentKpi, quien a su vez busca al el padre de la capa que en la interfaz ha sido seleccionada. getParent itera en kpiList comparando el hijo o parámetro con alguno de la columna 1 que corresponde al hijo, retornando por ello la columna 0 que corresponde al padre, si no lo encuentra entonces se trata de la raíz cuyo padre es él mismo y es cabecera de la estructura.

```
private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {
    //Return the tree
    returnTree();
    //Update the gui
    updateGui();
}

private void returnTree() {
    //Get the actual parent of the actual kpi
    actualParentKpi = getParent(actualParentKpi);
    }

private String getParent(String childString) {
    for (String[] iterateArray : kpiList) {
        if (iterateArray[1].equals(childString))) {
            return iterateArray[0];
        }
     }
     return headTree;
}
```

Fig. 11. Botón Capa Anterior llamando returnTree() quien dispara a getParent.

El algoritmo para la función Enviar a Servidor (Fig. 13), llama al método convertDocument () que toma la lista de kpis tipo árbol y los retorna en forma de cadena de caracteres en la estructura de documentos JSON e inserta cada uno mediante el ciclo for en la colección de objetos de mongodo por la conexión realizada en otro método.

```
private void btnSendServerActionPerformed(java.awt.event.ActionEvent evt) {
  for (String temporalString : convertDocument().split("\\n")) {
     DBObject dbObject = (DBObject) JSON.parse(temporalString);
     collection.insert(dbObject);
  }
}
```

Fig. 12. Función Enviar a Servidor.

El algoritmo del método convertDocument (Fig. 14) tiene dos actividades, la primera es generar el documento JSON para el objeto raíz, el cual no posee padre, el

id es el nombre del modelo guardado en la variable headTree, se asigna a la variable value el valor null reservada para futuro almacenamiento de la evaluación final global.

La segunda actividad es generar los documentos de cada kpi dentro de todas las anidadas. Itera en el contenido de la lista, agregando el id del padre que le pertenece a cada uno y que se encuentra en la misma lista. Añade el nombre del modelo que es headTree y la variable value con valor en null.

El formato de dos columnas de kpiList permite almacenar en la posición 0 el padre, y en la posición 1 el nombre del indicador.

```
private String convertDocument() {
    //A stringbuilder to generate the values
    StringBuilder stringBuilder = new StringBuilder();
    //Append the head
    stringBuilder.append("{'_id' : "'}).append(headTree).append("','Parent' : null,'value' : null,'Model' :
"').append(headTree).append(""});
    stringBuilder.append("\n");
    //Iterate the tree
    kpiList.stream().forEach((iterateArray) -> {
        stringBuilder.append("{'_id' : "'}).append(iterateArray[1]).append("','Parent' :
"').append(iterateArray[0]).append("','value' : null,'Model' : "').append(headTree).append("'});
    stringBuilder.append("\n");
    });
```

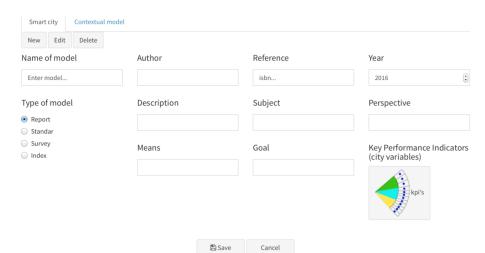
Fig. 13. Método convertDocument.

2.5. Interfaz para el modelo de métricas

El esquema general de control de kpis de ciudad debe comprender además la interfaz y funciones para fundamentar cada modelo, con él es posible para el usuario establecer el alcance de las métricas, el alcance de los resultados a obtener al aplicar el modelo, y además producir un marco de referencia que conduzca a la elección de un modelo adecuado a un contexto de ciudad, para ello se propone un diseño preliminar de interfaz que complementa a tal esquema (Fig. 15).

Las funciones principales se concentran en Crear nuevo modelo, Editar y Borrar (New, Edit y Delete), con las opciones para modelos concretos como *Smart Cities*, ISO37120 y modelos de contexto o híbridos que mezclan kpis de distintos autores. En esta interfaz se realiza así la especificación de la fuente del modelo, como autor, año y referencia bibliográfica, se especifica su tipo (reporte, estándar, cuestionario o índice), y la meta, perspectiva y sujeto para especificar la perspectiva desde el cual deberán ser interpretados los resultados.

El elemento KeyPerformanceIndicators enlaza a la interfaz de captura para el manejo de capas en la perspectiva de árbol, que es el prototipo funcional expuesto ampliamente en este estudio.



Elsa Estrada, Adriana Peña Pérez Negrón, Graciela Lara López, Luis Casillas

Fig. 14. Interfaz para el Modelo de Métricas.

3. Conclusiones y trabajo futuro

Debido a la gran cantidad de variables que manejan los modelos Smart City es necesaria una herramienta que nos permita mostrar los diferentes indicadores clave de desempeño o kpi de manera ordenada y lógica. En este documento se propone un manejo por capas tipo árbol para los diferentes kpi que maneja un modelo. La herramienta permite la visualización de diferentes modelos para facilitar su comparación. La determinación del modelo es fundamental para conocer mejor el estado de una ciudad.

El método presentado y sus algoritmos permiten además crear un modelo propio para compararlos con los ya existentes o con otros modelos creados.

Como trabajo a futuro se desarrollarán los algoritmos para la plataforma integral "visualization of the Smart City tool", entre ellos se encuentran principalmente los de despacho de recursos y servicios en modo inteligente, que deberán incorporar funciones de monitoreo inteligente basándose en el análisis científico de kpis capturados, para obtener no sólo la evaluación de una ciudad sino también reflejar y ejecutar las acciones para impactar positivamente en la huella ecológica y la sustentabilidad.

Referencias

- Chourabi, H., Nam, T., Gil-Garcia, J.R., Mellouli, S., Pardo, T.A., Scholl, H.: Understanding Smart Cities: An Integrative Framework. In: 45th Hawaii International Conference on System Sciences, pp. 1–9 (2011)
- 2. Townsend, A.M.: Smart Cities (2013)
- 3. ITU: General specifications and KPIs. Pp. 1–34 (2012)
- 4. Moonen, T., Clark, G.: The Business of Cities 2013. Jones Lang Lasalle IP, INC (2013)

- 5. Cohen, B.: Boyd Cohen, https://www.smart-circle.org/smartcity/blog/boyd-cohen-the-smart-city-wheel/
- 6. 37120, S.I.D.: ISO37120
- 7. Institute for Urban Strategies the Mori Memorial Foundation: Global Power City 2017. (2017)
- 8. Eclogical Footprint. 1-114 (2016).
- Estrada, E., Maciel, R., Ochoa, A., Bernabe-Loranca, B., Oliva, D., Larios, V.: Smart City Visualization Tool for the Open Data Georeferenced Analysis Utilizing Machine Learning. International Journal of Combinatorial Optimization Problems and Informatics. 9, pp. 25–40 (2018)
- 10. SENER ed: Sector Eléctrico Nacional. SENER.
- 11. International, E. ed: Introducing JSON. ecma international.
- 12. Gaurav, G., Karanjit, S., Ramkumar, K.R.: A detailed analysis of data consistency concepts in data exchange formats (JSON & XML) Presented at the March 21 (2017)
- 13. Codd, E.F.: A Relational Model of Data for LArge Shared Data Banks. Presented at the June 15 (1970)
- 14. Leavitt, N.: Will NoSQL Databases Live Up to Their Promise? (2010)
- 15. Inc, M.: MongoDB nosql document database.

Hacia un sistema de seguridad y alerta temprana de riesgos basado en agentes cognitivos

Wulfrano Arturo Luna-Ramírez^{1,2}, Anasol Peña-Rios², Emmanuel Ferreyra², Gustavo Ortiz Hernández³, Carlos R. Jaimez-González¹

¹ Universidad Autónoma Metropolitana, Unidad Cuajimalpa, Ciudad de México

{waluna,acpena,eferre}@essex.ac.uk,gustortiz@uv.mx

Resumen. Los desastres ya sea naturales o antrópicos son un peligro recurrente, ya que son causantes de disturbios y daños a la infraestructura pública y la propiedad privada de los ciudadanos. Los desastres, en combinación con diferentes grados de vulnerabilidad, tanto de individuos como de comunidades, configuran una gama de riesgo de sufrir los efectos negativos de estos eventos, que incluyen la muerte de personas. Desgraciadamente, muchos de estos fenómenos se caracterizan por su impredictibilidad. Sin embargo, la tecnología actual promete avanzar en la identificación de las causas y circunstancias asociadas a la ocurrencia de desastres. Hoy en día, mediante el uso de diferentes tecnologías, es factible obtener información útil para ciudadanos y autoridades con el fin de evitar riesgos y minimizar el impacto de un evento destructivo. Tal es el propósito de los Sistemas de Seguridad y Alerta Temprana de Riesgos. En este artículo se presenta una propuesta de un sistema de alerta temprana basado en agentes heterogéneos (cognitivos y reactivos) que explota el potencial de los dispositivos móviles, basándose en un diseño centrado en el usuario.

Palabras clave: sistemas de seguridad y alerta temprana de riesgos, sistemas multi-agente, diseño centrado en el usuario, agentes BDI.

An Early Warning Risk and Safety System Based on Cognitive Agents

Abstract. Disasters, whether natural or anthropic, are a recurring danger causing, disturbances and damage to public infrastructure and private property. Different degrees of vulnerability, both of individuals and of communities, combined with disasters, bring about negative effects of these events to the population, including the death of people. These phenomena are characterized by their unpredictability. However, current technology is promisory in advancing towards the identification of the causes and circumstances associated with the occurrence of disasters. Nowadays, technology supports the gathering of useful information for

² School of Computer Science & Electronic Engineering, University of Essex

 $^{^3}$ Universidad Veracruzana, Facultad de Ciencias Agrícolas, Xalapa, México $\{ {\tt wluna,cjaimez} \} {\tt @correo.cua.uam.mx},$

citizens and authorities. Early Warning Risk and Safety Systems are computational systems focused on avoiding risks and minimize the impact of a disruptive event. This paper presents a proposal for an early warning system based on heterogeneous agents (cognitive and reactive) relying on a user-centered design that exploits the potential of mobile devices.

Keywords: early warning risk and safety systems, multi-agent systems, user-centered design, BDI agents.

1. Introducción

A lo largo de la historia han ocurrido de manera cotidiana distintos desastres, tanto antrópicos (por intervención humana) como naturales (resultado de la propia actividad de los elementos y la evolución normal del ambiente). En la actualidad, nada nos exenta de sufrir los efectos de estos eventos. Muestra de ello son los huracanes, incendios, inundaciones y terremotos que han tenido lugar en distintas latitudes alrededor del globo, incluyendo territorio mexicano.

De acuerdo con las Naciones Unidas [20,23] los desastres representan distintos niveles de *peligro* (un evento natural o humano potencialmente dañino para la población y la infraestructura, disruptivo de la economía y degradante del medio ambiente) y *riesgo* (la probabilidad de sufrir consecuencias negativas o pérdidas inesperadas) para el ser humano tanto en las ciudades como en el campo. El grado de *vulnerabilidad* de comunidades o individuos, derivado de sus condiciones socioeconómicas y de infraestructura así como su exposición a desastres, determina sus posibilidades de permanecer a salvo y evitar sufrir efectos negativos.

El dominio de respuesta a la crisis se caracteriza por ser un entorno que requiere de control distribuido, y que involucra gran cantidad de datos, incertidumbre, ambigüedad, con múltiples partes interesadas, con diferentes objetivos y recursos limitados que varían continuamente [1]. Pese a que se han desarrollado sistemas de detección y prevención de riesgos, conocidos como Sistemas de Seguridad y Alerta Temprana de Riesgos⁴ (SSATR) [2,7,8,9,23], aún existe una brecha en la efectividad de los sistemas para salvar vidas de ciudadanos. Las condiciones de inseguridad se relacionan con el inadecuado nivel de capacidades técnicas (recursos, conocimiento y servicios de alerta operativos) que persiste mayormente en ciertos sectores y comunidades, particularmente en los países en vías de desarrollo (aunque este fenómeno no es solamente exclusivo de ellos). Una de las recomendaciones para eliminar esta brecha es el desarrollo de sistemas de alerta temprana integrados con los recursos ya existentes (sistemas actuales, bancos de datos, metodologías de minimización de riesgos) para que aúnen a las capacidades de éstos; diseñados con una perspectiva centrada en la gente, cuyo alcance vaya de lo comunitario a lo nacional.

⁴ Del Inglés Early Warning Detection and Safety Systems (EWDSS)

Un SSATR completo y efectivo, involucra cuatro elementos interrelacionados [23]:

- 1. Detección v valoración de riesgos
- 2. Servicios de monitoreo y alerta
- 3. Diseminación y comunicación
- 4. Capacidad de respuesta

En [13], los autores refieren un esquema integrado de trabajo para dar respuesta a emergencias (denominado iERF ⁵) que involucra tres componentes a analizar: los desastres (de origen natural o humano), los agentes involucrados (cuerpos de rescate y seguridad), y la aplicación computacional para dar respuesta a las crisis (planeación, análisis de vulnerabilidad, identificación y detección, etc.). De esta forma, un sistema de respuesta a emergencias debe ser capaz de contender con las características de una situación de crisis, a saber:

- Sobrecarga de los recursos e infraestructura de combate a crisis (estando sujetos a daños).
- Respuesta inmediata, debido a que los sucesos son altamente impredecibles.
- Gestión de la incertidumbre.
- Capacidad de trabajar con información escasa e incompleta.
- Anticipación y contención de crisis secundarias.

Los autores enfatizan que el sistema debe basarse sobre todo en la información del dominio de aplicación para así aumentar su efectividad [13]. Es esencial que sean descentralizados, con capacidad de coordinación de recursos, fusionar datos e integrar múltiples sistemas, adaptarse frente a cambios ambientales y con aprendizaje.

En este artículo se presenta una propuesta de SSATR basada en agentes artificiales heterogéneos, tanto reactivos como cognitivos, cuyo fin es identificar situaciones potenciales de desastre para proporcionar al usuario final (ya sea individuos o cuerpos de rescate y seguridad) información útil para ponerse a salvo, y/o facilitar la intervención y las actividades de rescate y salvamento. El sistema que se propone incorpora el uso de dispositivos móviles y su diseño considera una perspectiva centrada en el usuario.

El artículo está organizado como sigue: En la Sección 2 se hace un breve repaso a otros trabajos relacionados. la Sección 3 describe el problema de investigación además de proporcionar los conceptos básicos para entender el sistema propuesto. La Sección 4 se enfoca en explicar el SSATR propuesto y sus elementos constitutivos. Para finalizar, la Sección 5 presenta algunas reflexiones finales, así como propuestas de trabajo futuro.

2. Trabajo relacionado

La detección y prevención de desastres naturales y de origen humano ha ocupado tanto a académicos como a instituciones gubernamentales nacionales

 $^{^{5}}$ Por sus siglas en Inglés $integrated\ Emergence\ Response\ Framework$

e internacionales. Actualmente, la tecnología permite el uso de sensores y sistemas de detección tanto en la infraestructura (sensores ambientales) como en dispositivos móviles (sensores personales). A medida que la cantidad de datos disponibles aumenta, los sistemas automatizados pueden ayudar a gestionar el flujo de datos para facilitar la labor del usuario.

Por ejemplo, en cuanto a detección de sismos se refiere, los trabajos existentes generalmente realizan un procesamiento a partir de la información sísmica recolectada, principalmente de acelerómetros y acelerógrafos de las centrales sismográficas [2,7,8,9]. Tales dispositivos están conectados a un sistema de radiocomunicación encargado de, en su caso, diseminar las noticias hacia una central de control y registro. Las centrales gestionan los avisos para generar señales de alerta a la población mediante radio y televisión, y activar las alarmas sonoras y luminosas instalados en las edificaciones. En este escenario se ejemplifica el uso de sensores ambientales para recolectar información contextual, y movilizar a grandes cantidades de personas.

De manera complementaria, en [11], los autores presentan un sistema para monitorear la comunicación y coordinación de brigada de extinción de incendios usando teléfonos inteligentes. Similarmente, en [6], los autores presentan un escenario para equipos portátiles basados en micro y nanotecnología integrados en un uniforme "inteligente" que portan operadores de emergencias (por ejemplo, bomberos y rescatistas). El objetivo de estas prendas es monitorear los parámetros fisiológicos, la posición y la postura de los operadores, así como la presencia de posibles fuentes externas de peligro, y enviar estos datos a una unidad de coordinación remota. Estos ejemplos ilustran el uso de sensores personales para coordinar y movilizar grupos pequeños de personas. Por lo tanto, es importante considerar la adaptabilidad de los SSATR de acuerdo a la situación de riesgo y a la tecnología disponible.

El uso de agentes artificiales heterogéneos para el desarrollo de SSATRs ha sido propuesto con diferentes enfoques. Por ejemplo, en [17], los autores se basan en el sistema inmunológico humano combinado con algoritmos genéticos para explotar sus características adaptativas en el desarrollo de sistemas de gestión de emergencia e identificación del riesgo. Los autores mencionan que su sistema esta implementado basándose en agentes, sin embargo, omiten los detalles de la arquitectura de éstos, sus perfiles y mecanismos de aprendizaje y comunicación, dando mayor peso a su enfoque inmunológico.

En [13], se realiza un análisis del diseño de varios SMA enfocados en la simulación de respuesta a crisis. Cabe señalar que las implementaciones examinadas (DrillSim [3], DEFACTO [15], ALADDIN [1], RoboCup Rescue [14] y FireGrid [5]) están más centradas en la simulación de sistemas y no propiamente al apoyo in situ o durante el evento. Además, los autores mencionan que los sistemas analizados adolecen de nula tolerancia a fallos, rigidez y ausencia de aprendizaje y/o adaptación.

Algunos de estos sistemas aplican el enfoque de Diseño Centrado en el Ser Humano⁶, el cual tiene como objetivo hacer que los sistemas interactivos sean

⁶ del Inglés Human-Centered Design (HCD)

más útiles a las necesidades del usuario; aplicando conocimientos y técnicas de factores humanos, ergonomía y usabilidad [12]. Los sistemas centrados en el usuario pueden proporcionar una serie de beneficios que incluyen una mayor productividad, mayor bienestar para el usuario, prevención del estrés, mayor accesibilidad y menor riesgo de daños [12].

De este modo, el sistema propuesto en este documento busca tomar ventaja de la tecnología de detección antes mencionada, el Diseño Centrado en el Ser Humano, junto con las posibilidades de telecomunicación actuales, explotando los beneficios que ofrecen los Sistemas Multi-Agente para desarrollar de forma efectiva sistemas de alerta temprana de riesgos de desastres naturales y antrópicos.

3. Definición del problema

La Tabla 1 muestra un resumen del análisis de trabajos relacionados presentado en la Sección 2. Tal y como fue reportado en [13], muchas de las propuestas se podrían categorizar de acuerdo con la funcionalidad del sistema en: (i) sistemas que se enfocan en manejar un tipo de crisis específico, y (ii) sistemas que se enfocan en integrar subsistemas para construir un marco de respuesta y gestión de crisis.

El presente estudio se ubica dentro de la categoría (ii) antes mencionada, y se enfoca en generar un modelo de interacción entre sistemas y usuarios escalable a cualquier situación de desastre. Se busca abstraer las soluciones aplicables a los distintos tipos de amenazas a partir del estudio de las herramientas existentes, con el fin de extenderlas y generar un diseño de la respuesta a cualquier situación de desastre que considere a los usuarios y sus interacciones como su guía principal.

De esta manera, el proyecto se centra en resolver las siguientes preguntas de investigación:

Dado un conjunto de *peligros* identificados, dependiendo de las circunstancias particulares de un usuario, ¿es posible determinar automáticamente el grado de *vulnerabilidad* para evaluar el *riesgo* en que se encuentra?; y si es así, ¿es posible implementar un protocolo de detección de riesgo, que identifique el tipo y grado de riesgo que dispare procesos de alerta para el usuario y otros componentes del sistema, proporcionando información de seguridad oportuna?

Para resolver estas preguntas, el proyecto se enfoca en el desarrollo de un sistema para detección y alerta oportuna de riesgo derivado de desastres naturales o antrópicos que proporcione apoyo a la toma de decisiones para dos tipos de usuarios: víctimas potenciales, y autoridades y equipos de rescate. Para ello, nuestro enfoque se basa en el uso de agentes artificiales cuyos objetivos son:

 Mejorar la seguridad de los usuarios en condiciones peligrosas, proporcionando información útil para evitar el peligro o minimizar el riesgo.

Tabla 1. Trabajos relacionados y sus principales características.

Sistema(s)	Características	Categoría	Notas
[6,11]	Sistemas basados en dis-	(i)	Usan información física (p.ej.
	positivos prendables pa-		ubicacion) y fisiológica pro-
	ra detección de peligro.		porcionada por los dispositivos
			prendables. Orientados a coor-
			dinación de equipos de rescate
			en escenarios definidos.
[2,7,8,9]	Sistemas de detección de	(ii)	Se enfocan primordialmente en
	señales sísmicas y alerta		la detección y la difusión de
	a la población.		alerta por medios masivos de
			comunicación.
[17]	SMA de detección de	(ii)	Está enfocado únicamente en la
	riesgo basado en el siste-		detección del riesgo.
	ma inmunológico y algo-		
	ritmos genéticos.		
[22]	SMA para la administra-	(i)	Utiliza dispositivos móviles para
	ción de desastres natura-		administrar información pro-
	les.		porcionada por el usuario acerca
			de desastres naturales, usando
			agentes.
[1,3,15]	SMA para simulación de	(ii)	Utilizan un simulador basado en
	respuesta a emergencias.		agentes, combinado con espa-
			cios inteligentes (instrumentado
			con capacidades de detección y
			comunicación), o representacio-
			nes 3D.
[14]	Robots para búsqueda y	(i)	Basado en robots autónomos
	rescate urbanos.		para exploración de superficies,
			mediante la construcción de
			mapas bidimensionales.

- Aumentar la eficiencia y coordinación de los equipos de rescate durante las intervenciones, dotándolos de información sobre vulnerabilidad, amenazas y las condiciones físicas del terreno donde sucede el evento.
- Mejorar las condiciones de seguridad, evaluación del riesgo y prevención para todos los usuarios involucrados en el sistema y su área de influencia.

3.1. Enfoque general

El sistema propuesto implica cuatro etapas para el logro de sus objetivos. Éstas comprenden los elementos constitutivos de un SSATR y guían el desarrollo del mismo, los cuales representan los componentes clásicos de un sistema de cómputo: Entrada, Procesamiento, Salida y Retroalimentación. A continuación se da una breve descripción de dichas etapas:

- Detección del peligro (Entrada): Involucra la recopilación de datos a través de dos diferentes medios: a) sensores y plataformas de *hardware* especializado, y su procesamiento inicial para obtener información relevante (por ejemplo, la identificación de señales asociadas a sismos o disturbios urbanos), así como b) alertas e información proporcionada por los mismos usuarios (por ejemplo, mediante plataformas sociales) o por sistemas ya existentes (por ejemplo, sistemas gubernamentales).
- Evaluación del riesgo (Procesamiento): Caracterización del peligro, análisis de la vulnerabilidad e identificación del riesgo. Realiza un procesamiento tanto individual como colectivo a partir de la información sensorial y la procedente tanto de usuarios como de otros sistemas para determinar el nivel de riesgo, y en su caso, disparar alertas y generar la información relevante para la toma de decisiones, enfocada a que las personas se pongan a salvo, minimizar el riesgo y facilitar la intervención de los rescatistas en el terreno.
- Alerta oportuna y soporte al usuario (Salida): Interfaz entre el sistema, los usuarios finales y otros sistemas. Proporciona los avisos de alerta y la información de soporte a la toma de decisiones para usuarios finales. Este componente implica el diseño y la creación de las interfaces para permitir la comunicación entre el sistema, los usuarios finales y otros sistemas, de manera que la información sea accesible de manera ubicua, permitiendo la toma de decisiones oportuna.
- Evaluación de riesgos posterior al evento (Retroalimentación): Realiza un proceso de evaluación de la efectividad del sistema y lo retroalimenta, utilizando diferentes fuentes, como datos autogenerados o registros históricos. Con ello, el sistema debería ser capaz de clasificar los desastres y proporcionar información relevante para la evaluación y pronóstico de riesgos (cuando sea posible), generando visualizaciones de datos para la facilitar la percepción humana y permitir la oportuna toma de decisiones.

Estas etapas se muestran en la Fig. 1 junto con las actividades generales realizadas en cada una. De igual manera, se muestran tanto las actividades realizadas en cada etapa, como sus tecnologías asociadas.

El sistema basa su operatividad en un modelo computacional de la interacción combinada entre usuarios, agentes artificiales y otros sistemas, donde la interoperabilidad e integración de sistemas juegan un papel transversal dentro del mismo. Sin embargo, el propósito clave de esta propuesta es el apoyo a los usuarios finales, es por ello que la interfaz cobra una dimensión crucial en su funcionamiento.

A continuación se introducen algunos conceptos básicos relacionados con la propuesta que se detalla en la Sección 4.

3.2. Conceptos preliminares

El término agente define a un sistema computacional, tanto hardware como software situado en un ambiente, en el que se desenvuelve con el fin de conseguir sus objetivos de diseño [19]. Sin embargo, hay ciertas propiedades inherentes

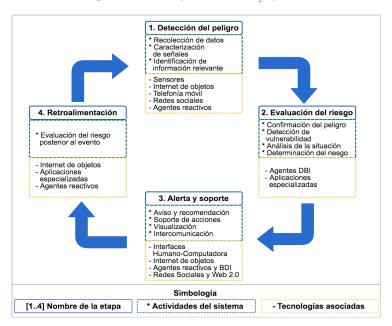


Fig. 1. El sistema comprende cuatro etapas: Detección del peligro, Evaluación del riesgo, Alerta oportuna y Soporte al usuario, y Retroalimentación.

a los agentes, como proactividad (ser persistente en el logro de sus objetivos), deliberación (razonar sobre sus percepciones y estímulos) y capacidad social (intercomunicación entre agentes y con otros sistemas). Esta última característica permite crear ensambles de agentes, conocidos como Sistemas Multi-Agente (SMA), que comparten su ambiente e interactúan para conseguir objetivos comunes [21,24]. Una característica importante de los SMA es su tolerancia a fallos y su control descentralizado, lo que los hace ideales para enfrentar distintas tareas con alto grado de robustez. Los SMA han sido aplicados a distintos dominios, desde el comercio electrónico y aplicaciones industriales, hasta simulaciones de fenómenos sociales y naturales [10,18,21,24].

A su vez, un SMA puede estar compuesto por agentes homogéneos (similares en capacidades y diseño) o heterogéneos, que además de cumplir distintas tareas y poseer diferentes capacidades, son diseñados bajo los principios de distintas arquitecturas. Los agentes reactivos, implementados como conjuntos de reglas si-entonces han sido utilizados extensamente para el desarrollo de SMA [18,24] en distintos dominios incluyendo simulaciones de desastres y como apoyo a tareas tareas de salvamento y recuperación post evento. Sin embargo, pese a, la facilidad de implementación y su efectividad, los agentes reactivos son insuficientes para representar individuos o conglomerados humanos. Por ello, se ha propuesto una arquitectura más ad hoc para representar ciertos aspectos del comportamiento humano, los agentes cognitivos cuya arquitectura es más robusta en cuanto a sus modelos de deliberación [4,18,21,24].

En el sistema propuesto en la Sección 4, concurren agentes cognitivos y reactivos, además de otros elementos de cómputo encargados de realizar tareas inteligentes para el cumplimiento de los objetivos de diseño.

Dentro de los agentes cognitivos una de las arquitecturas más conocidas y reputadas es la arquitectura BDI ⁷ que concibe a los agentes desde una perspectiva intencional, es decir, incorpora una representación de *creencias* (el conocimiento que el agente posee), *deseos* (los objetivos a cumplir) e *intenciones* (lo que se propone hacer en un momento dado) para conformar el comportamiento del agente [4,10,19,24].

4. Diseño del sistema de seguridad y alerta temprana de riesgo basado en agentes

El diseño del Sistema de Seguridad y Alerta Temprana de Riesgo Basado en Agentes (MAS_{SSATR}) sigue la metodología de desarrollo de SMA denominada Prometheus [16] propia para el desarrollo de agentes BDI, cuyo nivel de abstracción permite definir la arquitectura del sistema y su interacción interna (entre agentes y con otros sistemas, como redes sociales, dispositivos móviles, servicios web) y externa (con otros sistemas, como el software especializado de las autoridades de seguridad y protección civil).

4.1. Panorama general del SMA

La visión general del MAS_{SSATR} se muestra en la Fig. 2, conocido en la metodología Prometheus como Analisys Overview Diagram. El diagrama referido incluye una vista abstracta del sistema, muestra los agentes que lo componen así como sus interacciones, y lo más relevante de las percepciones que reciben y las acciones que llevan a cabo.

En breve, el usuario, representado en el diagrama como *Usuario* envía la percepción inicio para arrancar el sistema a través de un agente embebido en un teléfono celular (dispMovilAg) que actúa en representación del usuario en el SMA. Al iniciar su funcionamiento, el agente dispMovilAg busca cumplir sus objetivos dentro de tres posibles escenarios⁸: 1) vigilancia (detección de peligros), 2) evalRiesgo (Evaluación del riesgo), y evalSitActual (Alerta, soporte y retroalimentación). Para realizarlo, el agente dispMovilAg intercambia una serie de mensajes con los agentes embebidos en la infraestructura urbana y rural (infraestrAg), los cuerpos de seguridad (rescatistas y policía) y los medios masivos de información (medMasivos) junto con las redes sociales (colabColect y redesSociales) para verificar la existencia de un riesgo para la seguridad del usuario o descartar posibles falsas alarmas, y finalmente, proporcionar las recomendaciones de seguridad pertinentes. Por otro lado, tanto autoridades como

⁷ Del Inglés Belief-Desire-Intention

⁸ En Prometheus un escenario representa una descripción abstracta de una secuencia particular de pasos dentro del sistema.

Wulfrano Arturo Luna-Ramírez, Anasol Peña-Rios, Emmanuel Ferreyra, et al.

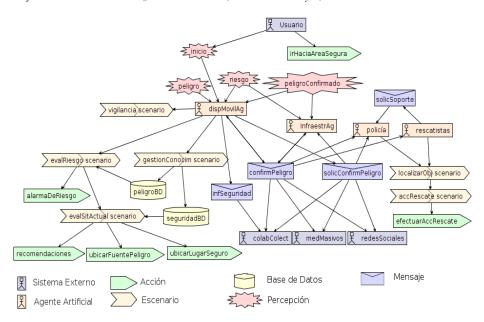


Fig. 2. Analisys Overview Diagram o Diagrama general del SMA, muestra los agentes involucrados así como sus percepciones y acciones más relevantes.

cuerpos protección civil son representados como agentes en el sistema cuya función es primordialmente emitir y recibir información de seguridad, como la verificación de riesgos o la reacción ante alarmas para el despliegue de misiones de socorro y rescate en las áreas afectadas por un desastre.

La misión principal de los agentes que actúan en representación de los ciudadanos (dispMovilAg) es identificar si algún peligro potencial puede caracterizarse como un riesgo y dar soporte al usuario para evitarlo y ponerse a salvo.

El riesgo R depende de dos componentes: el peligro detectado P y el grado de vulnerabilidad del usuario V. De manera simple, esta relación se muestra en la ecuación (1):

$$R = P + V. (1)$$

Para ilustrar la forma en que el SMA implementa el protocolo de detección de riesgo, se muestra el comportamiento que los agentes dispMovilAg deben realizar, denominado $Comportamiento_{SSATR}$:

- 1. Sensar el ambiente
- 2. Analizar las sensaciones para obtener su conjunto de percepciones asociadas
- 3. Identificar un posible peligro P:
 - a) Caracterización (determinar sus rasgos y su clasificación)
 - b) Definir la zona de responsabilidad
 - c) Identificar la fuente de peligro
 - d) Caracterizar sus manifestaciones

- e) Parametrización y medición
- f) Caracterización del área de impacto y el despliegue probable
- g) Representación gráfica
- 4. Determinar el grado de *vulnerabilidad V* analizando su situación actual (que correspondería a la del usuario final que el agente representa):
 - a) Identificar la distribución del peligro en el lugar actual
 - b) Identificar los elementos expuestos del lugar
 - c) Determinar el grado de exposición del usuario
- 5. Determinar si existe un riesgo potencial R [Eq. (1)]
- 6. Clasificar el riesgo R_i , donde $i = \{categorías de riesgos identificados\}$

Finalmente, el sistema incorpora una medición del cumplimiento de su objetivo general, denominada Índice de Riesgo General (IRG), que expresa el grado de riesgo colectivo de todos los agentes dispMovilAg que componen el sistema. De este modo, cualquier agente que sobrepase un umbral $T_{R_{A}g}$ ejecuta el comportamiento $Comportamiento_{SSATR}$ y contribuye a esta medida. Esto se expresa brevemente en la ecuación (2):

$$IRG = \sum (w_i, R_{Ag_i}), \tag{2}$$

donde i = 1...N, $N = \{Conjunto de agentes\}$.

La operación general del MAS_{SSATR} se resume en el Algoritmo 1.

Algoritmo 1: Operación General MAS_{SSATR} .

```
MAS_{SSATR} // Las etapas del sistema están encerradas entre
     paréntesis
1 Calcular IRG
  mientras VERDADERO hacer
     // Sensación y Percepción (ENTRADA)
     Para todos los agentes mobile Device Aq:
2
      Detección del peligro P y determinación de la vulnerabilidad V
3
     // Identificación y gestión del riesgo (PROCESAMIENTO):
      Computar el Riesgo: R = P + V
4
     // (SALIDA Y RETROALIMENTACIÓN)
     si R > T_{R_{A}g} entonces
5
6
         Alarma correspondiente a R
         Valoración post evento
         Recalcular IRG
8
     fin
9
      Interacción con usuarios e intercomunicación con otros sistemas
  fin
```

5. Conclusiones

Los desastres tienen origen en las fuerzas de la naturaleza pero también se presentan como resultado de la intervención humana. Ocurren en distintas magnitudes tanto en regiones rurales como urbanas asolando a las poblaciones también en distinta escala.

Un sistema computacional que apoye en casos de desastre debe cumplir con varias características para enfrentar situaciones de crisis, como el control descentralizado, adaptación y manejo de incertidumbre e información incompleta, para dar una respuesta inmediata a los eventos cuya impredictibilidad los hace aún más difíciles de tratar.

En este artículo se hace una propuesta para desarrollar un Sistema de Seguridad y Alerta Temprana de Riesgos basado en agentes. El objetivo del sistema es proporcionar información oportuna y pertinente en caso de desastre y dar apoyo a la toma de decisiones tanto a las víctimas como a las autoridades y cuerpos de protección civil. Se presenta la arquitectura del sistema y sus elementos constitutivos. Debido a que los agentes cognitivos, en particular los agentes basados en el enfoque BDI, son flexibles y robustos, pueden representar a personas y colectivos de ellas de manera más realista que otras arquitecturas de agentes, asignándoles distintos perfiles y comportamientos esperados, lo que los hace una herramienta prometedora para el desarrollo del sistema. Adicionalmente, la propuesta incorpora la visión del diseño centrado en el ser humano, lo que permite ofrecer facilidad de uso y una comunicación efectiva que redunde en la prevención y el manejo adecuado de una crisis derivada de algún desastre antrópico o natural y las consecuencias derivadas de éste.

El trabajo a futuro incluye la implementación de la propuesta y la evaluación del sistema desarrollado, para lo cual es necesario diseñar una estrategia de pruebas aplicable. Se prevé conectar el SSATR con una interfaz de usuario para dispositivos móviles que permita recuperar información para realizar la retroalimentación inicial. Así mismo, se plantea trabajar en diferentes escenarios de simulación para evaluar el SSATR. Los resultados obtenidos serán reportados en subsecuentes publicaciones.

Referencias

- Adams, N., Field, M., Gelenbe, E., Hand, D., Jennings, N., Leslie, D., Nicholson, D., Ramchurn, S., Rogers, A.: The ALADDIN Project: Intelligent Agents for Disaster Management. In: Proceedings of the first international joint conference on Autonomous agents and multiagent systems part 3 AAMAS 02 p. 1405 (2008)
- Asming, V.E., Baranov, S.V., Vinogradov, Y.A., Voronin, A.I.: Seismic and infrasonic monitoring on the Spitsbergen archipelago. Seismic Instruments 49(3), 209–218 (2013)
- Balasubramanian, V., Massaguer, D., Mehrotra, S., Venkatasubramanian, N.: A simulation framework for emergency response drills. pp. 237–248 (2006)
- 4. Balke, T., Gilbert, N.: How do agents make decisions? a survey. Journal of Artificial Societies and Social Simulation 17(4), 13 (2014)

- Berry, D., Usmani, A., Torero, J.L., Tate, A., McLaughlin, S., Potter, S., Trew, A., Baxter, R., Bull, M., Atkinson, M.: FireGrid: Integrated emergency response and fire safety engineering for the future built environment FireGrid technologies. (2005)
- 6. Bonfiglio, A., Carbonaro, N., Chuzel, C., Curone, D., Dudnik, G., Germagnoli, F., Hatherall, D., Koller, J.M., Lanier, T., Loriga, G., Luprano, J., Magenes, G., Paradiso, R., Tognetti, A., Voirin, G., Waite, R.: Managing catastrophic events by wearable mobile systems. In: Löffler, J., Klann, M. (eds.) International Workshop on Mobile Information Technology for Emergency Response. pp. 22–23. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
- 7. Cuéllar, A., Suárez, G., Ibarrola, G., Uribe, A., Rodríguez, F.H., Islas, R., Rodríguez, G.M., García, A., Frontana, B.: The Mexican Seismic Alert System (SASMEX): Its Alert Signals, Broadcast Results and Performance During the M7.4 Punta Maldonado Earthquake of March 20th, 2012. In: Wenzel, F., Zschau, J. (eds.) Early Warning for Geological Disasters, chap. 4, pp. 71–87. Springer-Verlag Berlin Heidelberg (2014)
- Echenique, M.: Mexico Urgently Needs to Overhaul Earthquake Warnings (2017), https://www.citylab.com/environment/2017/09/the-case-for-overhauling-mexicos-earthquake-alert-system/539448/. Último acceso: 15 Mayo 2018.
- Espinosa-Aranda, J.M., Cuéllar, A., Ibarrola, G., Islas, R., García, A., Rodríguez, F.H., Frontana, B.: The Seismic Alert System of Mexico (SASMEX) and their Alert Signals Broadcast Results. 15th World Conference on Earthquake Engineering (2012)
- 10. Fasli, M.: Agent technology for e-commerce. John Wiley & Sons Chichester (2007)
- Feese, S., Arnrich, B., Rossi, M., Troster, G., Burtscher, M., Meyer, B., Jonas, K.: Towards monitoring firefighting teams with the smartphone. 2013 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2013 (March), 381–384 (2013)
- 12. ISO: ISO 9241-11:2017 Ergonomics of human-system interaction (2017)
- Khalil, K.M., Abdel-Aziz, M., Nazmy, T.T., Salem, A.B.M.: Multi-Agent Crisis Response systems - Design Requirements and Analysis of Current Systems. In: Fourth International Conference on Intelligent Computing and Information Systems, p. 6 (2009)
- 14. Kleiner, A., Steder, B., Dornhege, C., Höfer, D., Prediger, J., Stückler, J., Glogowski, K., Thurner, M., Luber, M., Schnell, M., Kuemmerle, R., Burk, T., Bräuer, T., Nebel, B.: RoboCupRescue Robot League Team. Tech. rep. (2005)
- Marecki, J., Schurr, N., Tambe, M.: Agent-Based Simulations for Disaster Rescue Using the DEFACTO Coordination System. In: Emergent Information Technologies and Enabling Policies for Counter-Terrorism. pp. 281–297 (2005)
- Padgham, L., Winikoff, M.: Developing Intelligent Agent Systems—A Practical Guide. John Wiley & Sons (2004)
- Qing, Y., Huimin, M., Yanling, Y.: Multi-Agent Risk Identifier Model of Emergency Management System Engineering Based on Immunology*. Systems Engineering Procedia 4(91024020), 385–392 (2012)
- 18. Railsback, S.F., Grimm, V.: Agent-Based and Individual-Based Modeling: A Practical Introduction. Princeton University Press (2012)
- Rao, A.S., Georgeff, M.P.: BDI Agents: From Theory to Practice. In: Proceedings of the First International Conference on Multiagent Systems, 95, 312–319 (1995)
- Renda, E., Garay, M.R., Moscardini, O., Torchia, N.P.: Manual para la elaboración de mapas de riesgo. Ministerio de Seguridad de la Nación (2017)

- 21. Russell, S.J., Norvig, P.: Inteligencia Artificial: un enfoque moderno. Colección de Inteligencia Artificial de Prentice Hall, Prentice Hall Hispanoamericana (1996)
- 22. Sharmeen, Z., Martinez-Enriquez, A.M., Aslam, M., Syed, A.Z., Waheed, T.: Multi agent system based interface for natural disaster. In: Active Media Technology. pp. 299–310. Springer International Publishing, Cham (2014)
- 23. United Nations: Global Survey of Early Warning Systems. Tech. rep. (2006), http://www.unisdr.org/ppew/info-resources/ewc3/Global-Survey-of-Early-Warning-Systems.pdf. Último acceso: 15 Mayo 2018.
- 24. Wooldridge, M.: An introduction to multiagent systems. John Wiley & Sons (2009)

Generación de control de dirección para vehículo autónomo por medio de programación genética

Edgar Miguel Aguilar Diaz, Katya Rodríguez Vázquez

Universidad Nacional Autónoma de México, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, México

xeicker@gmail.com, katya.rodriguez@iimas.unam.mx

Resumen. En este artículo se presenta el desarrollo de un controlador para vehículo autónomo con dirección Ackermann [11] evolucionado por medio de Programación Genética. La dirección Ackermann es la utilizada por un coche común de cuatro ruedas. En este caso, la Programación Genética hace uso de una función de evaluación que describe la geometría característica con la que se mueve el vehículo, sin considerar los efectos dinámicos, sino únicamente los cinemáticos. Se resuelve el problema mediante su división en dos partes. Primero, llevar al auto a una recta cualquiera, esta parte se soluciona independientemente y crea el controlador para implementarlo en la simulación de la siguiente parte. Y segundo, se generaliza para una trayectoria representada por un conjunto de rectas, en forma de una recta poligonal. Para hacer esta generalización se utilizan coordenadas relativas, lo que implica transformación de coordenadas. Esto con la finalidad de utilizar la misma referencia que en la primera parte, sin importar la recta en que se encuentre el auto. Los resultados obtenidos presentan comportamientos adecuados que cumplen con la tarea definida como objetivo.

Palabras clave: programación genética, dirección Ackermann, control.

Generation of the Steering Control for an Autonomous Vehicle using Genetic Programming

Abstract. This article presents the development of the steering control for an autonomous vehicle with Ackermann steering [11] evolved using Genetic Programming. Ackermann steering is the one used in most four-wheeled cars. In this case, Genetic Programming uses an evaluation function which describes the specific geometry the vehicle moves with, which means that it doesn't take into consideration dynamic effects, only kinematics. The problem is solved by first dividing it into two different parts. First, to take the car toward any straight line, this part is treated entirely independent from the rest and generates a controller to be implemented in the simulation of the next part. And second, it is generalized to any trajectory made of straight lines, in the form of a polyline. This generalization is achieved using relative coordinates, which implies that a transformation is needed, with the aim of always using the same reference as in the previous part, without the need of knowing the line where the car lies. The

results obtained show an adequate behavior which meets the assigned task defined as its objective.

Keywords: Genetic Programming, Ackermann Steering, Control

1. Introducción

La Programación Genética (PG) propuesta por John Koza [3] es una técnica evolutiva basada en poblaciones, la cual, a diferencia de un Algoritmo Genético (AG) [1,2] no trabaja sobre una estructura de datos de longitud fija, trabaja sobre una estructura de longitud variable y tiene como objetivo evolucionar funciones, modelos o programas, más que encontrar un conjunto de parámetros.

Hay diferentes antecedentes de desarrollo de controladores [3,7,8,10], se presentan diferentes propuestas del uso de la PG para evolucionar estructuras de controladores. En el caso de un vehículo autónomo, el desarrollo de un controlador de dirección es complejo en el sentido de que es un sistema no lineal, y las acciones de control están muy limitadas por el ángulo máximo que permiten las ruedas de dirección. Por esta razón se optó por utilizar programación genética, dado que, aún sin tener una propuesta de controlador, este enfoque de cómputo bioinspirado sería capaz de generar un controlador satisfactorio.

En [8] se describe una forma de diseñar controladores no lineales, en la que por medio de programación genética se crean diferentes modos de control. A su vez, estos se activan y desactivan obedeciendo a otra función generada de la misma forma. En estas características esta investigación es similar, pues de igual manera se consiguieron dos controladores, en los que uno de ellos toma decisiones sobre el otro. Sin embargo, la forma en que se consigue tiene mayor similitud con [3,7], donde la evaluación se hace directamente probando el control sobre el sistema, en lugar de sobre fórmulas que los validen.

La capacidad de generalización de la programación genética se observó al momento de someter al mejor individuo final a una serie de condiciones aleatorias, y observar su capacidad de resolverlas todas satisfactoriamente.

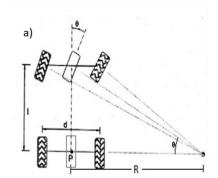
Partiendo de este planteamiento, este artículo está estructurado de la siguiente manera. En la sección 2 se presenta la descripción del simulador empleado. En la sección 3 se presenta una breve introducción de Programación Genética y sus operadores genéticos. En la sección 4 se expresa el planteamiento del problema. La sección 5 describe a detalle las primitivas y los parámetros de la PG empleados para este caso de estudio. En la sección 6 se presentan los resultados obtenidos y en la sección 7 se detallan las conclusiones y trabajos futuros.

2. Descripción del simulador

Para poder generar el controlador es necesario tener un simulador donde probarlo. Por esta razón se programó una función que describiera el movimiento del auto (ecuaciones (3 y 4)). En la Figura 1 se muestra cómo puede reducirse el cálculo del

movimiento de un vehículo de cuatro ruedas a uno de 2 [4,11]. Si el auto conserva una dirección θ por un tiempo t a una velocidad V, recorrerá una distancia D que representa un arco de ángulo φ contenido en el círculo con radio R (ecuación (1)).

$$\varphi = \frac{Vt}{R}; para R = \frac{\tan(\theta)}{L}.$$
 (1)



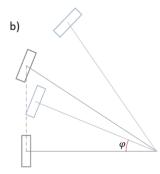


Fig. 1. a) Descripción de como la dirección Ackermann permite modelar un vehículo de cuatro ruedas como uno de dos. b) Ilustración del movimiento de un vehículo de dos ruedas.

Cabe mencionar el siguiente límite:

$$\lim_{R \to 0} \varphi \to \infty. \tag{2}$$

En el límite representado en la ecuación (2) se podría pensar que habría una indeterminación; sin embargo, dado que R al ser función del ángulo de las ruedas, no puede disminuir más allá de lo que permita la torsión máxima de la dirección (considerada en esta investigación como 40°). Con esta consideración pueden ahora plantearse los desplazamientos lineales en coordenadas cartesianas como:

$$\Delta X = R \sin(\varphi), \tag{3}$$

$$\Delta Y = R(1 - \cos(\varphi)). \tag{4}$$

3. Programación genética

La programación genética, como ya se dijo, es una técnica evolutiva basada en poblaciones, la cual utiliza estructuras de forma y tamaño variable. Dichas estructuras pueden ser árboles, grafos en general [8] o incluso líneas de código [4]; para cada una de ellas tiene diferentes operadores de cruza y mutación. Con todo esto, la PG puede generalizar una solución a un problema con diferentes condiciones variables; es aplicable a un sin número de situaciones, dependiendo de la interpretación que se le dé a la evaluación y la forma en que ésta se lleve a cabo. En la Figura 2 se presentan los pasos del algoritmo clásico de Programación Genética.

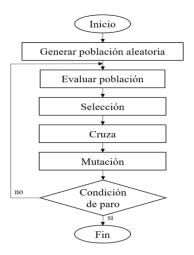


Fig. 2. Algoritmo de programación genética.

El caso concreto de la PG utilizada para este artículo es la estándar, que usa árboles como estructura a evolucionar [6]. Se incluyen las funciones definidas automáticamente [3], o ADFs por sus siglas en inglés, las cuales son árboles completos que representan una hoja en el árbol principal. Las ADFs tienen la particularidad de que dentro del árbol principal son tratadas exactamente igual a terminales, como si fuera un solo nodo y ese nodo tuviera simplemente un dato; por otro lado, fuera de ese árbol pueden cruzarse o mutar en cualquier punto de su estructura.

El operador de cruza seleccionado es uno muy simple, para efectuarlo se eligen aleatoriamente dos individuos, y con probabilidad *Pc* (Probabilidad de Cruza) se ejecuta lo siguiente: en cada uno de ellos se toma un nodo (con probabilidad uniforme) y se intercambian entre sí las ramas (o sub-árboles) asociados a esos nodos.

Además, se introdujo una pequeña adaptación al operador para evitar el crecimiento descontrolado en la profundidad de los árboles. La selección del nodo en el primer individuo sigue haciéndose de forma totalmente aleatoria, pero en el segundo solo puede seleccionarse un nodo cuya profundidad no provoque, después de la cruza, un crecimiento más allá del máximo establecido por individuo.

Para el caso de la mutación se utilizaron 3 operadores diferentes. En el primero se selecciona un nodo aleatorio y, junto con la rama que representa, es remplazado por otro generado aleatoriamente, tal que la profundidad total del individuo no sobrepase el límite. El segundo es muy similar, pero difiere en que en éste el nodo elegido se remplaza con una hoja. En el tercero, las terminales constantes son cambiadas por otras aleatorias.

El primero y segundo ejecutan su operación con una probabilidad Pm (Probabilidad de mutación); en el tercero la probabilidad es para cada nodo, no para la operación total.

4. Planteamiento del problema

Se desea generar el controlador de dirección de un vehículo autónomo. Se plantea que funcione sobre una trayectoria de rectas generada por un procesamiento previo de la información. Esto para reducir la dificultad de manejar coordenadas relativas, mismas utilizadas en [7], por otro lado, se contrasta con ese estudio en que ahí se utilizan curvas y el conjunto de trayectorias con que se entrena es reducido. Visto así, el controlador planteado es un poco complejo, pues el sistema tiene como entrada una trayectoria deseada y a la salida del controlador sólo hay una interacción con el ángulo de las ruedas delanteras. Lo más probable es que si se le da a resolver el problema completo a la PG, no podría resolverlo o requeriría demasiado poder de cómputo. Por lo que se dividió en dos partes.

Lo primero sería lograr un controlador capaz de hacer al vehículo seguir una recta dada, a partir de cualquier posición y orientación relativas a ésta. No obstante, este subproblema puede simplificarse aún más. Si se considera que el auto debe seguir al eje X, puede generalizarse a cualquier recta haciendo la correspondiente transformación de coordenadas. Además, no es necesario que el controlador funcione para cualquier posición y orientación, puesto que la trayectoria se mantendrá cercana al auto todo el tiempo. De esta forma, solo se utilizaron posiciones a menos de 2m de distancia a la recta y orientación de máximo 90° y mínimo -90°. El segundo subproblema sería entonces un controlador que decida en que momento pasar de seguir una recta a seguir la posterior. Las trayectorias se generan aleatoriamente para cada generación, lo cual promueve la generalización del resultado.

Para la simulación en ambos casos se debía discretizar el tiempo, y se hizo con intervalos de una décima de segundo de resolución. Se calcularon las derivadas, ecuación (6), e integrales, ecuación (5), tomando sólo dos puntos. Esto dado que la función que describe el movimiento del auto no es tan compleja, por lo cual con dos puntos se obtiene una aproximación aceptable. De la misma forma, la resolución es la suficiente para hacer una simulación fiel. Dado que la velocidad del auto se consideró en todo momento 1m/s, se tiene un desplazamiento de apenas 10cm para cada evaluación.

$$\int_{a}^{b} f(x)dx \approx \sum_{i=1}^{N} f(x_i) \Delta x \; ; \; N = \frac{b-a}{\Delta x}, \tag{5}$$

$$\frac{d}{dx} f(x) \approx \frac{f(x) - f(x - \Delta x)}{\Delta x}.$$
 (6)

5. Parámetros de PG

Para el primer subproblema, en cada intervalo se tomó una evaluación del controlador y el valor arrojado se consideró la velocidad con que se cambiaba la dirección de las ruedas en radianes/segundo. La simulación se corrió durante 20 minutos, después de los cuales se ejecutaron 100 generaciones. La función de evaluación fue multiobjetivo escalar [6], en la cual se tomaron en cuenta 3 parámetros: la integral de la curva descrita por el auto (dividida entre 100); la diferencia entre las oscilaciones hechas y las mínimas necesarias (divididas entre 15); y el error final del auto.

Para el segundo se hicieron dos versiones distintas. En ambos si el resultado de la evaluación del individuo es positivo, se decide cambiar a seguir la siguiente recta, de otra forma el auto se mantiene en la actual. Los cálculos se realizaron con las coordenadas relativas a la recta en que el auto determinaba estar.

Para la primera versión la simulación tardó poco más de una hora en llevar a cabo 100 generaciones. La evaluación de cada individuo tardó más dado que se evaluaron 10 trayectorias distintas, cada una conformada por 4 rectas. Este número es debido a que se quería considerar la mayor cantidad de casos distintos como: curvas en sentidos opuestos, rectas cortas entre vueltas, y demás.

La función de evaluación de nuevo fue multiobjetivo escalar, tomando en cuenta: la diferencia entre las oscilaciones hechas y las mínimas necesarias (divididas entre 5); la integral de la curva descrita por el auto (dividida entre 10); y, considerando que la trayectoria se dio en un arreglo de rectas, la diferencia entre el índice de la recta a la que llegó y aquella a la cual debía llegar, multiplicado por 100. Este peso tan alto se utilizó para forzar a la simulación a llevar al individuo hasta el final de la trayectoria. Para obtener un mejor controlador, se evaluó sobre 10 posiciones iniciales diferentes y se le asignó como puntaje de adaptación al individuo el peor de los 10.

Para la segunda versión se buscaba ya un controlador capaz de evitar colisiones. Se consideró agregar sensores como variables o formas de sensado como funciones. Sin embargo, la necesidad de procesamiento se incrementó demasiado. Por lo que se consideró al final, utilizar rectas delimitadoras de la trayectoria colocadas a 3 metros de ésta. El objetivo ahora sería no salirse de esas líneas, salirse se consideraría una colisión. De esta forma el generador de trayectorias previo a este controlador debería solo asegurarse de que los obstáculos estuvieran por fuera de dichas líneas.

En la simulación se llevaron a cabo sólo 50 generaciones en alrededor de 3 horas. Esto debido a que el cálculo de colisiones se llevaba muchos recursos de cómputo. La función objetivo se definió únicamente como la minimización de los choques. En este caso se evaluó a cada individuo en 10 trayectorias distintas, y como puntaje del individuo se tomó el promedio. A diferencia de las dos simulaciones anteriores se tuvo cuidado en que, aunque las trayectorias eran distintas en cada generación, eran las mismas para cada individuo, de esta forma se haría una clasificación más justa que llevara a una convergencia más rápida.

Los parámetros compartidos por todas las simulaciones se muestran a continuación, junto con las funciones base (Tabla 1), y las variables tomadas en cuenta en la simulación (Tabla 2):

- Tamaño de población: 500
 Probabilidad de cruza: 0.8
 Probabilidad de mutación: 0.1
 Número de generaciones: 100
- Máxima profundidad:
 - Para primer subproblema: 7
 - Para segundo versión 1: Adf0: 2; Adf1: 2; Árbol principal: 4
 Para segundo versión 2: Afd0: 3; Adf1: 3: Árbol principal: 7
- Elitismo: no
- Forma de selección: Torneo tamaño 2

Tabla 1. Funciones utilizadas.

Función	Resultado
Neg(x)	-X
Suma (x, y)	x+y
Resta (x, y)	x-y
Mul (x, y)	x*y
División (x, y)	$x/y \text{ si } y \neq 0$
	1 si y=0
IfR (a, b, c, d, e)	d si $b \le a \le c$
	e de otra forma
IfGT (a, b, c, d)	$c \text{ si } a \ge b$
	d de otra forma
Hyp (x, y)	$\sqrt{x^2 + y^2}$
Sin(x)	sin(x)
E(x)	$e^{-abs(x)}$
Raiz2(x)	\sqrt{x}

Tabla 2. Variables utilizadas en la simulación.

Variables	Representación
Largo del auto	L
Ancho del auto	A
Velocidad	V
Ángulo de las ruedas	THr
Distancia en dirección de la recta	X
Distancia a la recta	Y
Orientación del auto	ТНа
Máxima torsión de las ruedas	Ms
Velocidad en dirección de la recta	Vx
Velocidad perpendicular a la recta	Vy
Velocidad angular del auto	W
Distancia a la siguiente esquina	Desq
Largo de la siguiente recta	D_Sesq
Angulo de la siguiente esquina	Thesq
Angulo siguiente a Thesq	Th_Sesq

6. Análisis de resultados

El primer controlador fue crucial para continuar con la investigación, pues en él se basaría el resultado final. Se corrieron diferentes simulaciones con diferentes configuraciones; la final es la que se describe en la sección anterior. Además de ajustar esos parámetros, se corrieron 10 simulaciones distintas con ellos; y de éstas se eligió la que lanzó los mejores resultados, basándose directamente en el error y en la apariencia de la trayectoria, mostrada en la Figura 3. El controlador resultante fue el siguiente:

Neg(Resta(Mul(E(THr),Division(Y,L)),Resta(Neg(THr),IfR(X,Ms,THa,W,THa))))

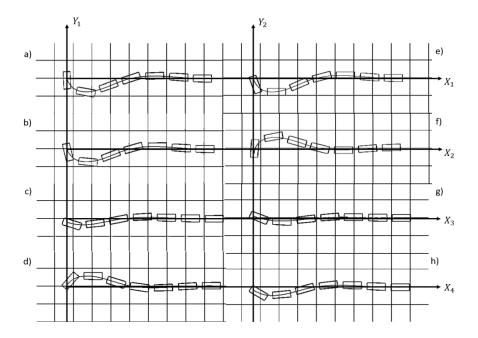


Fig. 3. Se muestran tres diferentes simulaciones con condiciones iniciales distintas (en cuadrícula 5x5m). Puede apreciarse la respuesta del sistema con este controlador y se observan muy pocas oscilaciones al tratar de seguir el eje Xn.

En este controlador se observan algunas variables cuya presencia podría predecirse fácilmente. Por ejemplo, la coordenada en Y comunica al control lo alejado que se encuentra de la recta; la velocidad angular, ayuda a evitar sobrepasos. Por otro lado, resulta interesante ver la presencia de la variable X, puesto que, indirectamente expresa como el error es diferente al inicio del recorrido. Esto dado que, dentro de una función condicional, esta variable modifica el resultado solo en un rango de valores.

También se ven involucradas algunas constantes que al parecer a las leyes de selección les parecieron útiles (el largo del auto y la máxima velocidad de torsión). Estos valores en un ambiente real varían, sin embargo, para una implementación

deberían sustituirse esas variables por constantes. Ya que la simulación aquí descrita, no toma en cuenta el cambio de las mismas ni el efecto que tendrían en el control.

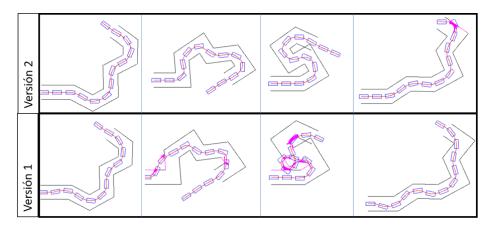


Fig. 4. Comparación de las dos versiones de los controladores. Puede observarse como la versión uno, al guiarse principalmente por la integral de la curva, evoluciona una estrategia que busca encontrar atajos. La versión dos, en contraste, al guiarse por las colisiones, se mantiene más apegada a la trayectoria que debe seguir.

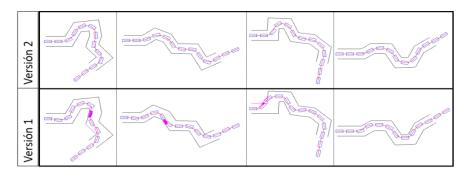


Fig. 5. Comparación de las dos versiones de los controladores. Se observa un comportamiento similar al presentado en la Figura 4.

Para el segundo subproblema, dado que se crearon 2 versiones distintas, se consiguió hacer una comparación de lo que se genera con PG cambiando la función objetivo. Aun cuando ambas versiones resuelven lo mismo, la forma en que se califica su desempeño es distinta, por lo tanto, la definición de "mejor" cambia para cada simulación, lo cual puede apreciarse en las Figuras 4 y 5. De esta forma los distintos controladores obtenidos fueron los siguientes:

- Versión 1:
 - ADF0:
 - o Suma(A,A)

- **–** ADF1:
 - o Hyp(Desq,THa)
- Árbol principal:
 - IfR(Resta(Th_Sesq,ADF1),Neg(D_Sesq),Seno(Resta(ADF0,Desq)), Raiz2(V),IfR(D_Sesq,ADF0,ADF0,THa,-0.659634))
- Versión 2:
 - ADF0:
 - IfR(Division(X,V),IfR(V,-1.422355,V,1.950899,V),IfR(Thesq,X,X,-1.422355,X),-1.422355,-1.422355)
 - ADF1:
 - Hyp(IfR(Ms,Vx,D_Sesq,ADF0,D_Sesq),Ms)
 - Árbol Principal:
 - Resta(Resta(Suma(Desq,Mv),Suma(Desq,ADF0)),Raiz2(Neg(Hyp (Desq , Hyp(Suma(Desq,ADF0),Suma(THr,ADF0))))))

Las distintas estrategias que siguen las versiones 1 y 2 tienen ventajas y desventajas. En la parte derecha de la Figura 4 puede observarse como la preferencia de atajos de la versión 1, le da una ventaja sobre la versión 2. Sin embargo, en el resto de las trayectorias la misma preferencia genera una desventaja, sobre todo en las imágenes centrales; mientras en la imagen de la izquierda igual se genera una trayectoria con más oscilaciones.

7. Conclusiones y trabajos futuros

La programación genética se utilizó con éxito para el desarrollo de un controlador, mismo que fue probado ante diferentes circunstancias y demostró resolver el problema. Esto sin importar la configuración de la trayectoria, y hasta cierto punto la complejidad de la misma, puesto que el controlador fue capaz de mantenerse dentro de las líneas delimitadoras en la gran mayoría de los casos.

Aquí puede verse la aplicabilidad de la programación genética en el diseño de controladores. En este caso se inició con un espacio de búsqueda bastante grande, utilizando todas las variables que podrían o no verse involucradas; e igual se hizo con las funciones. Y aún con un espacio de búsqueda tan basto, el algoritmo pudo entregar un resultado aceptable.

También cabe resaltar que es importante la separación de un problema en subproblemas. De esta forma, aunque se tenga un espacio muy grande de soluciones, los problemas que se intentan resolver son relativamente sencillos. Por ejemplo, al inicio se consideró realizar un controlador que llevara al auto de un punto A a uno B. Sin embargo, eso involucraba demasiadas variables, y se trataba de un problema muy complejo por lo cual no se obtuvieron resultados. De esta forma se notó la necesidad de dividir el problema en dos partes.

Otro punto para tomar en cuenta es la limitación de recursos de cómputo. Para la versión 2 del segundo subproblema se consideró utilizar información de sensores, pero eso hacía que la necesidad de recursos de cómputo creciera demasiado, al punto en que cada generación tomaba más de 20 minutos en evaluarse. Así, si bien no se

consiguió que el controlador utilizara información del ambiente directamente, si es capaz de evitar obstáculos indirectamente.

Esta investigación puede ampliarse en un futuro de dos formas: una es correr el algoritmo de la segunda versión por una cantidad mayor de generaciones. Esto podría perfeccionar la estrategia que se fomenta con esa función objetivo. La segunda forma es agregar los efectos dinámicos, lo cual aseguraría la posibilidad de aplicar el controlador a altas velocidades. Asimismo, resulta interesante combinar este estudio con otras aproximaciones, como la desarrollada en [7], donde se controla, además, el acelerador y se manejan curvas. El resultado sería capaz de enfrentarse a un mayor número de ambientes de una manera más eficiente.

Referencias

- Goldberg, D.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Boston (1989)
- 2. Holland, J.: Adaptation in natural and artificial systems. MIT Press, Cambridge (1992)
- 3. Koza, J.: Genetic programming: On the programming of computers by means of natural selection. MIT Press, Cambridge, Mass. (1992)
- 4. King-Hele, D.: Erasmus Darwin's improved design for steering carriages--and cars. Notes and Records of the Royal Society. 56, 1, pp. 41–62 (2002)
- 5. Brameier, M., Banzhaf, W.: Linear genetic programming. Springer, New York (2007)
- 6. Sekaj, I., Perkacz., J.: Genetic programming based controller design. In: 2007 IEEE Congress on Evolutionary Computation (2007)
- 7. Ebner, M., Tiede, T.: Evolving driving controllers using Genetic Programming. In: 2009 IEEE Symposium on Computational Intelligence and Games (2009)
- 8. Verdier, C., Mazo, Jr., M.: Formal Controller Synthesis via Genetic Programming. IFAC-PapersOnLine. 50, 1, pp. 7205–7210 (2017)
- 9. Mabu, S., Hirasawa, K., Hu, J.: A Graph-Based Evolutionary Algorithm: Genetic Network Programming (GNP) and Its Extension Using Reinforcement Learning. Evolutionary Computation. 15, pp. 369–398 (2007)
- Poli, R., Langdon, W.B., McPhee, N.F., Koza, J.R.: A field guide to genetic programming. Lulu Press, S.l. (2008)
- 11. Geometría Direccional: Estudio de las cotas, https://www.edu.xunta.gal/centros/cafi/aulavirtual2/pluginfile.php/14627/mod_folder/content/0/Geometria_direccional.pdf

Simulador con heurísticas de empaquetamiento para la asignación de memoria y despacho de procesos

Hilda Castillo Zacatelco, Rafael de la Rosa Flores, Claudia Zepeda Cortés, Ana Patricia Cervantes Márquez, Esly Cuautle Aguilar

Benemérita Universidad Autónoma de Puebla, Puebla, México {hildacz,rafa.elo31,czepedac,cervantes.patty,shyzumae}@gmail.com

Resumen. En este trabajo se persiguen varios objetivos: primero, enseñar a los estudiantes de la FCC de la BUAP la importancia de los algoritmos heurísticos de empaquetamiento; segundo, que comprendan las tareas involucradas en la asignación de memoria utilizando listas ligadas, así como el despacho de procesos que se lleva a cabo dentro de un Sistema Operativo; y por último, el desarrollo y la implementación de un simulador. El cual implementa algunos de los algoritmos heurísticos más usados de acuerdo al estado del arte revisado. El simulador consta de tres módulos principales: primero, un asignador de memoria; segundo, un despachador de procesos y por último un módulo de manejo de memoria virtual. En cada uno de los módulo que se presentan en esta propuesta es posible seleccionar entre diferentes técnicas de asignación de memoria y/o despacho de procesos, lo cual permite realizar reconfiguraciones del simulador para establecer comparativas entre los diferentes algoritmos heurísticos.

Palabras clave: heurísticas, FF, NF, BF, WF, asignación de memoria, despacho de procesos, simulador.

Simulator with Packaging Heuristics for Memory Allocation and Process Scheduling

Abstract. In this job, several objectives are pursued: first, to teach the students of the FCC of the BUAP the importance of packaging heuristic algorithms; second, that they understand the tasks involved in memory allocation using linked lists, as well as the dispatch of processes that takes place within an operating system; and by last, the development and implementation of a simulator. Which implements some of the most used heuristic algorithms according to the state of the art reviewed. The simulator

consists of three main modules: first, a memory allocator, second, a process dispatcher and by last a virtual memory management module. In each of the modules presented in this proposal it is possible to select between different memory allocation techniques and / or dispatch of processes, which allows reconfigurations of the simulator to establish comparisons between the different heuristic algorithms.

Keywords: heuristic, FF, NF, BF, WF, memory allocation, process scheduling, simulator.

1. Introducción

El problema de empaquetamiento ha sido ampliamente estudiado desde la década de los 70's y aún sigue siendo de suma importancia debido al crecimiento de la tecnología, prueba de ello es el problema de la asignación de máquinas virtuales en infraestructuras IaaS (Infraestructura como servicio) [1], el cual se puede resolver utilizando técnicas para solucionar el problema de bin packing. Por lo que se puede decir que su uso sigue vigente y abierto a mejoras.

Muchas de estas técnicas caen dentro del área de la Inteligencia Artificial (IA), que como parte de sus estrategias de solución utiliza algoritmos metaheurísticos para resolver problemas. Las metaheurísticas generalmente involucran una o varias heurísticas en sus algoritmos. Las heurísticas son algoritmos que obtienen buenas soluciones pero no demuestran si la solución es óptima, pueden ir desde muy simples a muy complejas [2]. Dentro de las simples y sobre las cuales se basan muchas otras heurísticas y metaheurísticas, se encuentran los algoritmos primer ajuste (FF, First Fit), siguiente ajuste (NF, Next Fit), mejor ajuste (BF, Best Fit) [3] y peor ajuste (WF, Worst Fit).

Estas heurísticas son utilizadas principalmente para resolver problemas de empaquetamiento, como el problema de bin packing (BPP), el de cutting stock (CSP), el de scheduling entre otros. Estos problemas pueden ser mapeados a problemas de la vida real. Éstas heurísticas a pesar de su simplicidad han demostrado dar buenos resultados.

Por otra parte, la enseñanza de los conceptos de sistemas operativos es un tema fundamental para los alumnos de Ciencias de la Computación. Su importancia radica en que un sistema operativo es un administrador de recursos [4] que utiliza diferentes algoritmos para mejorar su desempeño. Muchas de estas técnicas están basadas en algoritmos de la inteligencia artificial.

Este trabajo tiene tres objetivos importantes: primero, que los estudiantes reconozcan la importancia de las heurísticas de empaquetamiento y su aplicación en la solución de problemas reales; segundo, que comprendan las tareas que conlleva la asignación de memoria en Sistemas Operativos y el despacho de procesos, y los algoritmos que se pueden utilizar en cada caso; y por último, construir un simulador para el curso de Sistemas Operativos.

El simulador consiste de tres módulos: un asignador de memoria, un despachador de procesos y un módulo para el manejo de memoria virtual. El asignador de memoria es a su vez un conjunto de módulos que implementa una heurística de empaquetamiento para la asignación de memoria; en el caso del despachador, también son varios módulos que implementan diferentes técnicas de despacho y lo mismo con el módulo de manejo de memoria virtual en donde se implementan algoritmos de reemplazo de páginas.

Cada uno de estos módulos debe permitir utilizar diferentes algoritmos para configurarse e intercambiarse de diferentes formas para evaluar el desempeño de las diferentes combinaciones, de esta manera se pueden realizar comparaciones o incluso mejoras a los algoritmos.

De acuerdo a la experiencia de varios años en la impartición del curso de Sistemas Operativos II de la carrera de Ciencias de la Computación de la BUAP, se ha observado que la mejor forma de aprender es practicando, sin embargo, son tantos los algoritmos que en un curso de 16 semanas es casi imposible revisar a detalle cada uno de ellos. Por lo que la construcción de un simulador como herramienta de apoyo para este curso, se realizará por etapas.

El objetivo final es construir un simulador con módulos bien documentados que pueda utilizarse en los cursos de Sistemas Operativos en donde los alumnos puedan manipular el código para realizar cambios o mejoras.

En este trabajo se presenta el resultado de la primera etapa en la construcción del simulador. Por el momento se cuenta con el módulo de asignación de memoria que cuenta con la implementación de las heurísticas de empaquetamiento FF, BF, NF, WF y se agregó una heurística denominada ajuste rápido. El módulo de despacho de procesos implementa únicamente el algoritmo de Round Robin y esta en construcción el módulo de manejo de memoria virtual en donde se implementarán diferentes algoritmos de reemplazo de páginas.

Los alumnos implementaron estos módulos y compararon el desempeño de las diferente heurísticas, con una entrada de 2000 procesos. Los resultados muestran que con la implementación de estos módulos, los alumnos lograron explicar los conceptos involucrados en estos temas.

El documento está dividido en seis secciones. En la segunda sección se mencionan los trabajos relacionados y la importancia de las heurísticas de empaquetamiento. En la tercera sección se describe la arquitectura del simulador de asignación de memoria y despacho de procesos. En la cuarta sección se describe el funcionamiento de las heurísticas de empaquetamiento FF, BF, NF, WF y ajuste rápido. La quinta sección presenta los resultados en cuanto a la implementación del simulador y el desempeño de los estudiantes después de haber implementado el simulador. En la sexta sección se presentan las conclusiones y el trabajo a futuro.

2. Importancia de las heurísticas de empaquetamiento y trabajos relacionados

En los trabajos del estado del arte en donde se proponen estrategias para solucionar problemas de empaquetamiento y corte como el problema de bin packing, cutting stock o scheduling, se hace uso fundamentalmente de heurísticas.

El tema de asignación de memoria y despacho de procesos, son dos de los temas más interesantes y en muchas ocasiones difíciles de comprender para los estudiantes, por lo que muchos trabajos están enfocados en proponer estrategias o herramientas para la enseñanza de sistemas operativos, principalmente en estos temas. A continuación se mencionan algunos trabajos cuyo objetivo es la enseñanza de los sistemas operativos.

En [18] se presenta una página que describe de forma breve las técnicas de administración de memoria conforme éstas fueron evolucionando, desde particiones fijas, pasando por particiones variables hasta memoria virtual. Es un documento estático con texto plano e imágenes. En [5] Buendía et al. presentan un trabajo en donde hicieron uso de simuladores para la enseñanza de sistemas operativos, en éste concluyen que el uso de herramientas para tal fin requiere de un gran esfuerzo para crearlas por lo que el profesor al utilizar este tipo de herramientas debe cambiar su estrategia de enseñanza y su forma de evaluar.

Desde la década de los 80\u00e9, se ha visto la necesidad de emplear herramientas que le permitan al alumno comprender las tareas de un sistema operativo, tal y como lo muestra el trabajo de Santana y Santana [6] en 1987.

En [7] se presenta un trabajo para la enseñanza de sistemas operativos mediante un simulador que contiene componentes básicos de éste programa maestro, el cual es entregado a los estudiantes para que construyen en él algunos algoritmos de sistemas operativos. En [8] se presenta un trabajo en donde se desarrolló un software que permite simular el manejo de memoria del sistema operativo MAC OS.

Dado que los estudiantes tienen diferentes formas de aprender, es importante incorporar técnicas en el salón de clases que les facilite el aprendizaje de los diferentes temas del curso, tal es el caso de elementos lúdicos. En [9] Hil et. al. presentan un trabajo que utiliza rompecabezas y juegos para la enseñanza de sistemas operativos. Por otro lado, mientras algunos utilizan juegos o simuladores, otros trabajos utilizan sistemas operativos reales para que los estudiantes aprendan procesos, la forma en la cual los estudiantes se comunican con el sistema operativo es a través de una aplicación web, que envía las solicitudes de procesos al sistema operativo, los ejecuta y retorna resultados y retroalimentación [10], esta metodología, de acuerdo a sus autores, es adecuada para cursos de sistemas operativos a distancia.

En cuanto a la importancia del uso de heurísticas en la solución de problemas reales, se tiene que a la fecha se han desarrollados trabajos interesantes, que demuestran la importancia de los algoritmos heurísticos FF, BF, NF, tanto en su uso como en su implementación. A continuación se mencionan algunos de los más relevantes.

En un trabajo reciente en Ahmed [11] se argumenta que la asignación de recursos (procesos, uso de la memoria y migración de procesos) sobre una granja de servidores en la nube hace que se aumente la carga de trabajo y por consiguiente el consumo de energía. Para resolverlo proponen el uso de dos algoritmos; uno heurístico y el otro una metaheurística, FF y colonia de hormigas, respectivamente. El objetivo es eficientar la carga de trabajo en la nube para reducir el consumo de energía y reducir los niveles de CO2. En este mismo sentido, Kumar [12], propone el uso de las heurísticas FF, NF, BF, entre otras, ellos tratan de solucionar el problema de virtualización de hardware en la nube, conocido como IaaS, los servicios que se proporcionan son recursos de hardware tales como: memoria, almacenamiento, tiempo de CPU, ancho de banda, entre otros. Para ello, propone utilizar el conocido problema del BPP, donde cada uno de los servidores IaaS es un contenedor y las máquinas virtuales son objetos que se desea colocar en los contenedores y de esta forma reducir el consumo de energía. Así como también, en [13] se hace referencia a la problemática de encontrar una solución óptima al asignamiento de máquinas virtuales en una plataforma IaaS. Para ello, plantean que se puede resolver utilizando el problema del BPP. En este trabajo se hace hincapié en que a la fecha no existen trabajos serios que permitan evaluar de manera formal cuál algoritmo es mejor en términos de rendimiento. Para ello proponen una metodología que compara siete algoritmos heurísticos basados en best fit para resolverla y utilizan el simulador CloudSim [14], el cual realiza la asignación de máquinas virtuales a máquinas físicas.

Otro trabajo interesante, es el que propone Bein [1], donde enfatizan el uso del almacenamiento de la información en los centros de datos, así como la importancia del problema del BPP para resolverlo. Para ello, realizan un panorama de las heurísticas FF y BF aplicadas al BPP y proponen un algoritmo heurístico, llamado K-binary, éste lo comparan con los algoritmos propuestos por Lee [15] y Seiden [16] y con base en sus resultados experimentales muestran que su propuesta tiene un mejor desempeño.

3. Arquitectura del simulador

El simulador propuesto consta de tres módulos principales: el que lleva a cabo la asignación de memoria, el despachador de procesos y el módulo para el manejo de memoria virtual. La entrada de datos esta dado por un archivo de texto que contiene la información necesaria de cada proceso. En este caso, cada proceso tiene asociado un nombre de proceso, el número de bloques de memoria que necesita y su tiempo de ejecución. En la Figura 1 se muestra la arquitectura del simulador y la forma en la cual intectúan los módulos. Las partes sombreadas son los módulos que aún están en etapa de construcción.

Además de los módulos principales, el simulador consta de módulos auxiliares, tal es el caso del módulo que permite la lectura de los datos de procesos por lotes y el módulo de liberación de memoria. A continuación se describe el funcionamiento general de cada módulo y su interacción con los demás módulos. Arquitectura del simulador de administración de

memoria y despacho de procesos Siguiente Primer Mejor Algoritmos ajuste algoritmos ajuste remplazo despacho de páginas aiuste memoria Aiuste Round rápido Robin Despachador de procesos Asignador Liberador Lectura de de memoria de memoria procesos Estadísticas de calidad (tiempo de Lote de ejecución, procesos aniquilados procesos fragmentación interna, .. Análisis de datos

Fig. 1. Arquitectura del simulador en donde se muestran los diferentes módulos que lo integran y la interacción entre ellos.

Cada cierto tiempo aleatorio, el módulo Lectura de procesos lee un cierto número aleatorio de procesos y los envía al módulo Asignador de memoria. Éste módulo, puede elegir entre diferentes algoritmos para realizar la asignación de memoria, cada algoritmo se encuentra implementado en módulos diferentes que pueden integrarse al sistema. Manualmente se elige el módulo de la heurística o técnica de asignación de memoria que se quiere utilizar y se integra al simulador, esto es, se insertan las instrucciones para invocar a un determinado módulo y se compila el simulador. Lo mismo sucede con el módulo Despachador de procesos y el módulo de Manejo de memoria virtual, selecciona alguno de los algoritmos implementados y lo integran al simulador.

Una vez que todo esta listo, el simulador inicia su ejecución, lee por lotes los procesos y entonces cada vez que un proceso llega al simulador, el Asignador de memoria asigna los bloques o páginas necesarias al proceso. Si el proceso no encuentra espacio en la memoria, entonces es aniquilado y no se ejecuta. Una vez que un proceso esta cargado en memoria se envía su información al módulo Despachador de procesos, quien es el encargado de elegir al siguiente proceso en ser ejecutado. Éste módulo, hace uso de una estructura de datos especial para realizar su trabajo. Esta serie de pasos se llevan a cabo con cada proceso.

Para el caso de las heurísticas de empaquetamiento, utilizan una lista ligada como estructura de datos, para almacenar la información necesaria de los procesos.

El módulo *Liberador de memoria* se encarga de liberar los bloques o las páginas de los procesos que han finalizado su ejecución. En el caso de las heurísticas FF, BF, NF y WF, como se esta utilizando una lista ligada que mapea exactamente el estado de la memoria, entonces se debe revisar si el nodo a liberar tiene huecos adyacentes a la izquierda, a la derecha o a ambos lados, en caso afirmativo se deben fusionar los nodos, en cada contrario, únicamente se cambia el nombre del proceso por un nombre que indique que se trata de un hueco. El módulo *Liberador de memoria* es común para las técnicas FF, NF, BF y WF, en el caso de Ajuste rápido y de *Manejo de memoria virtual* la liberación de la memoria se ajusta a cada caso. El módulo *Despachador de procesos* utiliza una estructura de datos propia para realizar su tarea.

Todo el tiempo el simulador esta generando estadísticas con respecto al tiempo de espera de cada proceso, el tiempo que tardo un proceso en el sistema desde que ingresa hasta que es finalizado, la fragmentación interna que se genera, el número de procesos que son aniquilados. Con estos datos se puede realizar una comparación en cuanto al desempeño de cada heurística y se puede conocer el tipo de variables de las que dependen los resultados. Los estudiantes por el momento realizan el análisis de los datos apoyandose en herramientas estadísticas y dependiendo de los resultados emiten sus conclusiones.

4. Funcionamiento de las heurísticas de empaquetamiento

En esta sección se describirá el funcionamiento de cada una de las heurísticas que fueron implementadas para la asignación de memoria. Además de las heurísticas FF, NF, BF y WF, se implementó una heurística más denominada Ajuste Rápido.

Como la asignación de memoria con listas ligadas ya no es empleada en los sistemas operativos modernos, es frecuente que el estudiante se tope con el problema de no encontrar información que le permita entender a detalle como funcionan estas heurísticas para la asignación de memoria, ya que sus búsqueda se limita a los libros de sistemas operativos. Por ejemplo en [4] se describe de forma escrita el funcionamiento y en [17] no se hace mención a ellas. Dado lo anterior, en esta sección se describe el funcionamiento de forma gráfica tal y como los estudiantes del curso de sistemas operativos las implementaron.

Estas heurísticas utilizan como estructura de datos una lista ligada en donde cada nodo tiene tres campos: el primer campo contiene el nombre del proceso o H en el caso de que sea un hueco, es decir, una cierta cantidad de bloques que están libres; el segundo campo contiene el índice donde comienza el hueco o el proceso, y el tercer campo contiene el número de bloques de que consta el hueco o proceso. En la Figura 2 se muestra un ejemplo de como se pueden mapear los bloques de la memoria a una lista ligada.

En este caso los procesos deben ser asignados sin dividirlos en bloques, aunque la memoria sí está dividida en bloques de tamaño fijo. Al inicio se supone que la memoria está vacía, es decir, no existe ningún proceso asignado, por lo que la lista ligada consiste de un nodo. Suponga que se tiene una memoria de 1Mb

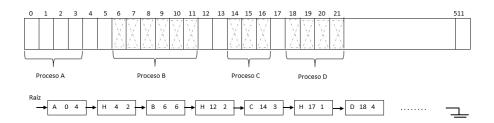


Fig. 2. Representación de procesos en la memoria utilizando listas ligadas.

dividida en bloques de 1Kb entonces al inicio se tendrá una lista ligada de un nodo con identificador H (indica "hueco"), bloque inicial 0 y número de bloques igual a 1024. Cada vez que un proceso entra al sistema, el administrador de memoria asigna el número de bloques necesarios al proceso, los bloques deben ser contiguos.

Cuando los procesos van entrando se les debe asignar memoria, es ahí donde se pueden emplear los algoritmos FF, NF, BF, WF y Ajuste Rápido. Cada una de éstas heurísticas toman los datos de entrada en cualquier orden y en lotes de procesos.

4.1. Primer ajuste (First fit)

El algoritmo inicia la búsqueda al inicio de la lista, si encuentra un hueco donde pueda colocar al proceso lo coloca, en otro caso, avanza hasta encontrar un hueco donde pueda colocar al proceso. Si el hueco es ás grande que el proceso, entonces ese nodo se divide en dos, en el primer nodo se coloca la información del proceso (nombre, número de bloque donde inicia el proceso, número de bloques que ocupa el proceso), en el segundo bloque se coloca la información de los bloques que no fueron asignados (identificador que indica que es un hueco, número de bloque donde inicia el hueco, número de bloques en que consiste el hueco). Si el hueco es exacto para el proceso, entonces únicamente se cambia el identificador de hueco por el nombre del proceso. Si se recorre toda la lista y no se encuentra hueco donde pueda colocarse el proceso, entonces el proceso es aniquilado y no se acepta en el sistema. La Figura 3 muestra los pasos para asignar memoria a tres procesos.

Los problemas que enfrenta este algoritmo es que siempre empieza en la cabeza de la lista ligada que representa la memoria; entonces en el peor de los casos, teniendo una lista ligada de una cantidad de nodos inmensa en la cual ningún hueco, excepto la cola de la lista, pueda albergar al nuevo proceso, recorrerá la lista completamente.

Simulador con heurísticas de empaquetamiento para la asignación de memoria y despacho de procesos

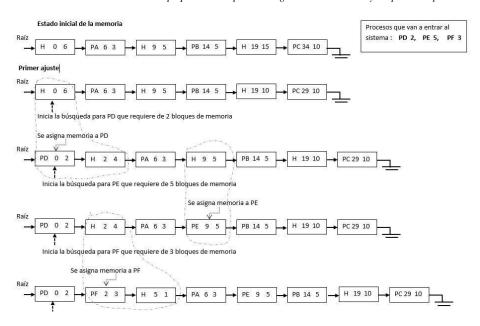


Fig. 3. Funcionamiento de la heurística primer ajuste.

4.2. Siguiente ajuste (Next fit)

Este algoritmo funciona de la misma manera que el primer ajuste cuando llega el primer proceso, esto es, inicia la búsqueda del hueco a partir de la raíz de la lista, cuando lo encuentra inserta el proceso en ese sitio y almacena el apuntador en ese sitio, para que cuando llegue un nuevo proceso, la búsqueda la inicie a partir de ahí, y así sucesivamente. Al igual que el primer ajuste, si no existe hueco para el proceso, éste será aniquilado. Los problemas que enfrenta son similares a los del primer ajuste. La Figura 4 muestra la misma entrada de procesos que para FF, pero ahora utilizando NF.

4.3. Mejor ajuste (Best fit)

Debido al problema que presentan los algoritmos anteriores en cuanto a no discriminar el hueco de memoria para asignar un nuevo proceso se crea este algoritmo, su funcionamiento es que revisará la lista ligada en búsqueda del hueco que mejor le ajuste al nuevo proceso, evitando así utilizar bloques demasiado grandes de memoria que podrían usar los siguientes procesos. Los problemas que presenta este algoritmo son dos, el primero es que debido a que tiene que buscar el mejor hueco para el proceso, esto significa recorrer toda la lista, lo que podría ocasionar un bajo desempeño en el proceso de asignación. El segundo problema es que si el hueco que más se ajusta al nuevo proceso, sigue siendo mayor se dejará un hueco de tamaño pequeño, lo puede ocasionar fragmentación

Hilda Castillo Zacatelco, Rafael de la Rosa Flores, Claudia Zepeda Cortés, et al.

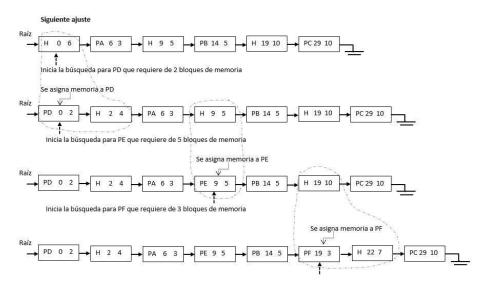


Fig. 4. Funcionamiento de la heurística siguiente ajuste.

externa. En la Figura $\,5$ se muestra el mismo ejemplo que para FF pero ahora con BF.

4.4. Peor ajuste (Worst fit)

Con la finalidad de evitar dejar huecos muy pequeños, se creó el algoritmo del peor ajuste. Al contrario del mejor ajuste, el peor ajuste busca el hueco de memoria más grande con respecto al tamaño del nuevo proceso, la lógica detrás de esto, es que al elegir el hueco más grande, el hueco resultante al asignarle un nuevo proceso tendrá espacio suficiente para albergar a otro y habrá un menor desperdicio de memoria. La Figura 6, otra vez con el ejemplo de FF pero ahora con WF.

4.5. Ajuste rápido

Este algoritmo fue creado para ajustar de manera rápida un nuevo proceso a un bloque que tenga un tamaño similar al del proceso. En este algoritmo los bloques de la memoria son clasificados por tamaño, de esta forma la asignación de la memoria es una tarea rápida. La estructura de datos cambia a un arreglo de listas en los cuales se almacenan los bloques vacíos de memoria por intervalos de tamaño, esto permite que, al llegar un nuevo proceso, dependiendo de su tamaño, se busque en el índice correspondiente al intervalo de su tamaño y se elige el primer bloque que se encuentre en ahí, en caso de que la lista este vacía, se va al siguiente índice, hasta encontrar un hueco al que se le pueda asignar. La

Simulador con heurísticas de empaquetamiento para la asignación de memoria y despacho de procesos

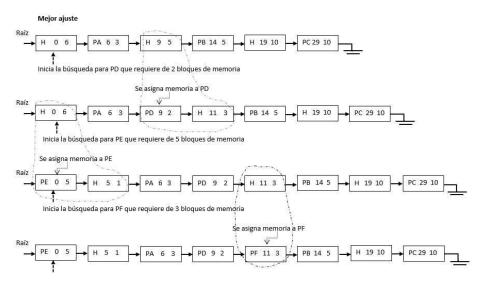


Fig. 5. Funcionamiento de la heurística mejor ajuste.

Figura 7 muestra la forma en la que las estructuras se configuran después de asignar un proceso.

El problema del ajuste rápido, es que no es tan rápido para liberar la memoria, debido a que toda la memoria no se representa en una sola lista ligada, al terminar un proceso no pueden checarse de manera sencilla quiénes son sus vecinos izquierdo y derecho y si estos son huecos. En lugar de esto tiene que buscarse dentro de todo el arreglo de listas donde se encontrarían estos vecinos.

5. Resultados de la construcción del simulador y del desempeño de los estudiantes

Este simulador se ha ido construyendo poco a poco con ayuda de los estudiantes del curso de Sistemas Operativos II del periodo de Primavera 2018 de la FCC-BUAP. Por el momento se han construido: el módulo *Lectura de procesos*, el módulo *Asignador de memoria* del cual se han implementado las heurísticas de empaquetamiento descritas en la sección anterior, el módulo *Despachador de procesos* del cual se ha implementado el algoritmo de Round Robin y el módulo *Liberador de memoria* adaptado a las técnicas de asignación de memoria construidas hasta el momento. Los lenguajes de programación que los estudiantes utilizaron en su mayoría fueron lenguaje C y Java. En la Figura 1 se muestran sombreados aquellos módulos que están en fase de construcción.

Se hicieron pruebas con 2000 procesos simulados. Los parámetros que maneja el simulador son los siguientes: se maneja un quantum de 2 segundos, se simulan 50 Mb con un tamaño de bloque de 1 Kb, los procesos tienen asociados entre 1 y 100 bloques cada uno, y los procesos tienen un tiempo de ejecución de entre

Peor ajuste

Raiz
H 0 6 PA 6 3 H 9 5 PB 14 5 PB 14 5 PD 19 2
Inicia la búsqueda para PD que requiere de 2 bloques de memoria

Se asigna memoria a PD

Raiz
H 0 6 PA 6 3 H 9 5 PB 14 5 PD 19 2 H 21 8 PC 29 10

Se asigna memoria a PE

Raiz
H 0 6 PA 6 3 H 9 5 PB 14 5 PD 19 2 PE 21 5 H 26 3 PC 29 10

Inicia la búsqueda para PF que requiere de 3 bloques de memoria

Hilda Castillo Zacatelco, Rafael de la Rosa Flores, Claudia Zepeda Cortés, et al.

Fig. 6. Funcionamiento de la heurística peor ajuste.

1 y 10 segundos. Los resultados de la evaluación que hicieron los estudiantes a los algoritmos de asignación de memoria, difieren uno de otro. Los estudiantes al realizar su reporte, explican el porqué de sus resultados, y mencionan algunas de las variables que pueden influir como el lenguaje de programación utilizado, la cantidad de procesos que entra cada cierto tiempo, el tamaño de los procesos, el tiempo de ejecución de los procesos, etc. Esto indica que los estudiantes han logrado un conocimiento significativo en éste tema.

La estrategia de solicitar a los alumnos la implementación de heurísticas de empaquetamiento para la asignación de memoria, se ha utilizado en varias ocasiones (al menos en tres ocasiones) en los cursos de Sistemas Operativos II, y se ha encontrado que los alumnos logran comprender mejor los conceptos relacionados con la asignación de memoria y con las heurísticas de empaquetamiento. En todas estas ocasiones se ha planteado a los estudiantes un escenario en el cual se debe asignar memoria a tres procesos, mediante heurísticas de empaquetamiento y el 90 % de los estudiantes han logrado resolver satisfactoriamente el problema. Además se ha aplicado un cuestionario con respecto a conceptos de las tareas que debe llevar a cabo un asignador de memoria y también el 90 % de los estudiantes contestan correctamente.

Además de la construcción de los simuladores, se está construyendo una página [18] en donde se describen brevemente cada una de las heurísticas aplicadas a la asignación de memoria y donde se pueden descargar dos de los simuladores construidos por los alumnos.

6. Conclusiones

Las heurísticas y/o metaheuristicas de la IA pueden utilizarse en los diferentes módulos del sistema operativo, de ahi la importancia de que los estudiantes las

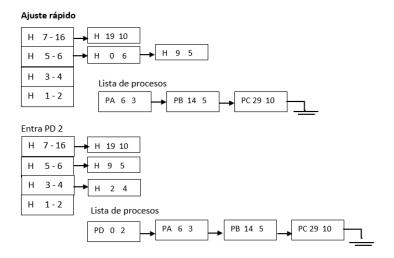


Fig. 7. Funcionamiento de la heurística ajuste rápido.

conozcan.

La implementación del simulador de asignación de memoria y despacho de procesos en donde se utilizan heurísticas de empaquetamiento fueron de vital importancia para que los alumnos del curso de Sistemas Operativos comprendieran y describieran las diferentes tareas que se tienen que realizar en el proceso de la asignación y liberación de memoria utilizando listas ligadas, así como las tareas que debe llevar a cabo un despachador de procesos. Después de que los estudiantes implementaron el simulador fueron conscientes del tipo de problemas que pueden resolverse mediante estas heurísticas y de las variables que podrían alterar los resultados.

En cuanto a la construcción del simulador, se implementaron los módulos de lectura de procesos, de asignación de memoria con heurísticas, de despacho con Round Robin y el liberador de memoria, pero queda como trabajo a futuro implementar el resto de módulos. Se pretende implementar algunos algoritmos para el remplazo de páginas y de despacho de procesos, y de esta forma elegir entre estos algoritmos y reconfigurar el simulador para evaluar su desempeño.

El simulador esta dirigido a la enseñanza aprendizaje de los sistemas operativos en cuanto a los temas de asignación de memoria y despacho de procesos, para que los estudiantes puedan evaluar el desempeño de las diferentes técnicas y que además puedan manipular el código para analizarlo y realizar los cambios pertinentes, e incluso puedan proponer mejoras.

Referencias

 Bein, D., Bein, W., Venigella, S. Cloud storage and online bin packing. In: Intelligent Distributed Computing V. pp. 63–68. Springer, Berlin, Heidelberg. (2011)

- 2. Calandín, G., Andrés C.: Métodos y Algoritmos para resolver problemas de corte unidimensional en entornos realistas. Aplicación a una empresa del Sector sider urgico. Tesis doctoral, Universidad de Valencia, Departamento de Organización de Empresas, España (2008)
- 3. Garey, M.R., Graham, R.L., Ullman, J.D.: An analysis of some packing algorithms. BeliTelephone Laboratories, Inc. and Princeton University (1979)
- 4. Tanenbaum, A., Baer, P., Gagne, G.: Fundamentos de Sistemas Operativos. 7a edición. Mc Graw Hill, España (2005)
- Buendía, F., Cano, Juan-Carlos, Sahuquillo, Julio: Uso de simuladores y herramientas Web para la enseñanza de Sistemas Operativos. In: Actas del SImposio Nacional de Docencia en la Informática, SINDI2005 (AENUI), 121–128 (2018)
- Santana, M., Santana, M.: Uso de minix para la enseñanza de sistemas operativos. Pro Mathematica 1(2), 95–115 (1987)
- Trefftz, H., Cardona, J.F.: Enseñanza de sistemas operativos con un simulador didáctico fácilmente extensible. In: Encuentro Internacional de Educación en Ingeniería ACOFI 2015, pp. 1–8. Asociación Colombiana de Facultades de Ingeniería, Colombia (2015)
- 8. Paginación de memoria en OS X, https://revistas.udistrital.edu.co/ojs/index.php/REDES/article/view/5924/7427. Última visita 19 de abril de 2018
- 9. Hill, J., Ray, C. K., Blair, J. R., Carver Jr, C. A.: Puzzles and games: addressing different learning styles in teaching operating systems concepts. ACM SIGCSE Bulletin 35 (1), pp. 182–186, ACM (2003)
- Buendía, F., Cano, J. C.: Webgene OS: A Generative and Web-Based Learning Architecture to Teach Operating Systems in Undergraduate Courses. IEEE Transactions on Education, 49(4), pp. 464–473 (2006)
- 11. Ahmed, A., Ibrahim, M.: Analysis of Energy Saving Approaches in Cloud Computing using Ant Colony and First Fit Algorithms. Analysis, 12, (2017)
- 12. Kumar, A., Sathasivam, C., Periyasamy, P.: Virtual machine placement in cloud computing. Indian Journal of Science and Technology, 9 (29), (2017)
- Mann, Z. A., Szabó, M.: Which is the best algorithm for virtual machine placement optimization?. Concurrency and Computation: Practice and Experience, 29(10) (2017)
- 14. Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., Buyya, R. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and experience, 41(1), pp. 23–50 (2011)
- Lee, C. C., Lee, D. T.: A simple on-line bin-packing algorithm. Journal of the ACM (JACM), 32(3), pp. 562–572 (1985)
- Seiden, S. S., Van Stee, R., Epstein, L.: New bounds for variable-sized online bin packing. SIAM Journal on Computing, 32(2), pp. 455–469 (2003)
- 17. Silberschatz, A.S.: Sistemas Operativos Modernos. 3ra edición. Pearson / Prentice Hall, México (2009)
- 18. Administración de memoria, http://algoritmosdeubicacion.mex.tl/. Última visita 19 de abril de 2018

Implementación de agentes mediante Máquinas de Estado Finito de Comportamiento (MEFC) aplicados a la simulación de multitudes

José Alberto Hernández Aguilar¹, Isaac Rudomín Goldberg², José Crispín Zavala Díaz¹, Leonel Toledo²

¹ Universidad Autónoma del Estado de Morelos, FCAeI, Cuernavaca, Morelos, México
² Barcelona Super Computer Center, Barcelona, España
jose_hernandez@uaem.mx

Resumen. En esta investigación se discute la simulación de multitudes y la generación de agentes mediante máquinas de estado finito (FSM) en XML, para este propósito se revisan herramientas open source líderes en esta área, particularmente se analizan Menge y FlameGPU, para cada una de ellas se realiza un análisis detallado del funcionamiento de XML aplicado para la generación de scripts de simulación. Luego del estudio comparativo, se propone un nuevo esquema general de XML para la definición de escenas y Máquinas de Estado Finito de Comportamiento denominado XBFSM (eXtendend Behavior Finite State Machine) que busca normalizar la forma en cómo operan cada una de estas plataformas. Nuestros resultados preliminares indican que la simulación de multitudes puede ser representada mediante una jerarquía de archivos XML, constituida por al menos dos esquemas XML, uno que describe el escenario y otro que establece el funcionamiento de la Máquina de Estado Finito de Comportamiento. Para un número pequeño de agentes se recomienda omitir los esquemas XML para la configurar las GPUs, mientras que para un número muy grande de agentes se recomienda configurar los esquemas XML que activan las GPUs.

Palabras clave: simulación de multitudes, XML, máquinas de estado finito, comportamiento, agentes inteligentes.

Implementation of Agents through Behavior Finite State Machines (BFSM) Applied to Crowd Simulation

Abstract. In this research we discuss the simulation of crowds and the generation of agents using finite state machines (FSM) in XML, for this purpose we review leading open source tools in this area, particularly Menge and FlameGPU are analyzed, for each of them a detailed analysis of the functioning of XML applied for the generation of simulation scripts is performed. After the comparative study, a new general scheme of XML is

proposed for the definition of scenes and Finite Behavior State Machines called XBFSM (eXtended Behavior Finite State Machine) that seeks to normalize the way in which each of these platforms operate. Our preliminary results indicate that crowd simulation can be represented by an XML file hierarchy, consisting of at least two XML schemas, one that describes the scenario and another that establishes the operation of the Behavior Finite State Machine. For a small number of agents it is recommended to omit the XML schemas for configuring the GPUs, while for a very large number of agents it is recommended to configure the XML schemas that activate the GPUs.

Keywords: crowd simulation, XML, finite state machines, behavior, intelligent agents.

1. Introducción

La Simulación de multitudes (Crowd Simulation en Inglés), más que una técnica de modelado, es un arte [5], algunas de las aplicaciones de la simulación de multitudes incluyen la detección de comportamiento de la población en lugares públicos como en un aeropuerto, una estación de trenes, o un partido de fútbol [4], su aplicación en la industria del cine se ha enfocado a la creación de grandes batallas (p.e. 300, Game of Thrones), invasiones humanas (por ejemplo: Guerra Mundial Z) y extraterrestres (p.e.: Día de la independencia). De acuerdo a este mismo autor "El objetivo principal en el estudio y simulación de multitudes es representar con precisión grupos de personajes autónomos, llamados agentes virtuales, con reglas y entornos lo más cercanos a aquellos encontrados en la vida real" [4].

En este sentido en [9] se discute un modelo para la simulación del flujo de caminantes y multitudes, este modelo se basa en fuerzas tales como: el deseo de alcanzar un lugar en un determinado tiempo, fuerzas para evitar la colisión entre caminantes, fuerzas para evitar paredes y obstáculos, donde a excepción de la fuerza del deseo, las situaciones se ven afectadas por el resultado del algoritmo del vecino más cercano.

El problema del vecino más cercano se puede resolver mediante la triangulación de Delaunay de manera incremental, en la cual se actualiza al paso del tiempo la posición de cada caminante y la *grid* de la que forma parte, sus resultados muestran que el modelo se desempeña bien para dinámicas como filas, adelantamiento, evitar obstáculos y comportamiento de pánico. En la tabla 1 se muestran los desarrollos más recientes en *Crowd simulation* y agentes.

1.1. Problema en manos

El problema en manos se ha formulado a partir de la siguiente pregunta de investigación: ¿Cómo se puede aplicar una BFSM (*Behavior Finite State Machine* – Máquina de Estados Finitos de Comportamiento) para la creación de agentes inteligentes y la simulación de multitudes en diferentes contextos?

Tabla 1. Desarrollos más recientes en Simulación de multitudes usando Agentes.

Autores	Descripción	Técnica
Luo, Chai, Ma, Zhou y Cai (2018)	Presentan un framework (marco de trabajo) para modelado basado en el comportamiento para modelar el proceso de cómo los humanos seleccionan y ejecutan proactivamente una estrategia de dirección en situaciones de mucha gente y evalúan el correspondiente comportamiento [10].	Modelado de Estrategias proactivas de dirección (proactive crowd) para minimizar colisiones potenciales basadas en comportamiento (gap seeking and following)
Karbovskii, Voloshin, Karsakov, Bezgodov & Gershenson (2018)	Discuten una técnica de simulación multi modelo basada en agentes que incorpora múltiples módulos. Esta integración se basa en un espacio abstracto común donde las entidades de los diferentes modelos interactúan y los agentes pueden ser manejados por diferentes modelos [6].	Simulación multi modelo basada en agentes. Se utiliza la presión de la multitud para estimar la capacidad de diferentes condiciones emergentes que afectan traumáticamente a los caminantes en la multitud.
Liu, Liu, Zhang et al.,(2018)	Presentan una simulación de evacuación de una multitud que está basada en el conocimiento de la navegación y un mecanismo de control de dos etapas. En este trabajo se utiliza un framework de un algoritmo cultural multi población, el mecanismo de control está dividido en dos partes: el espacio de creencias y el espacio de la población. El espacio de la población en grupos o (sub poblaciones), y un líder es seleccionado para cada grupo de acuerdo a un valor de aptitud (fitness value). El espacio de creencias comprende múltiples agentes y una base de conocimiento [8].	Un framework de Algoritmos culturales donde cada agente corresponde a un grupo y obtiene la posición del líder a través de una función de aceptación y más tarde pasa la información a la base de conocimiento.

1.2. Estructura del documento

En la primera sección se discute brevemente la simulación de multitudes, el funcionamiento de XML y su aplicación para la simulación de multitudes, se hace énfasis en la especificación de una BFSM (Behavior Finite State Machine - Máquinas de estado finito de comportamiento), en la segunda sección se describe el funcionamiento de los agentes bajo el contexto de los software de simulación Menge

y FlameGPU, la tercera sección presenta la metodología y nuestro modelo propuesto, la cuarta sección presenta los resultados y discusión. Finalmente, se presentan las conclusiones, trabajos futuros y referencias.

2. XML y las máquinas de estado finito de comportamiento (BFSM)

Las multitudes humanas son un problema de discusión actual, su modelado y simulación ha sido utilizado como herramienta importante para analizar el impacto del comportamiento de masas en un amplio rango de aplicaciones incluyendo la planeación de la seguridad, el diseño de arquitectura, realidad virtual, sistemas de entrenamiento y militares, entre otros [10].

Un documento XML consiste de una serie de caracteres, algunos de los cuales forman caracteres de datos y algunos de los cuales forman marcas. Estas marcas codifican una descripción del plan y la estructura del documento, e incluye comentarios, etiquetas o delimitadores (p.e., etiquetas de inicio, de final y espacios en blanco), declaraciones (p.e., declaraciones del tipo de documento, declaraciones XML, declaraciones de texto) e instrucciones de procesamiento. Los caracteres de los datos comprenden el texto que no está etiquetado.

Una aplicación de software que consume los datos del documento debe ser capaz de examinar el documento, accede a su estructura y contenido (p.e., separar los datos de las etiquetas) y colocar los datos en una forma interna útil. El módulo de software que analiza un documento XML es generalmente llamado procesador XML y trabaja en beneficio de la aplicación [15].

En [10] se señala que el modelado de multitudes basado en agentes ha emergido como el más popular y poderoso debido a su capacidad para modelar comportamientos individuales heterogéneos, así mencionan que la dinámica de la multitud está determinada por el comportamiento de los agentes individuales y su interacción. En este mismo sentido en [14] se generan máquinas de estado finitas a partir de archivos XML que se guardan en imágenes para ser consultadas por los agentes. De acuerdo a [11] un autómata finito (AF) o máquina de Estados Finitos (FSM - Finite State Machine por sus siglas en inglés) es un modelo matemático de un sistema compuesto por: estados, transiciones y acciones.

Un estado almacena información del pasado. Una transición indica un cambio de estado y este es determinado por la condición que es necesaria cumplir para activar la transición. Una acción es una descripción de una actividad que es realizada en un momento dado. Las máquinas de estado finitas, junto con los algoritmos A* para encontrar rutas, y los algoritmos para evitar colisiones son consideradas como técnicas de Inteligencia Artificial aplicadas para la simulación de multitudes.

En [16] se combinaron mapas y scripts XML, los mapas se definen en el script XML usando la etiqueta <map>. Diversos tipos de mapas pueden ser definidos, por ejemplo: mapas estáticos: mapas de ruta, mapas de altura, mapas de textura y mapas de etiquetas pueden ser cargados por el usuario, o actualizado interactivamente. Otros mapas son dinámicos (colisión, visibilidad, nivel de detalle) y pueden ser utilizados

por el sistema si se desea. La combinación de FSM y scripts XML resultaron ser muy productivas.

En [16] se desarrolló la idea de incluir probabilidad, jerarquía y capas al sistema de simulación. Los esquemas probabilísticos incrementaron la diversidad aparente de comportamientos, como comportamiento para los caracteres individuales no está únicamente guiado por scripts rígidos, pero aun así modificados en forma inesperada, pero aun controlada, al seleccionar aleatoriamente ciertas transiciones dentro de la FSM. En [17] se discutió el comportamiento de las multitudes "Crowd Behavior" a partir del uso de XML e imágenes, las multitudes pueden construirse tomando en cuenta los fluidos como base.

Algunos sensores originalmente desarrollados para los fluidos se desarrollaron y reformularon para diferentes usos en la pieza. Sin embargo, el principal propósito de la nueva pieza es permitir al artista construir escenas pobladas por diferentes caracteres. La tecnología que fue usada para implementar las multitudes fue XML.

A continuación, se discuten los agentes estacionarios derivados de las FSM con comportamiento.

2.1. Agentes estacionarios

De acuerdo con [12] cualquier sistema que esta principalmente enfocado al movimiento de agentes debería considerar todos los agentes estacionarios para ser equivalente. El comportamiento de cada agente puede ser diferente pero también tener trayectorias idénticas por medio de su BFSM (Behavior Finite State Machine-Maquina de estados finitos de comportamiento). Dos agentes estacionarios podrían ocupar diferentes estados en la BFSM, representando diferentes actividades o condiciones mentales. La técnica de modelado basada en agentes es una técnica para la simulación computacional de sistemas complejos que interactúan a través de la especificación del comportamiento de un número de individuos autónomos que actúan simultáneamente [7]. Este enfoque en los individuos es más costoso desde el punto de vista computacional, pero provee una manera más natural y flexible para estudiar los ambientes que demuestran el comportamiento emergente.

3. Funcionamiento de XML bajo Menge y FlameGPU

3.1. Menge

Es un ambiente de trabajo con una plataforma extensible y modular que permite simular el movimiento de los agentes en una multitud o aglomeración. La arquitectura de Menge está inspirada en la descomposición del problema de simulación de multitudes en sub problemas o componentes (véase la Figura 1).

3.2. Elementos de la arquitectura

Menge es una arquitectura modular basada en el concepto de elementos. Un tipo de elemento define un aspecto particular de un sub problema. El tipo de elemento define una interface que puede utilizarse para proveer una solución particular. Cada tipo de

elemento puede tener un conjunto diverso de implementaciones. Los elementos implementados son explícitamente instanciados vía una especificación de XML (véase la Figura 1).

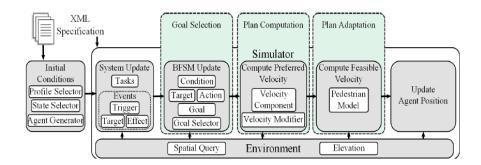


Fig. 1. La Arquitectura de Menge es una arquitectura modular, Fuente [13]. Como se aprecia XML es utilizado para establecer las condiciones iniciales y cómo se actualizará el sistema. La máquina de estado finito de comportamiento determina el comportamiento de cada agente, para el cual se establece su meta, se calcula su plan y se genera su modelo, el cual permite actualizar la posición del agente considerando un ambiente.

Como se aprecia en el siguiente código XML (véanse las Figuras 2 y 3) para la especificación de una escena se establece un experimento, en primer lugar, se define el tipo de consulta espacial que se llevará a cabo, el tiempo que se tomará la actualización, en este caso 0.1, se establecen también los modelos que se utilizarán en la simulación (en este caso los de Helbing y Karamouzas) y sus respectivos parámetros de ajuste.

Así mismo se determina el perfil del agente para el grupo 1, sus parámetros de ajuste: ángulo máximo de velocidad, máximo número de vecinos, obstáculos, tiempo de reacción, fuerza del cuerpo, fricción y fuerza de distancia), sus propiedades estadísticas (media y desviación estándar), se establecen también la masa para el modelo de Helbing, y el espacio personal y nivel de anticipación para el modelo de Karamouzas.

Más adelante, se establecen los parámetros para el grupo de agentes (Grupo 1), en los que se establece la grid (rejilla), la dirección, la densidad, el ancho, la población, la rotación, la distancia y el desplazamiento promedio, así como su desviación estándar. Finalmente, se establece el tipo de obstáculo -en este caso cerrado-, así como los vértices del mismo.

El código XML presentado en las Figuras 2 y 3 muestran la especificación de una Máquina de Estados Finitos de Comportamiento (BFSM) en Menge, en ella se establecen las metas, los posibles estados Caminar y Meta alcanzada, así como sus parámetros de operación. Se definen también las transiciones posibles: de caminar a la meta alcanzada, y de la meta alcanzada a caminar.

Fig. 2. Especificación de una escena para un corredor periódico [13].

Fig. 3. Especificación del comportamiento para un corredor periódico [13].

FlameGPU. El modelado basado en Agentes es una aproximación que de manera intrínseca implica paralelismo, sin embargo, muchos de los ambientes de trabajo existentes fallan para explotarlo y frecuentemente se basan en algoritmos altamente serializados que manipulan agentes discretos móviles. Lo cual tiene serias implicaciones en la escala de los modelos y la velocidad en la que ellos pueden ser simulados. El propósito de FlameGPU es atender estas limitaciones del MBA (modelado basado en agentes) al utilizar una arquitectura de alto desempeño basada

en unidades de procesamiento gráfico GPUs (Graphical Processing Units). El ambiente de trabajo está diseñado con el paralelismo en mente, de manera que permita a los modelos de agentes escalar a tamaños masivos y que permita correr simulaciones dentro de restricciones de tiempo razonables. Adicionalmente a esta visualización, ésta se alcanza como datos que se mantienen directamente en la memoria del GPU donde esta puede ser renderizada directamente [7]. Técnicamente el Framework FlameGPU no es un simulador, es un ambiente de simulación basado en un template que mapea las descripciones formales de los agentes en código de simulación [7].

FlameGPU usa un script de función, para ello este es definido en un número de archivos de Función de Agentes. Los modelos de simulación se especifican utilizando un formato llamado X-Machine Mark-up Language (XMML) el cual tiene la sintaxis en XML que cuenta con esquemas que gobiernan el contenido.

Un modelo típico de una XMML consiste de la definición de un número de agentes en máquinas X, que incluyen información acerca del estado y la memoria, así como las funciones de transición de los agentes, un número de tipos de mensajes (cada uno de los cuales tiene una lista de mensajes globalmente accesibles) y un conjunto de capas de simulación que determinan el orden de ejecución de las funciones del agente lo que constituye una iteración de simulación simple [7], véase la Figura 4.

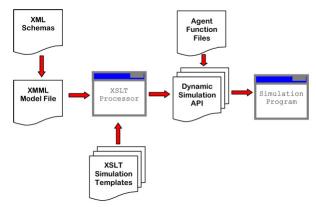


Fig. 4. Proceso de Modelado y Simulación en FlameGPU [7].

Fig. 5. Modelado en FlameGPU fuente propia basado en [7].

Dado una definición de un modelo en una máquina X, la generación de código se logra a través de transformaciones (XLST - Extensible Stylesheet Transformations, o transformaciones de estilo de hojas extensibles), a partir de ella se generan los templates para la simulación dinámica en una Aplicación (API) a partir de la cual se genera el programa de simulación.

En la Figura 4, se muestra la estructura XML para la especificación de un modelo en FlameGPU, primeramente se determina que se utilizaran GPUs, el nombre del modelo, el ambiente de trabajo de GPUs junto con todas sus especificaciones, los contenidos de los Agentes X (X-agents), los mensajes y las capas de aplicación.

En la Tabla 2 se muestra nuestra propuesta diseñada para la integración de estos esquemas, la cual hemos denominado XBFSM (eXtended Behavior Finite State Machine) o bien Máquina de Estado Finito con comportamiento extendido.

Tabla 2. Comparativo entre los esquemas XML de Menge, FlameGPU y nuestra propuesta.

Elemento	Menge	FlameGPU	Propuesta
Máquina de estado finito	BFSM	X-Agent	XBFSM
Definición de escenas en XML	<experiment> <spatialquery></spatialquery> <model> </model></experiment>	<pre><gpu model=""> <gpu:environment> </gpu:environment></gpu></pre>	<experiment> <spatialquery></spatialquery> <name></name> <gpu:environment> </gpu:environment></experiment>
<agentprofile> </agentprofile> <agentgroup> <obstacle> </obstacle> </agentgroup>	<xagents> </xagents> <messages> </messages>	<xagents> <agentprofile> </agentprofile> <agentgroup> <messages> <layers></layers> </messages></agentgroup> </xagents>	<agentprofile> </agentprofile> <agentgroup> </agentgroup>
Creación de agentes	<agentprofile> </agentprofile> <agentgroup> <obstacle> </obstacle> </agentgroup>	<xagents> </xagents> <messages> </messages> <layers> </layers>	<xagents> <agentprofile> </agentprofile> <messages> <layers></layers> </messages></xagents>
Final de la escena			

En esta tabla, la primera columna identifica el elemento analizado y las columnas restantes los esquemas XML para Menge, FlameGPU y XBFSM respectivamente.

Aunque se muestran tres secciones, cabe señalar que estas forman parte de la misma definición de los esquemas XML, la primera sección corresponde al inicio de

la definición de escenas. En Menge se llama experimento, en FlameGPU se denomina modelo, en la XBFSM, se llama experimento, e incluye la consulta espacial para identificar la posición del agente, el nombre y si establecerá un ambiente GPU o no en el experimento, cabe señalar que para pocos agentes este ambiente debe removerse o declararse como nula, para cantidades importantes de agentes esta definición es esencial para la realización del experimento en tiempos razonables.

La sección denominada creación de agentes, en Menge se define el perfil del agente, el comportamiento del grupo y los obstáculos, en FlameGPU se especifica el funcionamiento de los agentes X, los mensajes y capas que se utilizan en la simulación. En la propuesta consolidada se incorpora dentro de los agentes X, el perfil del agente, el funcionamiento del grupo, los mensajes y las capas, cabe señalar que en estas últimas se definen los obstáculos requeridos en Menge.

Finalmente, en el último renglón se define el final del esquema XML, para Menge mediante la etiqueta </Experiment>, para FlameGPU </gpumodel>, y para la XBFSM </Experiment>.

4. Metodología

La metodología propuesta para probar los esquemas XML arriba descritos es la siguiente:

- Si el número de agentes a simular es pequeño se utiliza el esquema XML de Menge, si el número de agentes a simular es grande se recomienda utilizar FlameGPU. Al emplear nuestra propuesta solo se deberá incluir la definición del ambiente GPU (<GpuEnvironment> </GpuEnvironment>).
- 2. Se realiza la simulación en los software de prueba (en Menge y FlameGPU) utilizando los esquemas propuestos.
- 3. Se enlaza la salida de la herramienta y se visualiza en Unity3D.
- 4. Se evalúan los resultados.

Hardware. Estación de trabajo Dell 7450, 6 Gigabytes de Memoria RAM, Disco Duro 1.3 Terabytes. 2 procesadores XEON.; 1 Tarjeta NVIDIA Tesla C2075.

Software. Microsoft Visual Studio Community 2013 [19], FlameGPU Versión 1.4 [7] para CUDA 7.0 [2]; Menge [13] y el Software de simulación Unity3D edición personal [18].

5. Resultados y discusión

5.1. Simulación con Menge y Unity3D

En la Figura 6 se muestra el resultado de la simulación del demo 4square de Menge, el cual fue descompuesto en sus componentes descritos en los esquemas XML, simulación y visualización en tiempo real como se describe en [3].

5.2. Simulación con FlameGPU

La figura 7 muestra la ejecución de Decenas de miles de agentes (caminantes) en un edificio, el visualizador permite analizar la dinámica de la simulación, la ejecución de esta simulación requiere de una tarjeta NVIDIA GPU con capacidades de procesamiento 2.0 o superior para su adecuado funcionamiento.

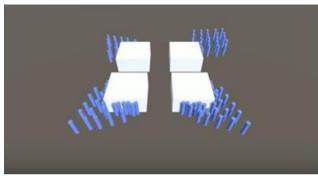


Fig. 6. Simulación del demo 4Square de Menge en Unity3D, basado en [3].

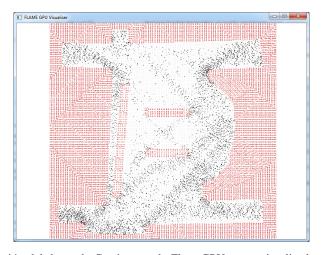


Fig. 7. Simulación del demo de Caminantes de FlameGPU en su visualizador (fuente propia obtenida de la operación de FlameGPU).

6. Discusión

La simulación de multitudes es implementada mediante agentes que emplean máquinas de estados finitas, estas utilizan XML para crear escenas, y definir el comportamiento de los agentes. Para el caso de Menge se utiliza una máquina de estados finitos con comportamiento (BFSM), para FLAMEGPU se utiliza un modelo para agentes denominado X-Agent. Al utilizar Unity3D la calidad y el efecto visual de la simulación realizada en 2D, tanto en Menge como en FLAMEGPU, es mejor y

abre la posibilidad a la realización de simulaciones más realistas. La fortaleza principal de la BFSM es la posibilidad de incorporar nuevos módulos que mejoren la simulación, mientras que la fortaleza más grande del X-Agent es la definición de etiquetas para explotar el uso de las GPU. El nuevo esquema XML propuesto integra estas dos fortalezas en lo que demos denominado eXtended Behavior Finite State Machine (XBFSM).

Las máquinas de estado finito son una herramienta muy poderosa para representar y modelar comportamientos para diversos tipos de agentes dentro de las simulaciones, ya que permiten ver de forma clara cuál es la condición actual del agente y que eventos o condiciones disparan las transiciones entre estados, sin embargo, esta aproximación dista mucho de ser óptima en el manejo de multitudes por varios factores. Las multitudes están compuestas por un conjunto de agentes individuales que deben comportarse de forma distinta en cualquiera de las situaciones que la simulación presente. Utilizando un esquema tradicional de máquinas de estado finito estamos limitados a un conjunto de estados y condiciones que se activan con parámetros definidos, por lo tanto utilizar este esquema con un grupo grande de agentes implica grandes esfuerzos para ajustar el sistema y de esta manera conseguir comportamientos aproximados a la realidad.

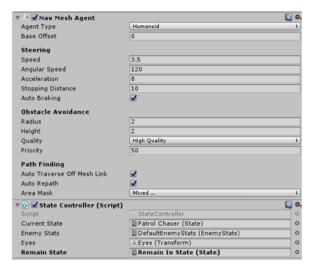


Fig. 8. Parámetros que pueden ser ajustados en la simulación desde el editor y una construcción simple de las máquinas de estado finito.

Para la simulación en Unity proponemos una modificación al manejo de las máquinas de estado finito, de forma que no es necesario hacer cambios manualmente o modificar las máquinas de cada agente para que se ajuste al perfil individual que se pretende simular. Nuestra simulación cuenta con una librería común de conocimiento en la cual los agentes pueden consultar los parámetros generales de la simulación, por ejemplo: velocidad, rango de visión y velocidad de giro por mencionar algunos, y dependiendo de su perfil se ajustan individualmente de forma automática; de esta

manera personajes que pertenezcan por ejemplo al grupo de "niños", tendrán distintas capacidades a los que pertenezcan al grupo de "adultos". Sin embargo ajustar parámetros no resuelve en la totalidad el problema de la individualidad y comportamientos distintos en los agentes, es por eso que también podemos ajustar las condiciones que disparan la transición de los eventos o incluso que en la misma máquina de estados los agentes reaccionen de forma distinta, por ejemplo en una situación de emergencia dos agentes en estado de pánico pueden tener reacciones totalmente opuestas, mientras que uno busca escapar el otro puede decidir intentar localizar a los miembros de un grupo determinado o seguir a otro agente. La figura 8 muestra cómo podemos ajustar y seleccionar esos parámetros desde el editor de Unity, o hacerlos de manera automática utilizando un script.

7. Conclusiones y trabajo futuro

Para el software analizado se identificó que para la realización de simulaciones se requieren de al menos un esquema XML que defina las escenas, y de otro esquema para determinar el comportamiento del agente y el de su grupo. Si se desean simular algunas centenas o algunos miles de usuarios la opción más adecuada es Menge, por otro lado, si lo que se desea es modelar decenas de miles, cientos de miles o más agentes la opción más adecuada es FlameGPU.

Contribución. En esta investigación se presenta una nueva propuesta para la generación de máquinas de estados finitos, denominada XBFSM (eXtended Behavior Finite State Machine) que permite integrar los esquemas XML de Menge y FlameGPU, al incluir dentro de su definición lo mejor de cada uno de ellos.

Limitaciones: Se requiere de un middleware que permita transformar el esquema XML de nuestra propuesta al esquema XML nativo de cada aplicación (Menge o FlameGPU). Este trabajo de ninguna forma está completo, es nuestro deseo realizar simulaciones diversas, entre las que se incluyen la simulación de multitudes en situaciones de emergencia y desastres, como sismos e incendios y probarlas en instituciones educativas en México utilizando Menge y FlameGPU, probar con distinto número de usuarios (densidad de población) y llevar a cabo estudios comparativos de desempeño. Es de particular importancia la preparación de reservas en caso de desastre, para ello se utilizarán como base las investigaciones de [1] al respecto. Así mismo se buscará mejorar el algoritmo de desplazamiento que utiliza la máquina de estados finitos mediante el uso de algoritmos bioinspirados.

Referencias

- 1. Campbell, A.M., Jones, P.C.: Prepositioning supplies in preparation for disasters. European Journal of Operational Research 209(2), pp.156–165 (2011)
- CUDA Homepage, CUDA Toolkit 7.0, https://developer.nvidia.com/cuda-toolkit-70, last accessed 2018/04/04.
- 3. Curtis, S.: Menge en Unity, https://www.youtube.com/watch?v=RpkL6HqqMvU (2018)

- 4. De Gyves, O., Toledo, L., Rudomín, I.: Comportamientos en simulación de multitudes: revisión del estado del arte. Research in Computer Science, 62, pp. 319–334 (2013)
- Helbing, D., Buzna, L., Johansson, A., Werner, T.: Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. Transportation Science, 39(1), pp. 1–24 (2005)
- Karbovskii, V., Voloshin, D., Karsakov, A., Bezgodov, A., Gershenson, C.: Multimodel agent-based simulation environment for mass-gatherings and pedestrian dynamics. Future Generation Computer Systems, 79, pp. 155–165 (2018)
- FlameGPU, Flexible Large Scale Environment Agent Mobile Environment for the GPU. http://www.flamegpu.com/downloads/links (2018)
- 8. Liu, H., Liu, B., Zhang, H., Li, L., Qin, X., Zhang, G.: Crowd evacuation simulation approach based on navigation knowledge and two-layer control mechanism. Information Sciences, 436, pp. 247–267 (2018)
- 9. Löhner, R.: On the modeling of pedestrian motion. Applied Mathematical Modelling, 34(2), pp. 366–382 (2010)
- 10. Luo, L., Chai, C., Ma, J., Zhou, S., Cai, W.: ProactiveCrowd: Modelling Proactive Steering Behaviours for Agent-Based Crowd Simulation. In: Computer Graphics Forum. 37(1), pp. 375–388. The Eurographics Association and John Wiley & Sons Ltd (2018)
- 11. Martínez-Romero, A.: Simultan: un Sistema de I.A. per a Simulació de Multituds. Tesis Grau en Enginyeria en Informàtica Especialitzat en Computació. Universitat Politècnica de Catalunya. Facultat d'Informàtica de Barcelona (2016)
- 12. Curtis, S., Best, A., Manocha, D.: Menge: A modular framework for simulating crowd movement. Collective Dynamics, 1, pp. 1–40 (2016)
- 13. Menge: What is Menge? http://gamma.cs.unc.edu/Menge/ (2018)
- Millan, E., Hernández, B., Rudomín, I.: Large crowds of autonomous animated characters using fragment shaders and level of detail. In: ShaderX5: Advanced Rendering Techniques, chap. Beyond Pix, pp. 501–510 (2006).
- 15. Patent US7596745B2: Programmable hardware finite state machine for facilitating tokenization of an XML document (2018)
- 16. Rudomín, I., Millán, E., Hernández, B.: Fragment shaders for agent animation using finite state machines. Simulation Modelling Practice and Theory, 13(8), pp. 741–751 (2005)
- 17. Rudomín, I., Millán, E., Hernández, B., Díaz, M., Rivera, D.: Art applications for crowds. The Knowledge Engineering Review, 23(4), pp. 399–412 (2008)
- 18. Unity3D: Software de simulación Unity3D edición personal (2018)
- 19. Visual Studio 2013: Visual Studio Community 2013 Release Notes, https://docs.microsoft.com/en-us/visualstudio/releasenotes/vs2013-community-vs (2018)

Comparación de la convergencia de cuatro algoritmos evolutivos al explorar conjuntos de soluciones discretas y continuas

Ibis Azael Arenas-Marín, Anabel Martinez-Vargas, M.A. Cosío-León

Universidad Politécnica de Pachuca, Zempoala, Hgo., México iazaelarenas@micorreo.upp.edu.mx, {anabel.martinez, ma.cosio.leon}@upp.edu.mx

Resumen. La computación evolutiva es un área de investigación dentro de las ciencias computacionales que se basa en la metáfora de la evolución natural. Sus algoritmos llamados evolutivos son herramientas que brindan soluciones a aquellos problemas donde los algoritmos exactos resultan insuficientes. En este trabajo se presenta un análisis de convergencia de las soluciones realizado a cuatro algoritmos evolutivos. Optimización Binaria por Cúmulo de Partículas (BPSO), Optimización Binaria por Cúmulo de Partículas con Mutación (BPSOm), Optimización Binaria por Cúmulo de Partículas Modificado (MBPSO) y Algoritmo Genético Binario (BGA). Se analiza la forma como las soluciones decodificadas por la función objetivo convergen hacia un espacio de soluciones adecuado. Las pruebas realizadas muestran que BPSOm con 10 partículas puede aportar mejores soluciones en comparación con BGA que considera 50 individuos. Lo anterior permite plantear la posibilidad de utilizar BPSOm como una herramienta aplicable al problema de optimización de ubicaciones de vehículos aéreos no tripulados para proveer comunicaciones en situaciones de desastre en la búsqueda de mejores resultados.

Palabras clave: BPSO, algoritmo genético, convergencia, representación binaria, codificación.

A Convergence Comparison Analysis on Four Evolutionary Algorithms Exploring Discrete and Continuous Solution Sets

Abstract. Evolutionary computing is a research area within computer science that bases on the metaphor of natural evolution. Its algorithms known as evolutionary algorithms are tools to resolve problems where exact algorithms offer solutions too expensive in terms of computational time. This work proposes a convergence analysis over four evolutionary algorithms, Binary Particle Swarm Optimization (BPSO), Binary Particle Swarm Optimization with Mutation (BPSOm), Binary Particle Swarm Optimization Modified (MBPSO) and Binary Genetic Algorithm (BGA). The analysis covers the way they move to identify a subset of solutions of interest from the solution space. The results of simulations

show that BPSOm having 10 particles finds better solutions than BGA having 50 individuals. This allows us to consider the possibility of using BPSOm as a tool applicable to the problem of optimizing the locations of unmanned aerial vehicles to provide communications in disaster situations, to search for better results.

Keywords: BPSO, genetic algorithm, convergence, binary representation, coding.

1. Introducción

La selección de la representación de la solución para algoritmos evolutivos no es una tarea trivial. Una correcta elección de la representación de la solución coadyuvará al buen desempeño del algoritmo, mientras que una elección "erróneaçomprometería la eficiencia del algoritmo, ya que generará soluciones inadecuadas o explorará áreas indeseables del espacio de soluciones [1,9].

Durante décadas se ha estudiado y desarrollado distintos tipos de representaciones que pueden emplearse en las diferentes clases de algoritmos evolutivos. Dentro de esas representaciones, se encuentran las de tipo binario y las de punto flotante. En general ambas representaciones poseen la capacidad de proporcionar soluciones adecuadas, sin embargo, la propia naturaleza de cada algoritmo provoca que estas representaciones tiendan a converger hacia óptimos locales o bien converjan hacia áreas de interés muy lentamente [1, 4].

Distinta literatura concuerda que las representaciones binarias de datos flotantes para algoritmos genéticos aportan soluciones con una mayor precisión, permitiendo aproximarlas a un espacio de soluciones deseable. Por ende la representación binaria por su simplicidad y practicidad se ha convertido en la más extendida en muchos algoritmos genéticos independientemente del problema que se está intentando resolver [1]

Otro algoritmo que hace uso de estas representaciones binarias es Binary Particle Swarm Optimization (por su sigla en Inglés, BPSO) [7]. Este, pertenece al grupo de los algoritmos evolutivos, su origen es el algoritmo Particle Swarm Optimization (por su sigla en inglés, PSO) [6], especializado en explorar espaciones de soluciones de tipo flotante, mientras que BPSO hace lo suyo en espacios de soluciones discretas. BPSO demostró que la representación binaria en algoritmos cuya naturaleza es la decodificación flotante, también puede aportar soluciones adecuadas [7]. Variantes del BPSO son el BPSO con una mutación (BPSOm) [8] y el Modified Binary Particle Swarm Optimization (MBPSO) [8], las cuales buscan evitar una problemática del BPSO que se identifica como saturación de la función sigmoide [8].

Cabe hacer mención que no toda la responsabilidad del desempeño de un algoritmo recae directamente en el tipo de representación que se utiliza, si bien es importante, también juegan un papel interesante en la búsqueda de las soluciones idóneas el proceso de selección de dichas soluciones y las distintas operaciones que se realizan con estas antes de ser evaluadas en una función objetivo.

En este trabajo de investigación analizamos la convergencia en el uso de la representación binaria utilizada en [9] para números de punto flotante y números enteros en: Algoritmo Genético Binario (BGA), BPSO, BPSOm y MBPSO. Para describir el proceso investigativo y los resultados obtenidos, este articulo está organizado de la siguiente manera: la Sección 2 detalla las características de cada uno de los algoritmos evolutivos analizados. Las secciones 3 y 4 muestran los parámetros de simulación así como las observaciones realizadas durante las ejecuciones de cada uno de los algoritmos. Finalmente la Sección 5 concluye el trabajo y menciona el trabajo a futuro.

2. Algoritmos evolutivos

Un algoritmo evolutivo cuenta con tres procesos principales, inicialización, donde intervienen los distintos tipos de representación de las soluciones; evaluación, en donde las soluciones son decodificadas y puestas a prueba y finalmente la supervivencia, donde una solución puede permanecer sin cambio o bien aportar a la siguiente generación [5]. Tanto en Algoritmos Genéticos como en Cúmulo de Partículas estos comportamientos generales son observables, aunque no del mismo modo en cada uno de ellos, bastaría citar el cruzamiento (descrito en 2.1) y la memoria de BPSO como elementos característicos de cada algoritmo lo cual los vuelve diferentes. Vistos como herramientas que buscan explorar espacios de soluciones en busca de soluciones óptimas, o bien ubicando áreas de soluciones deseables, los algoritmos evolutivos deben apoyarse de ciertas prácticas que mejoran su eficiencia.

Desde su aparición una de las principales áreas de estudio en algoritmos evolutivos ha sido la adecuada selección y uso de representaciones para las soluciones. Diversos estudios refieren que el uso de representaciones más cercanas a la naturaleza del problema son las que mejores resultados aportan [9]. Esto no representa una regla, ya que por otro lado varios autores documentan que las características propias del algoritmo son también limitantes. En ocasiones estas resultan en desventajas durante la búsqueda de un área de soluciones idónea [2,4,9], lo cual abre una puerta al análisis de las distintas representaciones en los diferentes tipos de algoritmos evolutivos.

Durante décadas las representaciones de tipo binaria y las de tipo flotante se han extendido como las más utilizadas dentro del área de algoritmos evolutivos. En este documento se evalúa la representación binaria como una herramienta que permita una convergencia aceptable a un espacio de soluciones idóneo en cada uno de los algoritmos analizados.

2.1. Algoritmo genético binario

Un BGA, es un caso particular de los algoritmos evolutivos que utiliza técnicas inspiradas en la naturaleza para simular la evolución natural [1,3,4]. Desde su aparición en 1975 ha sido el algoritmo evolutivo de más extenso uso. Básicamente se trata de un grupo de individuos cuya aptitud determina su reproducción y

supervivencia [1,4]. Aunque al igual que en la naturaleza no siempre sobreviven los más aptos, en el BGA las nuevas generaciones de individuos podrían contener individuos con aptitudes no muy buenas. Sin embargo, esta es una característica de estos algoritmos, misma que le permite explorar de forma adecuada el espacio de soluciones [4,9]. Una descripción detallada de las características de los algoritmos genéticos puede encontrarse en [1,3,4,9]

En los algoritmos genéticos la representación de las soluciones es la abstracción de los cromosomas de un individuo. Esta representación determina alguna característica del problema a resolver e influye directamente en los tipos de operadores genéticos que se usarán. Para el BGA se trata de una representación binaria, la cual consiste en generar cadenas de longitud n, de forma binaria (1 ó 0). Cada una de las cadenas representará un cromosoma, mientras que cada uno de los elementos (1 ó 0) representará un gen de dicho cromosoma [1–3] y la decodificación de este cromosoma será el fenotipo.

La forma más extendida de inicialización de los algoritmos evolutivos es la generación aleatoria de cadenas binarias, donde cada solución tiene la misma probabilidad de ser ubicada dentro del espacio de soluciones.

Generalmente para detener un algoritmo genético se utiliza un criterio de parada basado en alcanzar un determinado número de generaciones. Criterios de parada diferentes son expuestos en [1,4,9]. Para la evaluación de los algoritmos genéticos la función objetivo no afecta directamente en su comportamiento interno, es decir, el comportamiento del algoritmo es independiente de la función de evaluación propuesta [4]

La selección dentro de los algoritmos genéticos está determinada por la aptitud de cada individuo. Si se trata de un problema de maximización, con una aptitud alta, el individuo tendrá mayores probabilidades de sobrevivir a la siguiente generación y heredar parte de sus características a los nuevos individuos. Desde la aparición de los Algoritmos Genéticos (GA), los procesos de selección de individuos y/o padres han sido estudiados y modificados [3,9]. La forma más utilizada y que es usada en este trabajo, es el método de ruleta detallado en [1,3]. Para otros métodos de selección puede consultarse [1,3].

En algoritmos genéticos con representación binaria, el cruce consiste en dada una probabilidad de cruce, se generará un número aleatorio entre rango de (0,1). Si este número es menor a la probabilidad de cruce entonces se realizará el cruce entre las parejas de los individuos seleccionados. Para determinar cuál será la información genética que intercambiaran los padres se hace uso de cortes mediante la generación de números aleatorios entre [1, n-1] donde n es la longitud del cromosoma. En este trabajo se utilizará el cruce por un solo punto. Por ello se utiliza la expresión (n-1), ya que un corte en el punto n daría el mismo cruce que un corte en el punto n, es decir intercambiarían la cadena completa de genes.

En la naturaleza la mutación es una peculiaridad de cada individuo que sucede no muy cotidianamente. En los GA una mutación está determinada por una probabilidad de mutación en cada uno de los genes. Al igual que en los procesos anteriores se presenta mediante la generación de números aleatorios

entre (0,1) donde, si el número generado de manera aleatoria es menor a la probabilidad de mutación se presentará la mutación en el gen en turno, de lo contrario el gen permanecerá sin alteraciones. La mutación permite a las soluciones candidatas (individuos) cambiar su aptitud, sin embargo, no siempre esta mutación aumentará la aptitud de los individuos. Por ejemplo, una cadena de 8 bits, donde el bit más significativo sea el que se verá afectado por la mutación, donde la función de evaluación está determinada por $f(x) = x^2$ y se trate de un problema de maximización, el genotipo [1 0 0 0 0 0 0 0 1] representaría el fenotipo 129 y produciría el fitness de 16641, sin embargo al sufrir una mutación en el bit más significativo el genotipo resultaría [0 0 0 0 0 0 0 1] el cual representa el fenotipo 1 que evaluado en la función objetivo daría como resultado 1, lo cual lo convierte en una solución indeseable por su pobre desempeño.

2.2. Algoritmo BPSO y variantes

BPSO vio su primera aparición en el mundo de los algoritmos evolutivos en 1997 como una opción de PSO [7] con representación binaria. Es un algoritmo cuya naturaleza es buscar soluciones mediante el análisis de cadenas binarias, que con auxilio de parámetros de constricción y de interacción social proveen de soluciones a los problemas de programación no lineal [7, 8]. En BPSO, un conjunto de partículas es evaluado en una función objetivo una a una y el desempeño de cada una de estas partículas es almacenado en una memoria. A diferencia de GA, BPSO cuenta con la cualidad de almacenar por cada iteración la mejor partícula (solución candidata). En resumen, este algoritmo cuenta con dos funciones básicas: la actualización de la velocidad y la actualización de la posición. Sin embargo presenta cierta tendencia a la saturación de las función sigmoide [8].

En BPSO la forma de inicializar las partículas es similar a la utilizada en BGA, se inicializan la matriz de posición y velocidad de forma aleatoria y la matriz de memoria en su estado inicial será una copia de la matriz de posición [1,7]. En BPSO el criterio de parada más utilizado es alcanzar un determinado número de iteraciones para después obtener la mejor partícula almacenada en la memoria que será la mejor solución encontrada por el algoritmo para el problema propuesto.

Las operaciones de actualización de velocidad (Ecuación (1)) y posición (Ecuación (2) y Ecuación (3)) en BPSO están determinadas por:

$$v_{(i,j)}(t+1) = wv_{(i,j)}(t) + c_1R_1(p_{best,i,j} - x_{i,j}(t)) + c_2R_2(g_{best,i,j} - x_{i,j}(t)), (1)$$

$$x_{i,j}(t+1) = \begin{cases} 0 \text{ si rand}() \ge S(v_{i,j}(t+1)), \\ 1 \text{ si rand}() < S(v_{i,j}(t+1)), \end{cases}$$
(2)

$$S(v_{i,j}(t+1)) = 1/(1 + e^{(-v_{i,j}(t+1))}),$$
(3)

donde: $v_{i,j}(t)$ representa la velocidad de la n-ésima partícula, $x_{i,j}(t)$ representa la posición de la partícula, w es el coeficiente de inercia, c1 y c2 son los coeficientes

cognitivo y social, R1 y R2 son números aleatorios entre 0 y 1, $p_{best,i,j}$ es la mejor posición de la partícula, y $g_{best,i,j}$ es la mejor posición del cúmulo para Ecuación (1). $x_{i,j}(t+1)$ representa la posición de la i-ésima partícula, rand() es un numero aleatorio entre (0,1) y $S(v_{i,j}(t+1))$ representa el valor de la funcion sigmoide evaluada mediante el valor de $v_{i,j}(t+1)$. En cada una de las iteraciones, la velocidad modifica la posición de la partícula de acuerdo a las ecuaciones expuestas en [9] por lo que se puede considerar un algoritmo más simple que el algoritmo genético, más no por esto resulta ser un algoritmo menos completo.

BPSO con una mutación Lee et al. en [8] hace notar cierta tendencia de BPSO a la saturación de la función sigmoide, lo cual haría que las soluciones localizadas por BPSO tiendan a converger de forma prematura a un óptimo local, y que se vean comprometidas para lograr salir de este. Para ello propone una mutación mediante:

$$Para(i = 1; i < n; i = i + 1) \{ if(rand() < r_{mu}) then v_{i,jr}(t+1) = -v_{i,jr}(t+1) \},$$
(4)

donde r_{mu} es un la probabilidad de mutación, $v_{i,jr}$ es el elemento dentro de la matriz de velocidad en la ubicación (i, jr), la cual permitirá cambiar el sentido cuando los valores de la velocidad están muy cercanos a la velocidad máxima o la velocidad mínima y así evitar una convergencia prematura. La ecuaciónes (1) (2) y (3) permanecen sin cambios por lo que el pseudocódigo quedaría de la siguiente manera:

Algoritmo 1 BPSOm

```
1: INICIO
2: inicializar particulas x_{i,j}(t)
3: evaluar x_{i,j}(t)
4: mientras condicion\_de\_parada \neq \mathbf{cierto} hacer
5: v_{i,j}(t) \leftarrow actualizar \ v_{i,j}(t) {mediante (1)}
6: v_{i,j}(t) \leftarrow actualizar \ v_{i,j}(t) {por medio de (4)}
7: x_{i,j}(t) \leftarrow actualizar \ x_{i,j}(t) {por medio de (2) y (3)}
8: evaluar x_{i,j}(t)
9: t=t+1
10: fin mientras
11: FIN
```

MBPSO. En MBPSO el concepto de genotipo-fenotipo es aplicado a BPSO de la siguiente manera: considere la velocidad de BPSO como un genotipo y la posición binaria de BPSO como un fenotipo. Luego las operaciones de actualizacion de velocidad y posicion (1), (2) (3) de BPSO original y la operación de mutación (4) de BPSOm, cambian por las siguientes expresiones:

$$v_{(i,j)}(t+1) = wv_{(i,j)}(t) + c_1R_1(p_{best,i,j} - x_{i,j}(t)) + c_2R_2(g_{best,i,j} - x_{i,j}(t)),$$
 (5)

Comparación de la convergencia de cuatro algoritmos evolutivos al explorar conjuntos...

$$x_{q,i,j}(t+1) = x_{q,i,j}(t) + v_{i,j}(t+1), \tag{6}$$

$$para(i = 1; i < n; i = i + 1)$$

$$if(rand() < r_{mu})$$

$$then x_{q,i,jr}(t + 1) = -x_{q,i,jr}(t + 1),$$
(7)

$$x_{p,i,j}(t+1) = \begin{cases} 0 \text{ si rand}() \ge S(x_{g,i,j}(t+1)), \\ 1 \text{ si rand}() < S(x_{g,i,j}(t+1)), \end{cases}$$
(8)

$$S(x_{g,i,j}(t+1)) = 1/(1 + e^{(-x_{g,i,j}(t+1))}), \tag{9}$$

donde: $v_{i,j}(t)$ representa la velocidad de la n-ésima partícula, $x_{i,j}(t)$ representa la posición de la partícula, w es el coeficiente de inercia, c1 y c2 son los coeficientes cognitivo y social, R1 y R2 son números aleatorios ente 0 y 1, $p_{best,i,j}$ es la mejor posición de la partícula actual, $g_{best,i,j}$ es la mejor posición del cúmulo. No confundir $x_{g,i,j}$ con $g_{best,i,j}$; $x_{g,i,j}$ representa la matriz del genotipo calculada a través de (6). $x_{p,i,j}$ representa al fenotipo obtenido por medio de (7) (8) y (9), no confundir con $p_{best,i,j}$. El pseudocódigo es:

Algoritmo 2 MBPSO

```
1: INICIO
 2: inicializar particulas x_{p,i,j}(t)
3: evaluar x_{p,i,j}(t)
 4: mientras condicion\_de\_parada \neq cierto hacer
        v_{i,j}(t) \leftarrow actualizar \ v_{i,j}(t) \ \{\text{mediante } (5)\}
       x_{g,i,j}(t) \leftarrow actualizar \ x_{g,i,j}(t) \ \{\text{mediante } (6)\}
 6:
 7:
        x_{g,i,j}(t) \leftarrow mutar \ x_{g,i,j}(t) \ \{\text{por medio de } (7)\}
 8:
        x_{p,i,j}(t) \leftarrow actualizar \ x_{p,i,j}(t) \ \{\text{por medio de (8) y (9)}\}\
        evaluar x_{p,i,j}(t)
 9:
10:
        t=t+1
11: fin mientras
12: FIN
```

3. Simulación

Para las pruebas de optimización se utilizó la función uni-modal siguiente:

$$f(x) = \frac{\sin(\pi * x)}{256}, 0 \le x \le 255,$$
 (10)

y cuya gráfica se muestra en la Fig. 1.

Como se puede observar es una función que permite evaluar las convergencias de las soluciones de los algoritmos antes mencionados, encontrando un máximo

global en x = 128. Tiene áreas de interés entre los valores de x de 125 y 137 como puede observarse en el área verde de la Fig. 1.

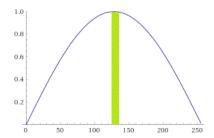


Fig. 1. Gráfica de la función objetivo.

Además, presenta un reto durante la exploración al tener la dificultad de cambiar su valor de x=127 a 128 al utilizar la codificación binaria, por la cantidad de bits que debe cambiarse.

Tal como se describe en las secciones anteriores la forma de representar las soluciones de estos algoritmos está determinada por cadenas binarias, auxiliándonos de las ecuaciones (11) y (12) para la codificación y decodificación de números de punto flotante, el proceso completo se encuentra descrito en [4]:

$$a_i \le (b_j - a_j) * (10)^n \le 2^{mj} - 1,$$
 (11)

$$x_{j} = a_{j} + decimal(substring) \times \frac{(b_{j} - a_{j})}{(2^{m_{j}} - 1)},$$

$$(12)$$

donde a_i representa el valor mínimo del área de exploración, b_j es el valor máximo del área de exploración, n la cantidad de dígitos después del punto decimal y m_j es la longitud de la cadena.

La ejecución de los algoritmos se realizó en un equipo de cómputo de las siguientes características Core2Duo a 2.1 GHz con 4 GB de RAM DDR3 a 1333 MHz, y un HDD de 320 GB a 5200 RPM. El área de exploración de los algoritmos estuvo limitada a los valores de [0, 255], con un óptimo global en x=128. Los parámetros utilizados para cada algoritmo genético BGA se encuentran listados en la Tabla 1 y para los BPSO y variantes en la Tabla 2.

Tabla 1. Parámetros utilizados en BGA con decodificación flotante y entera.

Parámetro	BGA	BGA		
	decodificación Flotante	decodificación Entera		
Porcentaje de cruce	0.25	0.25		
Porcentaje de mutación	0.1	0.1		
Longitud de la cadena de bits	22	8		
Cantidad de iteraciones	1000	1000		
Dígitos después del punto	3	3		

Tabla 2. Parámetros utilizados para BPSO decodificación entera y flotante.

Parámetro	decodi	ificación	Flotante	decod	lificación	Entera
	BPSO	BPSOm	MBPSO	BPSO	BPSOm	MBPSO
$[-V_{min}, V_{max}]$			[-4.0	,4.0]		
Longitud de la cadena						
de Bits		22			8	
c1 y c2	1.42694					
w	0.68343					
Probabilidad						
de mutación	N/A	0.3	0.3	N/A	0.3	0.3
Cantidad de iteraciones	1000					
Digitos despúes del punto	3 N/A					

Cada uno de los algoritmos fue programado en el lenguaje programación **Julia 6.2.2**, bajo el entorno de desarrollo Atom.

4. Resultados

Se realizaron 30 ejecuciones de cada uno de los algoritmos. Las gráficas en Figura 2(a) y Figura 3(a) pertenecen a las mejores ejecuciones y reportan los valores generados en cada iteración del BGA con codificación flotante y BGA con codificación entera respectivamente. Las gráficas en Figura 2(b) y Figura 3(b) representan el tiempo (en segundos) de esas mejores ejecuciones para realizar las operaciones de evaluación, cruce, mutación y evolución en cada una de las iteraciones.

Para algoritmos genéticos es apreciable que la convergencia de las soluciones resulta demasiado lenta, y tal como lo definen algunos autores se puede observar que su característica mayormente marcada es la exploración del área de soluciones. Por otro lado, en cuestión de tiempo como se puede ver en las Figura 2(b) y Figura 3(b), permiten concluir que la representación binaria para codificación flotante con 3 dígitos después del punto decimal, en BGA converge de manera más lenta que la codificación entera.

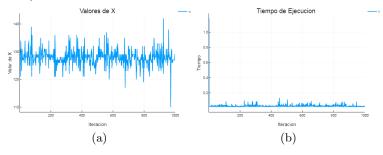


Fig. 2. Codificación entera de BGA para 50 individuos con 8 bits de longitud.

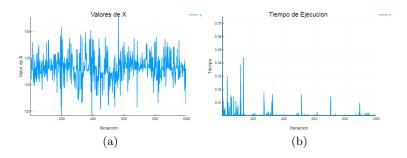
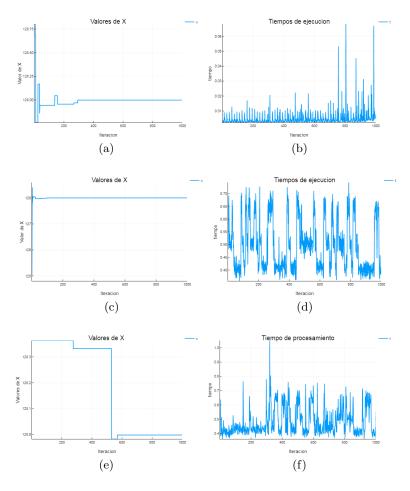


Fig. 3. Codificación flotante de BGA para 50 individuos con 22 bits de longitud.



 $\bf Fig.\,4.$ Codificación flotante para 50 partículas de los algoritmos: (a) BPSO; (c) BPSOm; y, (e) MBPSO.

Comparación de la convergencia de cuatro algoritmos evolutivos al explorar conjuntos...

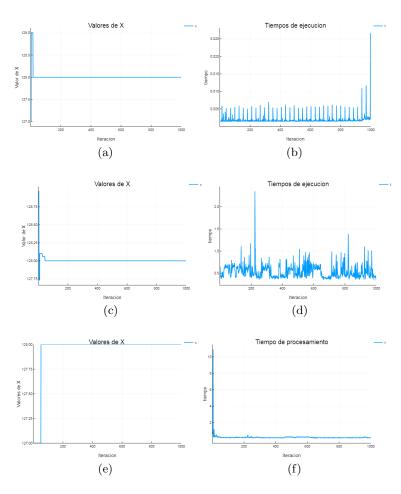


Fig. 5. Codificación entera para 50 partículas de los algoritmos: (a) BPSO; (c) BPSOm; y, (e) MBPSO.

Las gráficas en Figura 4 muestran las mejores ejecuciones para BPSO, BPSOm, y MBPSO con codificación flotante respectivamente. Las gráficas en Figura 4(a), Figura 4(c) y Figura 4(e) muestran las mejores ejecuciones para BPSO, BPSOm y MBPSO con codificación flotante respectivamente.

Las gráficas en Figura 4(b), Figura 4(d) y Figura 4(f) representan el tiempo de las mejores ejecuciones.

Las gráficas en Figura 5(a), Figura 5(c) y Figura 5(e) muestran las mejores ejecuciones para BPSO, BPSOm y MBPSO con codificación entera. Las gráficas en Figura 5(b), Figura 5(d) y Figura 5(f) representan el tiempo de las mejores ejecuciones respectivamente.

Las tablas 3 y 4 muestran las mediciones realizadas a 30 ejecuciones para cada algoritmo. Como puede apreciarse BPSOm y MBPSO generan resultados con mayor precisión que BGA. Además cabe resaltar la capacidad que tiene BPSOm para realizar una mejor aproximación al óptimo global con una cantidad de recursos mínimos. Se hace también perceptible el desempeño promedio de cada algoritmo, observando que MBPSO resulta ser el algoritmo con mejor desempeño promedio.

Tabla 3. Calidad de las soluciones para codificación entera.

Partículas	Algoritmo	Mejor	Peor aptitud	Mejor	Peor va-	Aptitud	Valor de x pro-	Tiempo
/Individuos		aptitud		valor de	lor de x	promedio	medio	promedio
				x				de ejecución
								(s)
	BGA	1	0.63439328416	128	56.0	0.98842140599	127.518933	0.1296936203
10	BPSO	1	0.91911385169	128	95.0	0.99995985537	127.518933	0.1296936203
10	BPSOm	1	0.89867446569	128	91.0	0.99997897453	127.518933	0.1298182704
	MBPSO	1	0.99729045667	128	122.0	0.99999560840	128.002333	30.579896232
	BGA	1	0.96377606579	128	126.0	0.99886441814	128.123666	0.2136438140
30	BPSO	1	0.99992470183	128	127	0.99999977410	127.999	0.7063252750
	BPSOm	1	0.99879545620	128	124.0	0.99999761568	127.983	81.428241680
	MBPSO	1	0.99932238458	128	131.0	0.99999859454	128.00466	87.903250711
50	BGA	1	0.99090263542	128	139.0	0.99961292221	128.0083333	0.4128423363
	BPSO	1	0.99992470183	128	129.0	0.99999974900	128.0013333	1.6418504633
	BPSOm	1	0.99247953459	128	118.0	0.99999842451	127.99926666	137.75230444
	MBPSO	1	0.99932238458	128	131.0	0.99999957333	128.003	153.73698562

Tabla 4. Calidad de las soluciones para codificación de punto flotante.

Partículas	Algoritmo	Mejor aptitud	Peor aptitud	Mejor valor	Peor valor de	Aptitud	Valor de x pro-	Tiempo pro-
/Individuos				de x	x	promedio	medio	medio de eje-
								cución (s)
	BGA	0.99999999998	0.5791642999	127.9998404	205.6650163	0.9884305	127.8960934	0.1281748
10	BPSO	0.99999999999	0.740624959	127.9999620	67.9604334	0.9999446	127.9989473	0.1561739
10	BPSOm	0.99999999999	0.8043359628	128.0000228	76.15455893	0.9999592	127.9754834	69.554611
	MBPSO	0.99999999998	0.9822086929	128.0001444	143.3941253	0.9999919	127.9910014	63.779639
30	BGA	0.99999999999	0.8350242862	128.0000228	80.52389991	0.9969653	127.8809693	0.1715704
	BPSO	0.99999999998	0.9884961460	128.0001444	115.6278838	0.9999973	127.9991538	1.0581364
	BPSOm	0.99999999999	0.9868285867	128.0000228	141.2403395	0.9999959	127.9854557	219.63925
	MBPSO	0.99999999999	0.9953752558	128.0000228	135.840007	0.9999989	128.0004667	184.40917
	BGA	0.99999999999	0.9740595627	128.0000228	146.6010324	0.9995684	127.9390172	0.4003717
150	BPSO	0.99999999999	0.9970689913	127.9390172	134.2405055	0.9999988	128.0009653	2.9002064
	BPSOm	0.999999999999	0.9931025973	128.0000228	137.5763017	0.9999978	127.9864345	359.92917
	MBPSO	0.9999999999999	0.9992821536	128.0000228	124.9122166	0.9999995	127.9970819	398.89039

Considerando la información de las tablas, la convergencia observada en los algoritmos de cúmulo de partículas resulta interesante al observar que en las evaluaciones de 10 partículas, son capaces de alcanzar una buena aproximación. En algunos casos mejor a la de GA, sin consumir mayores recursos (poblaciones grandes) lo cual los vuelve candidatos a ser utilizados en la búsqueda de soluciones, cuando los recursos y/o el tiempo es una variable a considerar.

Se debe hacer mención que la propiedad de memoria de cúmulo de partículas, le permite tener un parámetro de mejora (experiencia) y es esto lo que se puede apreciar en las gráficas de convergencia. Una vez que BPSO ha encontrado una solución suficientemente buena, la probabilidad de que esta cambie está determinada por la exploración.

5. Conclusión y trabajo futuro

La exploración es mejorada por las variantes de BPSOm y MBPSO. Estas evitan un atrapamiento en óptimos locales con convergencia rápida y mejoran la exploración, sin alterar la capacidad de aprendizaje de BPSO.

Durante las pruebas se pudo apreciar la capacidad que tiene BPSO y sus variantes sobre BGA para localizar de forma rápida un área de soluciones adecuada, aun cuando sus recursos son limitados. Si bien el tiempo es un parámetro interesante donde BGA supera a BPSO y sus variantes. Se puede visualizar que BPSOm con 10 partículas logra una aptitud semejante a BGA con 50 individuos y, en promedio, su fitness resulta mayor. Por lo que se puede considerar el tiempo como una variable de poco impacto en el análisis de la convergencia.

En general BPSOm registra mejores valores tanto para aptitud como para valores de x que cualquiera de los otros algoritmos analizados. Como trabajo futuro, se implementarán las variantes de BPSO presentadas en esta investigación, para ubicación de vehículos aéreos no tripulados en zonas de desastre. Además, se cambiará el enfoque empírico hacia un análisis cuantitativo del comportamiento de cada uno de los algoritmos.

Agradecimientos. Trabajo apoyado por CONACyT mediante beca nacional.

Referencias

- Eiben, A.E., Smith, J.E.: Theory. In: Introduction to Evolutionary Computing. Springer (Jan 2015)
- 2. Goldberg, D.E.: Genetic algorithms. Pearson Education India (2006)
- 3. Holland, J.H.: Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press (1992)
- 4. Houck, C.R., Joines, J., Kay, M.G.: A genetic algorithm for function optimization: a Matlab implementation. Ncsu-ie tr 95(09), 1–10 (1995)
- Kachitvichyanukul, V.: Comparison of three evolutionary algorithms: GA, PSO, and DE. Industrial Engineering and Management Systems 11(3), 215–223 (2012)
- 6. Kennedy, J.: Particle swarm optimization. In: Encyclopedia of machine learning, pp. 760–766. Springer (2011)
- Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. In: Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, IEEE International Conference on. vol. 5, pp. 4104–4108. IEEE (1997)
- 8. Lee, S., Soak, S., Oh, S., Pedrycz, W., Jeon, M.: Modified binary particle swarm optimization. Progress in Natural Science 18(9), 1161–1166 (2008)
- Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution. Springer: New York (1992)

Cuerpos y planos geométricos usando realidad aumentada y computación afectiva

Ramón Zatarain-Cabada¹, María Lucia Barrón-Estrada¹, María Blanca Ibañez-Espiga², Aldo Uriarte-Portillo¹

¹Instituto Tecnológico de Culiacán, Culiacán, Sinaloa, México

² Universidad Carlos III de Madrid, Departamento de Ingeniería Telemática, España {rzatarain, lbarron, auriarte}@itculiacan.edu.mx, mbibanez@it.uc3m.es

Resumen. La adquisición de competencias en Ciencia, Tecnología, Ingeniería y Matemáticas será esencial para que las nuevas generaciones puedan enfrentar el competitivo mundo donde les tocará desenvolverse. Si bien nuestros estudiantes están rodeados de estímulos tecnológicos que los apasionan y a veces distraen, los sistemas de aprendizaje no están aprovechando estas tecnologías para atraer la atención de los estudiantes, facilitarles la comprensión de los temas abstractos propios de estas asignaturas. El trabajo que aquí se presenta pretende contribuir a la enseñanza de los temas "Cuerpos y Planos Geométricos" facilitando el aprendizaje gracias a las posibilidades de visualización tridimensional e interacción en tiempo real que ofrece la tecnología de realidad aumentada e incrementando el nivel de motivación utilizando técnicas de computación afectiva. A partir de los requerimientos educativos realizados por 7 profesores de diferentes escuelas secundarias, se diseñaron actividades en un entorno de realidad aumentada para facilitar el aprendizaje de "Cuerpos y Planos Geométricos". Las actividades están integradas en un entorno educativo utilizando técnicas de computación afectiva.

Palabras clave: realidad aumentada, inteligencia artificial, computación afectiva, reconocimiento de emociones, aprendizaje móvil.

Geometric Shapes and Planes Using Augmented Reality and Affective Computing

Abstract. The acquisition of competences in Science, Technology, Engineering and Mathematics will be essential so that the new generations can face the competitive world where they will have to progress. Although our students are surrounded by technological stimuli that are passionate and sometimes distracting, the learning systems are not taking advantage of these technologies to attract the attention of the students, to facilitate their understanding of the abstract topics of these subjects. The work presented here aims to contribute to the teaching of the topics "Geometric Shapes and Planes" by facilitating learning thanks to the possibilities of three-

dimensional visualization and real-time interaction offered by augmented reality technology and by increasing the level of motivation using techniques of Affective Computing. Based on the educational requirements made by 7 teachers from different middle and high schools, activities were designed in an augmented reality environment to facilitate the learning of "Geometric Shapes and Planes". The activities are integrated into an educational environment using affective computing techniques.

Keywords: augmented reality, artificial intelligence, affective computing, affective recognition, mobile learning.

1. Introducción

La Agenda 2030 para el Desarrollo Sostenible [1] propone un plan de acción para enfocar de manera integrada las causas que permitan la superación de la pobreza, para orientar el desarrollo hacia la sostenibilidad y el establecimiento de una calidad de vida digna para todos. Los aportes de las ciencias, la tecnología, la innovación son indispensables para que se avance en las direcciones deseadas [2] La educación científica, en la educación obligatoria, debe asegurar a todos sus estudiantes aprendizajes de calidad. Sin embargo, el escenario de la región latinoamericana muestra que la educación no solo no brinda estos aprendizajes, sino que la manera como se presenta el conocimiento científico tiende a que los jóvenes pierdan el interés por aprender ciencias, y no se despierten vocaciones científicas [2].

En los últimos años, los investigadores están explorando las posibilidades educativas de nuevas tecnologías emergentes como la realidad aumentada, los mundos virtuales guiados por métodos como la computación afectiva, y la gamificación para diseñar nuevos entornos educativos que faciliten y mejoren el aprendizaje. Las nuevas tecnologías aplicadas a la educación están empezando a producir aplicaciones con nuevos métodos como es el caso de la realidad aumentada [3,4] o los sistemas reconocedores de emociones [5,6] cuya aplicación pueden facilitar el aprendizaje. La realidad aumentada (RA) es la tecnología que permite aumentar la percepción del mundo real con información digital superpuesta en el mundo real. La información puede consistir en texto, imágenes, audio, video. El usuario tiene la posibilidad de interactuar en tiempo real con la información digital, esto incluye la observación tridimensional de la información digital. Esta tecnología requiere de elementos visuales que permitan mostrar la información relevante en cada momento. Estos elementos son imágenes o puntos geolocalizados.

En 2016 hubo un incremento del 5% del uso de dispositivos móviles respecto a 2015 [7]. El tiempo que los usuarios dedican a usar un dispositivo móvil cada vez es mayor. Hoy en día un gran número de personas con diferente edad, género, condición social, nivel educativo usan algún dispositivo móvil desde cualquier lugar, a cualquier hora, y con ello pueden realizar alguna actividad recreativa, capturar algún momento importante, o incluso realizar alguna tarea de su área de trabajo. Estas acciones generan una gran cantidad de información de diversos tipos que puede ser utilizada

con fines de investigación, tal es el caso del reconocimiento de emociones en tiempo real desde un ambiente natural.

Por otra parte, la enseñanza de las Matemáticas, particularmente el área de Geometría, causa dificultades a los estudiantes debido al esfuerzo de abstracción que deben realizar para imaginar cuerpos en el espacio tridimensional cuando las explicaciones se realizan en pizarrón, un espacio bidimensional. La posibilidad que la tecnología de RA brinda de poder interactuar en tiempo real con elementos digitales tridimensionales, brinda potencialmente dos posibilidades. Por un lado facilitar la comprensión y ejercitación de los temas de geometría a tratar. Por otro lado, ayudar a aumentar los niveles de interés y concentración de los estudiantes involucrándolos en el aprendizaje activo que será soportado por el entorno educativo que aquí se presenta.

La principal contribución de este trabajo es el diseño y la implementación de un entorno de aprendizaje que permite integrar la tecnología de realidad aumentada con un sistema reconocedor de emociones. El diseño se realiza a partir del análisis de las necesidades educativas en un entorno real a partir de las opiniones de profesores que dictan el tema de Geometría en educación Secundaria en México El ambiente complementa un entorno real superponiendo información digital a través de modelos en 3D, logrando obtener la atención de los estudiantes al visualizar los elementos geométricos desde la perspectiva elegida, fomenta el aprendizaje activo y por último, adapta las actividades a realizar en función de los estados de ánimo del estudiante.

Este artículo está estructurado en el siguiente orden: en la Sección 2 presentamos los trabajos relacionados en las áreas de computación afectiva y en realidad aumentada orientada a la educación; en la Sección 3 se presenta un análisis de las entrevistas realizadas con los profesores; en la Sección 4 se plantea la estructura del entorno de aprendizaje; en la Sección 5 se describen las evaluaciones de la herramienta de RA realizadas con profesores, y para finalizar en la sección 6 se presentan las conclusiones y trabajos futuros.

2. Trabajos relacionados

En esta sección se describen algunos trabajos de investigación relacionados en el campo de la computación afectiva y de realidad aumentada en la educación. Estos trabajos guardan similitud con este trabajo de investigación y fueron considerados como punto de partida para el desarrollo de este proyecto.

En el reconocedor facial presentado por Khandait *et al.* [8] detecta una imagen y realiza el reconocimiento facial con base en distancias entre rasgos faciales. Este sistema utiliza como clasificador a las redes neuronales artificiales con retro propagación y reconoce las emociones básicas de sorprendido, neutral, triste, disgustado, feliz y enojado. Para la extracción de características implementa SUSAN para la detección de bordes en cejas, ojos, boca y nariz, alcanzando una precisión de efectividad de 95.26%.

El proyecto propuesto por Patwardhan et al. [9] es un sistema desarrollado que utiliza múltiples modalidades, como cara, mano, cabeza y habla para reconocer el

ISSN 1870-4069

estado afectivo de un individuo, por ello utilizan un dispositivo que permite detectar color y profundidad como el Microsoft Kinect; con este hardware pueden realizar la extracción de características del rostro, y del movimiento de articulaciones. El autor señala que se centra en las emociones básicas. Utiliza como clasificador una Maquina de Vector Soporte (SVM por sus siglas en inglés) para el reconocimiento de emociones en cada modalidad. Patwardhan y sus colegas, obtuvieron una precisión de reconocimiento de emociones en rostro del 84% en las emociones de felicidad, 77% en la emoción de enojo y 71% en tristeza, en tanto modo multimodal fue del 88%, 83% y 77% respectivamente.

El proyecto de Wang et al. [10] también implementa como clasificador una SVM por su buen rendimiento para el reconocimiento de patrones, específicamente en el reconocimiento de emociones a través de la voz. El clasificador fue entrenado con base de datos con voz etiquetada del idioma alemán, chino, y emociones de la antigua china. El modelo de datos del proyecto implementa los parámetros de Fourier (FP por sus siglas en inglés) y el coeficiente cepstral de frecuencias Mel (MFCC). Los resultados obtenidos tienen una precisión promedio del 87.5%.

En el marco del uso de la realidad aumentada en el campo educativo, los estudios iniciales han estado dedicados principalmente a estudiar factores como la usabilidad y la motivación en prototipos educativos. Aplicaciones basadas en marcadores se han utilizado para ayudar a los estudiantes a mejorar sus habilidades espaciales. Así, Liao et al. [11] desarrollaron una herramienta en Unity, Vuforia y JavaScript como motor de scripts que facilita a los estudiantes resolver el problema de un cubo de Rubik. Ellos implementan la RA para mostrar pistas en el proceso de solución del cubo de Rubik además de aprender conceptos elementales de volumen y superficie de cuerpos geométricos. El proyecto está en fase de pruebas con el fin de evaluar la eficiencia del aprendizaje esperado en los estudiantes.

El proyecto ARMatika desarrollador por Young et al. [12] es un juego que implementa un conjunto de elementos enfocados a mejorar el aprendizaje del estudiante, utilizando efectos visuales, técnica de puntajes, reglas de juego, movimiento de personajes y donde se tiene un numero de aciertos y errores permitidos. El porcentaje de preferencia de sus estudiantes para aprender es del 85.3%.

En el estudio realizado por del Cerro et al. [13] abordan temas de figuras geométricas adaptadas al plan de estudio de tercer grado de secundaria, con herramientas como Junaio y Sketchup. Las actividades educativas consistieron en ejercicios prácticos en donde usando marcadores QR convencionales obtenían diferentes vistas de la figura solicitada. El 82.6% quedo satisfecho con el uso de la herramienta diseñada y el 78.26% de los participantes afirman haber comprendido con mayor claridad el tema visto en clase al usar la herramienta.

3. Análisis de entrevistas a profesores

El primer paso antes de realizar el diseño del entorno de aprendizaje fue entrevistar a un grupo de 7 profesores que imparten la materia de matemáticas en el segundo y tercer grado de escuelas mexicanas. De acuerdo a un análisis que llevamos a cabo de encuestas abiertas realizadas a los profesores podemos establecer que:

- (1) 6 de 7 docentes afirman que la enseñanza de la Geometría es un reto debido a la necesidad de utilizar un instrumento como el pizarrón para explicar conceptos tridimensionales.
- (2) 5 de 7 docentes indicaron que los estudiantes manifiestan poco interés en el estudio de temas ligados a las matemáticas.
- (3) 6 de 7 docentes indican que a sus estudiantes les atrae más el uso de un dispositivo móvil que hacer actividades en su pupitre con papel y lápiz.
- (4) 4 de 7 docentes indican que los estudiantes por timidez o temor a equivocarse participan poco en las actividades en el aula.
 - (5) 4 de 7 docentes indican que el nivel de los estudiantes en sus clases es variable.

Con base a lo anterior podemos establecer las siguientes implicaciones que corresponden a cada una de las respuestas de los profesores:

La respuesta 1 implica que la RA puede ser el uso de la tecnología de realidad aumentada para la realización de actividades educativas en el campo de la Geometría porque la realidad aumentada le permite la visualización en tiempo real de fenómenos que ocurren en espacio 3D, logrando disminuir la dificultad para comprender, visualizar, imaginar y asociar elementos geométricos.

La respuesta 2 implica que la RA puede ser útil, ya que estudios como [11,12 13] indican que la RA fomenta la motivación de los estudiantes, llamando su atención, comprometiéndolos a realizar la actividad y con ello facilitar su aprendizaje.

La respuesta 3 implica que la RA puede ser útil puesto que requiere de uso de dispositivos móviles con los cuales los jóvenes estudiantes están familiarizados.

La respuesta 4 implica que la RA tendría un efecto positivo porque permite la interacción individual, la retroalimentación en el mismo momento con evaluación formativa, aumentando la actividad del estudiante en el tema.

Respuesta 5 implica que la RA combinada con la computación afectiva puede tener impacto positivo ya que permite colocar elementos de andamiaje educativo que se ajusten al perfil general y actual de cada estudiante.

4. Estructura del entorno de aprendizaje

En esta sección se presenta la arquitectura de la aplicación y los marcadores diseñados para el ambiente de aprendizaje.

4.1. Arquitectura

ARGeo es una aplicación móvil desarrollada en Unity implementando el sistema de reconocimiento de marcadores Vuforia y Scripts programados en C#, la cual fue diseñada para los sistemas operativos Android e IOS. ARGeo le muestra al estudiante un conjunto de ejercicios de área, volumen y cortes aplicables a cuerpos geométricos; mientras el estudiante interactúa con la aplicación, ésta recolecta una imagen y clasifica la emoción detectada mostrándola en pantalla. El componente principal del

dispositivo móvil que tiene función en este proyecto es la cámara. En la Figura 1 se presenta la arquitectura que fue diseñada para esta aplicación, la cual está formada por los componentes Vista, Control y Reconocedor.

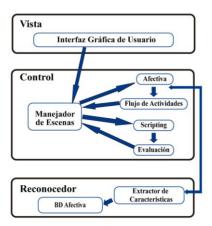


Fig. 1. Arquitectura de ARGeo.

La capa **Reconocedor** fue implementada mediante una red neuronal convolucionada (CNN) utilizada para identificar las emociones a través de imágenes tomadas a los estudiantes [14]. La CNN contiene tres capas convolucionales, tres capas de tipo "max-pooling" y tres redes neuronales con conexiones intermedias de tipo "dropout" (ver Figura 2).

La CNN fue entrenada usando tres diferentes bases de datos afectivas: una base de datos con emociones actuadas y dos bases de datos de emociones espontáneas creadas especialmente en el Laboratorio de Computación Móvil y Afectiva del Instituto Tecnológico de Culiacán, con un contenido orientado en las emociones centradas en el aprendizaje (emocionado, comprometido, aburrido o frustrado). La red neuronal se encarga de extraer las características del rostro del estudiante y de acuerdo a ellas reconocer la emoción del mismo. La emoción es enviada al módulo Afectiva para después ser utilizada por el módulo de flujo de actividades.

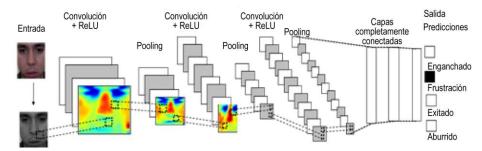


Fig. 2. Arquitectura de una CNN para Reconocer Emociones.

En la capa **Control** se efectúa el manejo de escenas mediante la superposición de la información digital sobre el entorno real; esta información digital puede ser audio, video, imagen, elementos 3D y animaciones. El módulo Afectiva quien recibe del reconocedor la emoción actual del estudiante, guía al módulo de Flujo de Actividades para mostrar al estudiante las posibles actividades que él requiera. El estudiante ejecuta una actividad mediante un script, realiza la interacción y el módulo de evaluación recibe cómo fue la actuación del estudiante. El módulo de evaluación le indica al estudiante cómo lo hizo y esto genera un posible cambio en las emociones del estudiante. Este estado de ánimo es recogido por el módulo de computación afectiva.

Para poder realizar realidad aumentada se requiere de diversos elementos: una cámara que capte una imagen del espacio real, un programa que interprete y reconozca la imagen captada y que pueda hacer la fusión con el entorno real, y una pantalla donde se pueda observar la combinación de la realidad con elementos virtuales y los marcadores [14]. Para conseguirlo se utiliza el entorno Unity 2017 [15] con Vuforia 7.2 [16]. El dispositivo desde el cual se utiliza la aplicación debe contar con una cámara. Para que se efectúe el reconocimiento de un marcador es necesario colocarlo en el campo de visión de la cámara, de tal manera que la cámara pueda detectar el patrón adecuadamente. En este proceso juega un rol importante el motor de realidad aumentada (Vuforia), el cual al reconocer el marcador superpone los elementos digitales (e.g. imágenes, objetos 3D o animaciones) sobre el objeto, para poder asignar algún proceso o realizar alguna operación en específico y así tener el objeto ya aumentado [17]. En este proyecto se ha utilizado el uso de comportamientos de colisiones y de movimiento de objetos.

Por último, la capa **Vista** contiene la interfaz principal que está formado por un menú que permite elegir una de las tres actividades educativas (ejercicios) diseñadas. En la Figura 3 podemos apreciar su menú principal y una vista previa de cada ejercicio. Se implementaron tres ejercicios:



Fig. 3. Interfaz principal del Ambiente de Realidad Aumentada.

El primer ejercicio consiste de cálculo de área y volumen de prismas regulares donde el estudiante debe reconocer el marcador que hace referencia al cuerpo geométrico que la aplicación le está solicitando. Si el estudiante acierta al seleccionar el marcador se plasman en pantalla la información necesaria para realizar los cálculos de área de la base y de volumen; en tanto en caso contrario se despliega un mensaje indicando al usuario que se ha equivocado.

El segundo ejercicio consiste en ver en tiempo real como se van acumulando volúmenes de prismas regulares. Para ello, el usuario debe colocar el marcador que hace visible el prisma regular al cual se pretende obtener textura de él. En la Figura 4 se puede apreciar este procedimiento. Una vez visible, se requiere elegir la medida correspondiente para llenar de textura el objeto. Con un marcador denominado cubo se realizan las colisiones a los prismas objetivo y en pantalla se va visualizando el llenado de textura de un prisma a otro, mientras se aprecia el incremento del volumen de la textura del prisma regular y en pantalla se refleja en centímetros cúbicos el tamaño textura que se ha obtenido hasta la última colisión. El ejercicio culmina al completar el cien por ciento de la textura en el prisma contenedor.



Fig. 4. Marcador obteniendo textura del objeto al colisionar con otro objeto.

El tercer ejercicio consiste en identificar correctamente las secciones de corte a cuerpos geométricos. Este ejercicio aplica la lluvia de objetos con el afán de que el estudiante vea desde diferentes ángulos y posturas el corte que se le pide y con ello reforzar su aprendizaje en el tema.

5. Evaluación con profesores

En esta sección abordaremos la evaluación y retroalimentación que los profesores de la asignatura de Matemáticas llevaron a cabo. A cada profesor se le explicó el objetivo de evaluar la herramienta de aprendizaje aplicando la tecnología de realidad aumentada. Posteriormente los profesores probaron el funcionamiento de la herramienta de RA directamente en el dispositivo móvil. El proceso de evaluación se clasificó en dos fases: la primera fase consiste en evaluar la parte pedagógica de la

aplicación, donde los profesores nos expresaron que el ambiente de aprendizaje es factible aplicarlo a segundo y tercer grado. Además, nos argumentaron que los alumnos debían contar con algunos conocimientos previos como conceptos de área, volumen e identificación de cuerpos geométricos.

En la parte pedagógica los profesores hicieron hincapié en el aspecto motivacional del estudiante, donde los profesores consideraron que el ambiente de aprendizaje influye en fomentar la motivación del estudiante por aprender temas de geometría. También se consideraron algunos aspectos como, el tiempo que toma comprender el funcionamiento de cada actividad, las repeticiones de cada ejercicio para que el alumno domine su funcionamiento y la reflexión del aprendizaje adquirido. En la segunda fase los profesores evaluaron la usabilidad de la aplicación, expresando, si la herramienta les resultó innovadora y fácil de usar por parte de los alumnos, concluyendo que les sería muy útil y que les gustaría implementarlo en sus prácticas docentes. Los profesores interactuaron con la aplicación y nos expresaron algunas observaciones que nos permitieron mejorar el ambiente de aprendizaje.









Fig. 5. Evaluación del ambiente por los Profesores.

Tabla 1. Comentarios relevantes por parte de los Profesores.

	<u> </u>
Docente	Comentario
Profesor 1	Los adolescentes están en la era de la tecnología. Para ellos todo lo que es tecnología y algo diferente en áreas lúdicas es muy motivador. La aplicación ayudaría a reforzar y mejorar el conocimiento del tema, puesto que el alumno es capaz de ver la geometría desde otra perspectiva.
Profesor 2	Respecto a los temas que se abordaron la aplicación es de mucha utilidad, fomenta la motivación, permite que el alumno tenga un aprendizaje mayor y de una forma sencilla de comprender y va de acuerdo a la época en que se está viviendo.
Profesor 3	La aplicación de RA les ayuda mucho a visualizar en que están trabajando, mostrarles de manera más clara lo que ellos pueden ver, manipular, y conocer fácilmente en comparación a si lo observas a través de una formula en el pizarrón, donde el conocimiento no queda tan abstracto y ni tan claro.
Profesor 4	Sería una herramienta más para mí y tendría un valor agregado para mi materia. Considero que motivaría a mis alumnos, aprenderían más, no los aburriríamos con lo mismo, el libro, la libreta, y de este modo los motivaría más hacia la asignatura de matemáticas que por lo regular no les agrada.
Profesor 5	La verdad esta herramienta ayudaría muchísimo y atacaríamos el problema de identificación de cuerpos, donde no saben identificar el nombre de cada cuerpo. Al momento de ver los cuerpos y manipularlos les queda más claro, ya que pueden observar la base, manipular el cuerpo de tal forma que les queda más claro. La conclusión a la que pueden llegar es que al momento de estar llenando un recipiente de manera virtual, la fórmula de volumen le queda más claro.

En la Figura 5 podemos observar a algunos de los profesores interactuando con la herramienta y en Tabla 1 podemos apreciar los comentarios más relevantes que los docentes hicieron respecto al entorno de aprendizaje.

6. Conclusiones y trabajos futuros

El sistema desarrollado es capaz de realizar el reconocimiento de emociones utilizando la información de la cámara de un dispositivo móvil a través de la interacción con los ejercicios. Cabe destacar que el reconocedor está diseñado para que interactúe con ARGeo o alguna otra aplicación que requiera de reconocer emociones. El sistema reconocedor solo está disponible a través de un servicio web que recibe como parámetro la imagen en formato de datos binarios que la conforman y da como resultado una clase, que equivale a la emoción que le corresponde.

La primera evaluación que se llevó a cabo de la herramienta de RA y computación afectiva aquí mostrada fue con profesores que imparten la materia de matemáticas a nivel secundaria. De esta evaluación se concluyó que los profesores creen que los estudiantes podrían sentirse más motivados en su aprendizaje y que los temas abordados sobre cuerpos geométricos tridimensionales pueden ser más simples de comprender. Por otra parte, los profesores hicieron algunas observaciones sobre detalles del software que fueron tomadas en cuenta y en su caso corregidas.

Dentro del marco de los trabajos futuros se encuentran a corto plazo el iniciar pruebas con estudiantes de las 7 instituciones educativas para así analizar el impacto de nuestra herramienta con los estudiantes, tanto en aspectos de motivación y usabilidad como en ganancia de aprendizaje. También queremos personalizar el aprendizaje de tal forma que se tomen en cuenta los resultados de los ejercicios más el estado afectivo del estudiante para determinar rutas de aprendizaje para los estudiantes.

Agradecimientos. Agradecemos a las instituciones educativas Colegios Sinaloa, A.C., Colegio Chapultepec, Escuela Secundaria Técnica # 19, Escuela Secundaria 24 de agosto y a la Escuela Secundaria Técnica # 92 ubicadas en la ciudad de Culiacán, Sinaloa, así como a la Escuela Secundaria Técnica # 38 ubicada en Guasave, Sinaloa, que nos han apoyado desde un inicio en este proyecto para poder llevar a cabo las evaluaciones de nuestro software. Asimismo, agradecemos a los docentes del área de geometría que con su experiencia contribuyeron al diseño de los ejercicios contemplados en este trabajo.

Referencias

- United Nations: Transforming our world: the 2030 Agenda for Sustainable Development. General Assembly Resolution 70/1, pp. 1–40. Recuperado a partir de http://unctad.org/meetings/es/SessionalDocuments/ares70dl_es.pdf (2015)
- Macedo, B.: Educación científica. Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura–UNESCO (2016)
- 3. Bower, M., Howe, C., McCredie, N., Robinson, A., Grover, D.: Augmented Reality in

- education–cases, places and potentials. Educational Media International, 51(1), pp. 1–15 (2014)
- 4. Radu, I.: Augmented reality in education: a meta-review and cross-media analysis. Personal and Ubiquitous Computing, 18(6), pp. 1533–1543 (2014)
- Corneanu, C. A., Simón, M. O., Cohn, J. F., Guerrero, S. E.: Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affectrelated applications. IEEE transactions on pattern analysis and machine intelligence, 38(8), pp. 1548–1568 (2016)
- Zatarain-Cabada, R., Barrón-Estrada, M. L., Hernández, F. G., Alor-Hernandez, G.: Evaluando afecto en un entorno de aprendizaje para Java. Research in Computing Science, 111, pp. 123–133 (2016)
- México edition, Deloitte Global Mobile Consumer Survey, May Jun 2015, May Jun 2016 (2016)
- 8. Khandait, S. P., Thool, R. C., Khandait, P. D.: Automatic facial feature extraction and expression recognition based on neural network. arXiv, pp.1204–2073 (2012)
- 9. Patwardhan, A., Knapp, G.: Multimodal Affect Recognition using Kinect. arXiv preprint arXiv: 1607–02652 (2016)
- 10. Wang, K., An, N., Li, B. N., Zhang, Y., Li, L.: Speech emotion recognition using fourier parameters. IEEE Transactions on affective computing, 6(1), pp 69–75 (2015)
- Liao, Y. T., Yu, C. H.Wu, C. C.: Learning geometry with augmented reality to enhance spatial ability. In: Learning and Teaching in Computing and Engineering (LaTiCE), 2015 International Conference on IEEE, pp. 221–222 (2015)
- 12. Young, J. C., Kristanda, M. B., Hansun, S.: ARmatika: 3D game for arithmetic learning with Augmented Reality technology. In: Informatics and Computing (ICIC), International Conference on IEEE, pp. 355–360 (2016)
- del Cerro Velázquez, F., Méndez, G. M.: Realidad Aumentada como herramienta de mejora de la inteligencia espacial en estudiantes de educación secundaria. Revista de Educación a Distancia, pp 54 (2017)
- Gonzalez-Hernandez, F., Zatarain-Cabada, R., Barrón-Estrada, M.L., Rodríguez-Rangel,
 H.: Recognition of learning-, entered emotions using a convolutional neural network. In:
 Mexican International Conference on Artificial Intelligence. Springer, Cham (2017).
- 15. Unity Homepage (2018)

ISSN 1870-4069

- 16. Vuforia Homepage, Developer portal (2018)
- 17. Patil, S., Prabhu, C., Neogi, O., Joshi, A. R., Katre, N.: E-learning system using Augmented Reality. In: Computing Communication Control and automation (ICCUBEA), 2016 International Conference on IEEE, pp. 1–5 (2016)

Ernest: sistema embebido para control de casas inteligentes mediante correos electrónicos con cifrado SSL

José Francisco Rodríguez Arellano, Emmanuel Dávila Delgado, Mario Alberto Ruíz Durán, Martín Isaac Falcón Segovia, Salvador Abraham Medina Rangel, Cristian Jael Mejía Aguirre

Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación Aguascalientes, Aguascalientes, México

{francisco.rodriguez,emmanuel.davila,mario.ruiz,cristian.mejia} @infotec.mx, elontesla@gmail.com, salami.95@hotmail.com

Resumen. El funcionamiento de los servicios de domótica o servicios orientados a las SmartHouse dio pie a la aparición de Ernest, aplicación orientada a transformar una vivienda o construcción con acceso a internet y a electricidad en una casa inteligente. Ernest hace uso de los servicios de correo electrónico sin necesidad de tener un contrato con una empresa que permita conectarse a la nube. La implementación de un circuito electrónico lleva a cabo esta tarea; en este artículo se incursiona en la arquitectura de este circuito que integra el Internet de las cosas y habilita o deshabilita cuatro tomacorrientes. Se presentan también el desarrollo de tres algoritmos que permitan acceder y usar los correos electrónicos para establecer comunicación usuario-tarjeta y viceversa, con la finalidad de controlar o informar del estado en que se encuentran sus tomacorrientes, así como filtrar los e-mails recibidos. Estos algoritmos permiten que el sistema embebido se conecte a internet haciendo uso del protocolo de seguridad SSL y agilicen la lectura de los nuevos correos electrónicos en la bandeja de entrada y, en caso de ser necesario, el envío de los estados de los tomacorrientes.

Palabras clave: internet de las cosas, domótica, sistemas embebidos.

Ernest: Embedded System to Control Smart Homes through Emails with SSL Encryption

Abstract. The performance of home automation services or SmartHouse-oriented services gave rise to the emergence of Ernest, an application aimed at transforming a home or building with access to the Internet and electricity in a smart home. Ernest makes use of email services without hire a company that allows to connect to the cloud. The implementation of an electronic circuit accomplishes this task; In this paper, we explore the architecture of this circuit that integrates the Internet of Things and enables or disables four power outlets. There is also the development of three algorithms that allow access and use of

emails to establish user-card communication, in order to control or report the status of their power outlets, as well as filter the e-mails received. These algorithms allow the embedded system to connect to the internet using the SSL security protocol and streamline the reading of new emails in the inbox and, if necessary, the sending of the current status of the power outlets.

Keywords: internet of things, domotics, embedded systems.

1. Introducción

Hoy la mayoría de los servicios para SmartHouse venden una aplicación que permite el acceso a la nube a través de la cual se ponen en contacto con las cosas, desde sensores hasta encendido de luces. Esto implica otorgarle información personal a una empresa que debe contratarse para la instalación o en caso de asistencia técnica lo que las pone en la mira de los hackers; por tal motivo, empresas como SmartThings [1] han optado por hacer uso de las redes sociales con la finalidad de evitar el hurto de información, bien sea de los usuarios o de la empresa misma.

Sin embargo, el despliegue de comunicación automática de objetos en nuestras vidas, de manera natural, representa un peligro para el futuro al poner en riesgo nuestra información personal [2]. Esto se debe a que la información en la nube trae grandes interrogantes y algunos problemas, dado que la información es vulnerable a ataques de hackers, o frente a malware o virus por lo que se buscan los errores de comunicación y las posibles amenazas a la seguridad tanto en la protección de datos, como en la privacidad de los usuarios [3,4,5]. En consecuencia, los ataques de seguridad se vuelven sumamente problemáticos para el IoT amenazándolas desde la accesibilidad física a sensores, actuadores y objetos, hasta la apertura de los sistemas donde se incluye el hecho de que la mayoría de los dispositivos se comunican de forma inalámbrica [6]. Por estas razones, entre otras, se vuelve obligatorio hacer uso de los estándares y protocolos; los estándares permiten aumentar la calidad del producto (es decir, cumplen con los requisitos de rendimiento y seguridad) y garantizan la interoperabilidad de varios componentes en un sistema [5].

Por lo anteriormente dicho, se vuelve necesario definir el término "correo electrónico" o e-mail, aunque para muchos expertos del tema es suficiente decir que no es más que una recreación del servicio de correo tradicional sobre la base de un medio de comunicación moderno y digital [7]. Y es que la mayoría de los e-mails en Internet se transfieren por el Protocolo de Transferencia Simple de Correo (SMTP) [8] y, para recuperarlos de los usuarios finales de almacenamiento, es necesario acudir a uno de los protocolos POP3 (Protocolo de Oficina, versión 3 [9]) o IMAP4 (Protocolo de Acceso a Correo de Internet versión 4 [10]). Ambos protocolos están basados en texto con roles distintos de clientes y servidores y son servicios de mensajería tradicional sobre la base de un medio moderno y digital de comunicación, pertenecientes a la familia de MHS (Sistemas de Manejo de Mensaje). Estos protocolos transportan los mensajes a través del MTS (Sistema de Transportación de Mensaje) –subsistema del MHS— y permiten la interacción con el usuario a través del UA (Agentes de Usuario) [7]. Una característica importante del SMTP es que cada servidor físico podría operar

tanto como cliente o como servidor SMTP; al ser un servidor, acepta un correo electrónico entrante y se convierte en un cliente para reenviarlo [11].

La Capa de Conexión Segura (SSL) denota el protocolo predominante de seguridad de comunicación de Internet, en particular para los servicios World Wide Web (WWW) relacionados con el comercio electrónico o la banca hogareña. La mayoría de los servidores web y navegadores soportan SSL como el estándar de facto para la comunicación segura entre el cliente y el servidor. El protocolo SSL crea conexiones de punto a punto que permiten el uso privado y sin interrupciones intercambiando mensajes entre partes fuertemente autenticadas [12].

Actualmente, el gran crecimiento del Internet en el mundo ha penetrado en poco más del 54% de la población mundial [13], permitiendo con ello transmitir información y facilitando la comunicación entre humanos, máquinas y objetos. Esto último es el objetivo del Internet de las Cosas (IoT, por sus siglas en inglés) que versa sobre la integración del Internet con sensores, identificadores de radio frecuencia u otros objetos, en sistemas embebidos.

Los objetos dentro del IoT tienen la capacidad de enviar y recibir información, permitiéndoles comunicarse con otros dispositivos o servicios en el Internet para cumplir ciertos objetivos. De manera concreta, el IoT se define como "Una red mundial de objetos interconectados con dirección única, basada en protocolos estándar de comunicación" [14]. De hecho, el vocablo "Internet de las cosas " se atribuye a The Auto-ID Labs [15], una red mundial de laboratorios de investigación académica en el campo de la RFID (Radio Frecuency Identification) en red y las tecnologías de detección emergente.

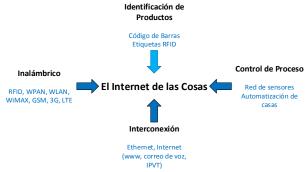


Fig. 1. Interacción de distintos componentes con el IoT [16].

En la Fig. 1, el IoT hace uso de productos de identificación, objetos inalámbricos, interconectividad y procesos de control, por tal motivo se requiere la coordinación de una entidad de mayor jerarquía que supervise el desarrollo de estas tecnologías. El uso y su consecuente importancia del IoT es de tal magnitud que, tan sólo en 2010, el número de objetos conectados en el Internet sobrepasó a la población humana [17]. Sin embargo, esto no quiere decir que el IoT sea perfecto pues aún tiene retos por afrontar. Estos se pueden resumir en los siguientes nueve puntos [18]:

217

- Disponibilidad de internet gratuito en todas partes.
- Temas de seguridad.

- Desarrollo de sistemas de detección inteligentes de bajo costo.
- Uso de distintos tipos de energía.
- Capacidad computacional.
- Escalabilidad.
- Tolerancia a fallos.
- Consumo de energía.
- Aceptabilidad entre la sociedad.

Además, la fragmentación de las tecnologías de la comunicación se ha considerado como uno de los principales obstáculos que entorpecen la visión de IoT. Por tanto y como requisito necesario, se tiene que el IoT debe basarse en tecnologías de comunicación estándar abiertas que permitan la interoperabilidad y las capacidades plug-and-play [19]. Esto ha motivado distintas propuestas y aplicaciones [1, 20, 21, 22] que toman ventaja de las redes sociales o de sus APIs (Interfaz de Programación de Aplicaciones) de web abierta con la intención de compartir Smart-things entre usuarios de confianza o bien, de aquellos usuarios que han aportado mecanismos interesantes y útiles destinados a integrar redes aisladas basadas en IoT en Internet a través de servicios en la nube.

El presente artículo se estructura de la siguiente forma: se aborda el estado del arte en la Sección presente, describiendo el internet de las cosas y de los correos electrónicos. La Sección 2 describe las necesidades que motivaron la creación de Ernest, los algoritmos utilizados y la arquitectura de la aplicación y, finalmente, se muestran los resultados en la Sección 3.

2. Desarrollo

Ernest es una aplicación que permite el control del encendido o apagado de las cosas —televisiones, tostadores, licuadoras, lamparas, etc.—conectadas a sus tomacorrientes haciendo uso de los correos electrónicos; al estar fundamentado en e-mails, la privacidad se mantiene, sin la intervención de terceros para acceder a la nube. Esto es posible gracias a su circuito y sus algoritmos para envío y lectura de correos electrónicos.

Actualmente, se tiene la segunda versión de Ernest, las principales diferencias entre la primera versión y la segunda, es el uso del módulo WI-FI que se emplea —un RN 131-G para la primera y un RN 1810 para la segunda—y la implementación del cifrado SSL, donde en la primera versión no se podían utilizar correos electrónicos de Gmail o Hotmail, entre otros. En este apartado se enfatizará la segunda versión por lo que se describe a continuación:

2.1. Arquitectura

Ernest cuenta con un elemento de Hardware. Dicha sección permite conocer qué enchufes están activos y cuáles no y, en caso de solicitarlo, cambiar el estado en que se encuentren. Como puede observarse en la Fig. 2, un microcontrolador ATMEGA328p tiene comunicación con un módulo RN 1810 el cuál otorga Internet al sistema embebido, también cuenta con una LCD donde se muestra el estado en el que se

encuentra el circuito, ya sea leyendo los correos, indicando si está o no conectado a Internet, o si listo en caso de haber realizado lo solicitado; y cuenta con su etapa de potencia para habilitar o deshabilitar los tomacorrientes con el uso de MOC3041.

Para energizar a Ernest se utiliza una clavija común y se conecta a un tomacorriente; para proporcionar los 5 volts requeridos por ciertos componentes del circuito, se utiliza un EME05US05, que otorga una salida de 5 V a 1 A para una entrada de 85 a 264 VAC.

Para la conexión al WI-FI se utiliza el módulo RN1810 que otorga Internet al sistema embebido y que permite la encriptación de datos con cifrado SSL a través de hardware para que no circulen sin protección por la Web.

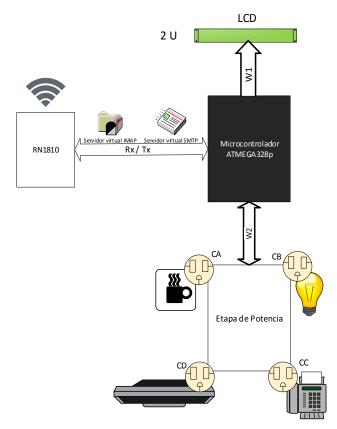


Fig. 2. Esquema del funcionamiento de Hardware de Ernest

Como medida de protección del circuito se utiliza un filtro de red, ver Fig. 3, para C.A. para eliminar parásitos en la red en modo común y diferencial; con ello se blindan los aparatos que se conecten a Ernest y a la fuente EME05US05.

Antes de pasar el circuito a una PCB, primero se prueba su funcionamiento en protoboard (Fig. 4ª). Una vez que se comprueba, se procede a pasar a la PCB, en la Fig. 4b se puede observar el circuito de Ernest en 3D donde ya se encuentra el microcontrolador, el módulo WI-FI, la etapa de potencia y el filtro, todo en el PCB.

José Francisco Rodríguez Arellano, Emmanuel Dávila Delgado, Mario Alberto Ruíz Durán, et al.

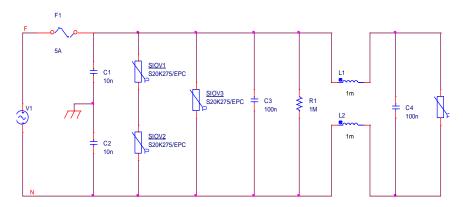


Fig. 3. Filtro de C.A. para Ernest.

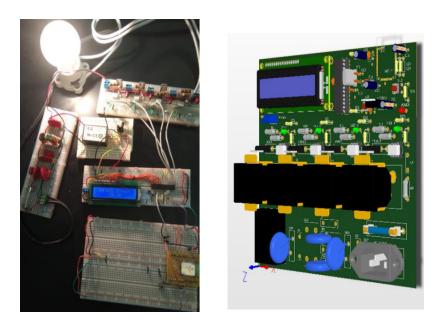


Fig. 4. a. Prueba del circuito en protoboard. b. Circuito de Ernest en 3D.

2.2. Algoritmos

Para la aplicación de Ernest se crearon tres algoritmos. Para mayor comprensión de su funcionamiento, la Fig. 6. muestra cómo Ernest hace uso del correo electrónico para la recepción de las órdenes del usuario.

Ingresar nombre de la red Ingresar a la red v clave Ingresar correo electrónico Abrir bandeja y contraseña que usará de entrada **Ernest** Filtrar los Ingresar correo electrónico estinatarios de del usuario los correos Cerrar bandeja Existe un correo nuevo? de entrada Envía el estado ¿El correo es una orden? de Ernest Ejecutar la instrucción

Ernest: sistema embebido para control de casas inteligentes mediante correos electrónicos...

Fig. 6. Funcionamiento general de la aplicación Ernest.

Conexión al módulo Wi-Fi. Inicialmente, se comienza agregando la conectividad WI-FI al sistema con el uso del módulo Wi-Fi RN1810, utilizando un microcontrolador ATMEGA32 y generando el primer algoritmo para conectarse a la red que se desee. Tal algoritmo se esquematiza en la Fig. 7, donde se exhiben la secuencialidad a seguir para la correcta comunicación y conexión a Internet.



Fig. 7. Algoritmo para conectar el módulo a Internet.

Lectura de correos electrónicos. Una vez realizada la conectividad a Internet, y dado que se imprime en la LCD "Already joined network", se prosigue con el acceso al correo electrónico, haciendo uso de los protocolos de los mecanismos de recuperación

ISSN 1870-4069

IMAP y accediendo a los mensajes de correo electrónico alojados en el servidor de éste. Tal como establecen los protocolos, se abre el IMAP por el puerto 993 del TCP (Protocolo de Control de Transmisión), mediante el comando "open IMAP 993"; una vez establecida la conexión, se saluda al servidor y se procede con la autentificación bajo el comando "logi".

Confirmado el acceso, se procede a la lectura de la bandeja de entrada. Para la lectura de los correos electrónicos nuevos, se hace uso del comando SELECT INBOX, y se utiliza después el comando FETCH seguido del número de correo por leer, tal y como estén ordenados en la bandeja. Si el correo contiene la información "prende" o "apaga", se sabe que es una instrucción que debe ejecutarse y se realiza la búsqueda de si el contacto desea "prender" o "apagar", y en caso de que no sea una instrucción, se interpreta como el estado en que se encuentra Ernest, propiciando el cambio del algoritmo de escritura de mensaje y cerrando el protocolo IMAP con el comando LOGOUT y, de la misma forma, lo cierra en caso de no haber nuevos correos sin leer.

Escritura de correos electrónicos. En la escritura del mensaje, se realiza un cambio de protocolo para pasar del IMAP al SMTP; para este protocolo se utiliza el puerto 465 TCP, iniciando con el comando "open smtp.gmail.com 465" donde gmail.com es el servidor de mensajería usado para Ernest. Para inicializar la sesión se escribe el comando "EHLO¹ Ernest", seguido de "AUTH LOGIN" y después se envía el correo electrónico y la contraseña del usuario codificado en base64. Realizado esto, se pueden mandar mensajes al HOST a través del comando MAIL FROM, el cual debe contener la dirección del remitente del sobre y se espera la respuesta. Para referenciar a quién se desea enviar el sobre, se utiliza el comando RCPT seguido de la(s) dirección(es) a la(s) que se enviará el correo. Una vez realizado esto, se puede enviar el (los) correo(s) electrónico(s), con la siguiente estructura:

DATA
From:
Subject: Ernest App
To:

Y se remite el estado de los contactos, en los cuáles se dice si los contactos en CA, CB, CC y CD están activos o no, enviándose la información de uno por uno. Una vez que se envió el mensaje, se procede a salir de este protocolo por medio del comando QUIT.

3. Resultados

En la Fig. 8 se aprecia la primera versión de Ernest donde se puede observar en funcionamiento, al estar leyendo un e-mail. En la Tabla 1 se muestra una comparativa entre ambas versiones.

¹ EHLO = (hello extendido) la respuesta del servidor a EHLO da el nombre del servidor en la primera línea.

Ernest: sistema embebido para control de casas inteligentes mediante correos electrónicos...



Fig. 8. Circuito de la primera versión de Ernest

Tabla. 1. Comparativa entre las dos versiones de Ernest.

	MCU	Transceptor	Encriptación	IPv6
Ernest 1ra versión	Atmega328p	Wifly rn-131G	NO	NO
Ernest 2da versión	Atmega328p	Wifly rn-1810	SSL	SI

Con la segunda versión de Ernest se desarrollaron pruebas de fidelidad. Antes de enviar algún correo se verificó que Ernest estuviera estableciendo una conexión exitosa con la red WI-FI y con el servidor IMAP. La prueba consistió en dejar conectada la placa por una hora revisando la transmisión a través del puerto serie, véase Fig. Cuando el internet se encontraba saturado o lento, o bien, si el módulo WI-FI tenía una señal muy baja -observado a través de dispositivos cercanos al área de trabajo de Ernest- se obtuvieron fallas con la conexión del servidor IMAP.

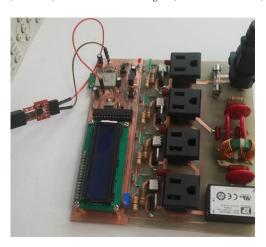


Fig. 9. Conexión de prueba del microcontrolador con el módulo Wifi.

Para comprobar la eficacia de dichos algoritmos, se realizaron dos pruebas en las cuales durante una hora se enviaron primero 50 e-mails y después en una segunda prueba se mandaron 100 e-mails y se comprobó que estos fuesen recibidos por Ernest, y de la misma forma, que fuesen ejecutado por este. Durante la primera prueba todos los correos enviados fueron recibidos. En la segunda prueba se enviaron 100 correos, en el lapso de esta hora se volvió a observar lo que ocurría en el puerto serie. Los primeros 52 correos que leyó se ejecutaron correctamente, el correo número 53 no lo ejecutó debido a que el Wifly obtuvo una conexión fallida con el servidor IMAP. Las posibles causas de esto son las siguientes:

- (1) Se perdió la conexión WI-FI.
- (2) El servidor de IMAP rechazó la solicitud.



Fig. 10. Ernest ejecutando comandos.

Una vez que se reinició el módulo WI-FI, Ernest continuó su trabajo de lectura de correos normalmente, del 54 al 100. En la Tabla 2 se muestra la efectividad de Ernest

en ambas pruebas con solo uno correo no recibido durante la segunda prueba. En la Fig. 10 se muestra a Ernest durante las pruebas realizadas en la recepción de correos.

	Correos enviados	Correos exitosos	Porcentaje
Prueba 1	50	50	100.00%
Prueba 2	100	99	99.00%

Tabla. 2. Efectividad de Ernest para recibir correos.

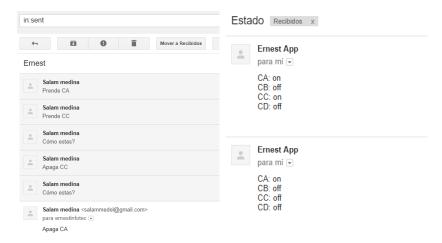


Fig. 11. a. E-mails enviados a Ernest b. E-mails recibidos de Ernest.

En la Fig. 11a puede observarse un ejemplo en el que se enviaron seis e-mails a Ernest de los cuáles cuatro fueron órdenes para su ejecución y los otros dos fueron preguntas para conocer el estado en que se encuentra. Las respuestas recibidas se observan en la Fig. 11b, donde se confirma que se ha realizado exitosamente el encendido o el apagado de un contacto.

4. Conclusiones

A largo del presente trabajo se presentó un sistema embebido para el control de casas inteligentes mediante correos electrónicos con cifrado SSL. Así como su metodología de desarrollo acompañada de una serie de pruebas para validar el funcionamiento del sistema. Ernest representa una opción tangible para aplicaciones o bien usuarios sensibles a compartir su información personal en una nube. Durante la experimentación, se demostró que Ernest tiene una certeza del 99 % de eficacia, lo cual demostró ser un prototipo funcional y confiable. Sin embargo, para poder tener una mayor certeza con respecto a las posibles fallas aún deben de realizarse más pruebas en distintos ambientes para poder obtener datos ya sea de cómo influye el tráfico de la red en la conectividad del módulo. Además, durante los experimentos realizados se observó

un gran desempeño por parte de los algoritmos para ejecutar las instrucciones, obteniendo más dificultades en el apartado de conexión al módulo Wi-Fi

Ernest también presenta áreas de oportunidad para ser un sistema más robusto. Actualmente, se trabaja en el desarrollo de la próxima versión de Ernest, donde se empieza a trabajar con el procesamiento de textos con la finalidad que Ernest se vuelva más intuitivo a las demandas de sus usuarios. También se está trabajando en un proceso handshake para evitar que ese bajo porcentaje de correos no ejecutados se corrija cuando al no recibir respuesta del sistema el mismo correo sea retransmitido. Por último, se observó que se debe cambiar de microcontrolador a causa de la poca memoria RAM que contiene el ATMEGA328p, por ello se está pasando a un STM32F42 por su capacidad de 192 Kbyte de la misma, permitiendo cargar con buena calidad la página del servidor del correo electrónico, además de que se le realizará su respectiva placa, pero ahora de montaje superficial.

Referencias

- 1. SmartThings-Make your world smarter, http://smartthings.com/, last accessed 2018/03/13
- Atzori, L., Iera, A., Morabito, G.: The Internet of Things: A survey. Comput. Netw., pp. 2787–2805 (2010)
- Alhakbani, N., Hassan, M.M., Hossain, M.A., Alnuem, M.: A Framework of Adaptive Interaction Support in Cloud-Based Internet of Things (IoT) Environment. In: Fortino G., Di Fatta G., Li W., Ochoa S., Cuzzocrea A., Pathan M. (eds) Internet and Distributed Computing Systems. Lecture Notes in Computer Science, vol. 8729, pp. 136–146. Springer, Cham (2014)
- 4. Joyanes-Aguilar, L.: Computación en la Nube. 2nd edn. Alfaomega, Panamá (2012)
- Guerrero-Ibáñez, J. A., Flores-Cortés, C., Zeadally, S.: Vehicular Adhoc Networks (VANETs): Architecture, Protocols and Applications. In: Next-Generation Wireless Technologies, pp. 49–70. Springer, London (2013)
- 6. Stankovic, J. A.: Research Directions for the Internet of Things. In: Life Fellow, Vol. 1, No. 1, pp. 3–9. IEEE Internet of Things Journal, Charlottesville (2014)
- Meinel, C., Sack, H.: Application Layer and Internet Applications. In: Internetworking, pp. 665–830. Springer, Berlin (2013)
- 8. Klensin, J.: Simple Mail Transfer Protocol, IETF RFC 5321 (2008)
- 9. Myers, J., Rosem, M.: Post Office Protocol Version 3, IETF RFC 1939 (1996)
- 10. Crispin, M.: Internet Message Access Protocol Version 4rev1, IETF RFC 3501 (2003)
- 11. Pakulin, N., Tugaenko, A.: Specification Based Conformance Testing for E-mail Protocols, pp. 371–381. Springer, Berlin (2010)
- 12. Tilborg H. C. A., Jajodia S.: Encyclopedia of Cryptography and Security. 1st edn. Springer, Boston, MA. (2011)
- Internet World Status Usage and Population Statistic http://www.internetworldstats.com/ stats.htm (2018)
- INFSO D.4 Networked Enterprise & RFID INFSO G.2 Micro & Nanosystems, in: Cooperation with the Working Group RFID of the ETP EPOSS, Internet of Things in 2020, Roadmap for the Future, Version 1.1 (2008)
- 15. Auto-Id Labs (2018)
- 16. Mainwaring, K., Srivastava, L.: The Internet of Things. In: H. Chaouchi (Ed.), ISTE edition (2010)

- 17. Evans, D.: The internet of things: How the next evolution of the internet is changing everything, CISCO White Paper (2011)
- 18. Mukhopadhyay, S. C., Suryadevara, N. K.: Internet of Things: Challenge and Opportunities. volume 9. In: Springer (2014)
- 19. Colitti, W., Long, N. T., De Caro, N., Steenhaut, K.: Embedded Web Technologies for the Internet of Things, Chandra Mukhopadhyay. In: Springer (2014)
- Guinard, D., Fischer, M., Trifa, V.: Sharing using social networks in a composable Web of Things. In: 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 702–707 (2010)
- 21. Cosm–Internet of Things Platform Connecting Devices and Applications for Real–Time Control and Data Storage (2013) (2018)
- 22. Paraimpu–The Web of Things is more than Things in the Web (2018)

Reducción de dimensionalidad en microarreglos: GN y GA

Flor Alejandra Romero-Montiel, Katya Rodríguez-Vázquez

UNAM, Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas, México f.alejandra.r.m@gmail.com, katya.rodriguez@iimas.unam.mx

Resumen. Los microarreglos de DNA surgen de la necesidad de analizar la cantidad de información procedente de los grandes proyectos de secuenciación de genomas [1]. Permiten investigar el nivel de expresión de genes en una muestra. Usando esta información es posible diagnosticar y predecir enfermedades, por lo tanto, la tarea de clasificación de microarreglos de DNA es importante en bioinformática [2]. En este trabajo se mostrará como un neurón generalizado puede usarse en la tarea de clasificación de microarreglos. En la metodología propuesta se inicia por seleccionar un conjunto de genes usando un algoritmo genético, después el neurón generalizado es entrenado con un algoritmo genético. Finalmente la precisión de la metodología es probada clasificando tres bases de datos de microarreglos de DNA: $Leukemia\ benchmarck\ ALL-AML$, $Colon\ Tumor\ y\ Prostate\ cancer$.

Palabras clave: microarreglos, algoritmos géneticos, clasificación, redes neuronales, reconocimiento de patrones.

Reduction of Dimensionality in Microarrays: GN and GA

Abstract. DNA microarrays arise from the need to analyze the amount of information coming from large genome sequencing projects [1]. They allow to investigate the level of gene expression in a sample. Using this information it is possible to diagnose and predict diseases, therefore, the task of classifying DNA microarrays is important in bioinformatics [2]. This paper we will show how a generalized neuron can be used in the task of microarray classification. In the proposed methodology begins by selecting a set of genes using a genetic algorithm, then the generalized neuron is trained with a genetic algorithm. Finally, the accuracy of the methodology is tested by classifying three databases of DNA microarrays: Leukemia benchmarck ALL – AML, Colon Tumor and Prostate cancer

Keywords: microarrays, genetic algorithms, classification, neural networks, pattern recognition.

1. Introducción

Los microarreglos de DNA son utilizados para la cuantificación masiva de la expresión de genes. Este análisis permite diagnosticar enfermedades, identificar diferentes tumores, seleccionar para un paciente específico el mejor tratamiento para resistir una enfermedad [3]. Para obtener esta información se han aplicado algunas líneas de investigación de la Inteligencia Arficial (IA). Una rama de IA, llamada clasificación de patrones, consiste en la identificación de diferentes clases o grupos asociados con una enfermedad particular, por ejemplo tipos de cáncer o detección en un paciente. El microarreglo de DNA tiene una enorme cantidad de genes para analizar (del orden de los miles) y un pequeño número de muestras, esto es una desventaja cuando se utiliza una tarea de aprendizaje y reconocimiento, por lo que es un reto para los métodos de clasificación actuales [4].

Las redes neuronales artificiales (ANN por sus siglas en inglés) son famosos modelos computacionales usados para clasificación y otras tareas [5]. En los trabajos encontrados en la literatura se puede notar que comparten una característica: los autores buscan reducir la dimensionalidad de los genes, por que muchos de ellos son irrelevantes y la consecuencia de conservarlos es costosa, aumenta el tiempo de cómputo, la alta complejidad y el bajo rendimiento en la clasificación o predicción de una enfermedad [6].

La red neuronal generalizada o neurón generalizado (GN) es otro tipo de ANN que fue desarrollado con el objetivo de reducir el diseño de una red neuronal, pues no tiene muchas conexiones, con un buen rendimiento en comparación con el ANN clásico, además de ser fácil de implementar.

Algunos trabajos aplicaron GN para resolver funciones de aproximación, para calcular estimaciones de densidad, predicción y problemas de clasificación [7,8]. En este trabajo se analiza un problema de clasificación de microarreglos de DNA. La metodología consiste en reducir primero la dimensionalidad de los genes usando un algoritmo genético, luego clasificar los datos con un GN donde algunos parámetros son determinados mediante un algoritmo genético.

Este trabajo se divide en seis secciones: en la segunda se describe el concepto de neurón generalizado. A continuación se describe el algoritmo genético. Además la metodología propuesta se describe en la sección cuatro, seguido por los resultados experimentales en la sección cinco. Finalmente las conclusiones de este trabajo se encuentran en la sección seis.

2. Neurón generalizado

El neurón generalizado (GN) fue propuesto en [9], ha sido aplicado en problemas de clasificación y aproximación de funciones.

La estructura general del modelo son dos funciones de agregación (suma y producto) y dos funciones de transferencia (sigmoidal y gaussiana). La función sigmoidal característica (f_1) se usa con la función de suma \sum_1 , mientras que la función gaussiana (f_2) se usa con la función producto \prod . La salida de la parte

 \sum_{1} con la función sigmoidal de activación es :

$$O_{\sum} = f_1(S_{net}) = \frac{1}{1 + exp(-\lambda_s * S_{net})},\tag{1}$$

donde:

$$S_{net} = \sum W_{\sum i} X_i + X_{0\sum},$$

y $\lambda_s,\,X_{O\sum}$ son la ganancia y el sesgo de la parte \sum respectivamente. La salida de la parte \prod con la función de activación gaussiana para f_2 es:

$$O_{\prod} = f_2(P_{net}) = exp(-\lambda_p * P_{net}^2), \tag{2}$$

donde

$$P_{net} = \prod W_{\prod i} X_i * X_{O \prod},$$

y λ_p , $X_{O\prod}$ son la ganancia y el sesgo de la parte \prod , respectivamente. La salida final O_{pk} del neurón es una función de las dos salidas O_{\sum} y O_{\prod} con los pesos W y 1-W respectivamente y se puede escribir como:

$$O_{GN} = W * O_{\sum} + (1 - W) * O_{\prod}.$$
 (3)

Para problemas de múltiples salidas, diversos modelos de neurón generalizado en paralelo son requeridos. El número de pesos en el caso del neurón generalizado es el doble del número de entradas más dos pesos del sesgo, más dos pesos de la ganancia más un peso que corresponde a W (porcentaje de contribución de cada una de las partes de la estructura del neurón: suma y producto). Esto es mucho menos comparado con el número de pesos en una red multicapa [10]. Al reducir el número de pesos desconocidos el tiempo de entrenamiento se reduce.

3. Algoritmo genético

Los algoritmos genéticos (GA) son estrategias de búsqueda estocástica basados en el mecanismo de selección natural, imitando a la evolución biológica como estrategia para resolver problemas [11]. Los GA trabajan sobre un conjunto de potenciales soluciones, llamado población. Esta población está compuesta de una serie de soluciones llamadas individuos y un individuo está conformado por una serie de posiciones que representan cada una de las variables involucradas en los procesos de optimización y que son llamados cromosomas. Estos cromosomas están compuestos por una cadena de símbolos que en muchos casos está presentada en números binarios, aunque también es posible usar codificación hexadecimal, octal, real, etc.

En un GA cada individuo está definido como una estructura de datos que representa una posible solución del espacio de búsqueda del problema. Las estrategias de evolución trabajan sobre los individuos, que representan las soluciones del problema, por lo que estos evolucionan a través de generaciones. Dentro de

la población cada individuo es diferenciado de acuerdo con su valor de aptitud o fitness, que es obtenido usando algunas medidas de acuerdo con el problema a resolver. Para la obtención de las próximas generaciones se crean nuevos individuos, llamados hijos, utilizando dos estrategias de evolución básicas como son el operador de cruce y el de mutación.

En Algoritmo 1.1 se muestra el pseudocódigo del algoritmo genético simple.

Algorithm 1.1 Algoritmo Genético Simple.

```
\begin{aligned} \mathbf{S} &\leftarrow \mathbf{Generar} \text{ una población inicial } S \\ \mathbf{while no se alcance la condición de parada do} \\ &= \mathbf{valuar} \text{ cada individuo de la población } S \\ \mathbf{for } i &= 1 \text{ hasta } \frac{tamaño \quad (S)}{2} \mathbf{do} \\ &= \mathbf{Seleccionar} \text{ dos individuos de la generación anterior.} \\ &= \mathbf{Cruzar} \text{ con cierta probabilidad los dos individuos obteniendo dos descendientes.} \\ &= \mathbf{Mutación de individuos.} \\ &= \mathbf{Insertar los dos descendientes mutados en la nueva generación.} \\ &= \mathbf{end for} \\ &= \mathbf{end while} \\ &= \mathbf{return la mejor solución encontrada.} \end{aligned}
```

4. Metodología para clasificar microarreglos de DNA

La metodología usada en este trabajo para desempeñar una tarea de clasificación binaria de microarreglos de DNA se divide en dos etapas:

- La primera dedicada a seleccionar el conjunto de genes que mejor describen el microarreglo de DNA.
- La segunda enfocada a entrenar un neurón generalizado para mejorar la precisión de la tarea de clasificación.

y la representación esquemática se puede observar en la figura 1. El primer paso propone una reducción de dimensionalidad del microarreglo de DNA, aminorando el número de características, con las cuales será entrenado el neurón generalizado. El segundo paso consiste en utilizar esta información para entrenar un neurón generalizado y realizar la clasificación. Estos dos pasos son repetidos hasta que el número máximo de iteraciones es alcanzado. Una descripción más detallada de las etapas se describen a continuación:

4.1. Reducción de dimensionalidad

De acuerdo con [12], la selección del conjunto con los mejores genes puede ser definido en términos de un problema de optimización. En este trabajo se usó un algoritmo genético, (algoritmo genético 2) para explorar el espacio de soluciones, los individuos son subconjuntos de genes del microarreglo de DNA de longitud

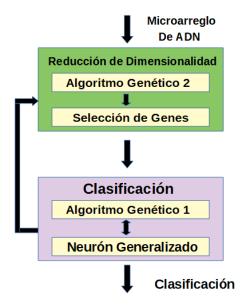


Fig. 1. Esquema de la metodología propuesta.

fija. La función fitness usada, es el mínimo número de elemento mal etiquetados al usar esos genes para entrenar un GN, es decir, con la base propuesta se entrenan distintos modelos de GN, cada uno de ellos con un resultado de clasificación distinto, tomando como valor de fitness la clasificación con menos errores. Un ejemplo de codificación es:

$$[19, 58, 7, 325, 205].$$
 (4)

Las entradas se encuentran entre 0 y el número de características que tiene la base original menos uno, indican los índices de las columnas que serán tomadas para formar una nueva base de datos. La cruza usada es, dados los individuos $Ind_1 = [I_{11}, I_{12}, ..., I_{1n}]$ y $Ind_2 = [I_{21}, I_{22}, ..., I_{2n}]$, se generan dos descendientes, $H_k = [h_{k1}, h_{k2}, ..., h_{kn}], k = 1, 2$ donde:

$$H_1 = Ind_1 + \alpha * (Ind_2 - Ind_1), \tag{5}$$

$$H_2 = Ind_2 + \alpha * (Ind_1 - Ind_2), \tag{6}$$

con $\alpha \in [-0.25, 1.25]$. La mutación es un operador unario, dado un individuo $Ind_1 = [I_{11}, I_{12}, ..., I_{1n}]$ este tiene una probabilidad de mutar del 10 % (en la mayoría de los casos es baja), si el individuo muta, cada entrada I_{1k} tendrá la probabilidad $\frac{1}{n}$ de mutar, es decir:

$$Im_{1k} = I_{1k} + (\beta - 0.5) * (p_2 - p_1) * 0.1,$$
 (7)

donde $\beta \in (0,1)$, $p_2=4$ y $p_1=-4$ después de aplicar estos operadores, se aplica la función piso a todos los individuos, comprobando en cada individuo que las

entradas son todas distintas entre sí, de no ser el caso se reemplazará con un aleatorio.

4.2. Clasificación usando un GN

Una vez propuesto el conjunto de características, estos genes formarán una nueva base de datos, que es particionada en dos conjuntos: entrenamiento y prueba. El conjunto de entrenamiento posee el 70 % y el conjunto de prueba el 30 % restante. Esta partición se realiza de forma aleatoria para asegurar que los conjuntos contengan elementos con ambas etiquetas. Después el GN es entrenado con un algoritmo genético, para términos prácticos llamado algoritmo genético 1, con codificación real. Las soluciones (individuos), generados con el algoritmo genético 1, codifican la estructura del GN en términos de los pesos sinápticos (W_{Σ}, W_{Π}, W), sesgo y parámetros de la función de activación (λ) para cada tipo de neurona ($\Sigma y \Pi$). Un ejemplo de codificación es :

$$[0.98, 2.12, -3.45, -0.34, 1.52, 0.872, -3.369, 2.548, -0.125, 1.417, -2.834, 0.723, -1.396, 2.196, -3.592]$$
(8)

Las entradas se encuentran en el intervalo (-4,4), si el número de características en la base es 5, la longitud de los individuos es de (2*5) + 5. La cruza y mutación son las mencionadas anteriormente.

La función escalón es aplicada a la salida del GN para determinar la clase a la que pertenece. En la figura 2 se muestra la gráfica de dicha función. La función fitness está basada en el número de elementos mal etiquetados, se obtiene la matriz de confusión comparando las etiquetas reales con las etiquetas propuestas, el número de falsos negativos y falsos positivos es el valor de la función fitness que se busca minimizar. Una vez entrenado el GN, se procede a evaluar la capacidad de generalización usando el conjunto de prueba, utilizando las métricas de precisión, recall y f1-score.

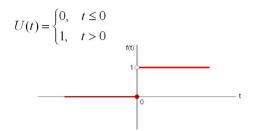


Fig. 2. Gráfica de la función escalón de Heaviside.

5. Resultados experimentales

En esta sección, se analizarán los resultados obtenidos con la metodología propuesta para determinar la precisión, usando tres bases de microarreglos distintas. En todos los casos, para seleccionar el mejor conjunto de genes usando el algoritmo genético 2, con codificación entera, se definieron los siguientes parámetros: tamaño de la población = 30 y máximo número de ciclos= 30, selección por torneo, probabilidad de cruza= $100\,\%$, probabilidad de mutación del individuo = $10\,\%$, probabilidad de mutación de cada gen= $1/(número\ de\ genes)$ y elitismo. En los casos donde la base de datos se encuentra dividida en datos de entrenamiento y prueba, se reagrupa y se realiza una partición aleatoria, $70\,\%$ para entrenamiento y $30\,\%$ para prueba.

lacktriangle La metodología fue aplicada para clasificar dos tipos de cáncer: la leucemia linfocítica aguda y la leucemia mieloide aguda, en la base de datos $Leukemia\ ALL-AML$.

Leukemia benchmark ALL-AML contiene las medidas correspondientes a muestras de ALL y AML de médula ósea y sangre periférica. Originalmente se compone de 38 muestras para entrenamiento (27 ALL y 11 AML) y 34 muestras para prueba (20 ALL y 14 AML) donde cada muestra contiene información de 7129 expresiones génicas, al realizar la partición se tienen 50 muestras en el conjunto de entrenamiento y 22 en el conjunto de prueba. En la tabla 1 se muestran algunos de los resultados obtenidos.

Tabla 1. Ent.= Entrenamiento, Pba.= Prueba.

Genes	Precisión		Recall		f1-score	
	Ent.	Pba.	Ent.	Pba.	Ent.	Pba.
						0.94
15	0.91	0.96	0.86	0.91	0.88	0.94
				0.89		
				0.95		
30	0.80	0.93	0.71	0.95	0.77	0.95

Después de concluir los experimentos, se puede notar que el mejor resultado obtenido fue al usar 15 genes, obteniendo un precisión del 96 % en el conjunto de prueba.

■ La base de datos *colon Tumor* contiene 62 muestras, de las cuales 22 son positivas para Tumor en colon y 40 negativas. Al realizar la partición se tienen 43 muestras para el entrenamiento y 19 para prueba. Cada muestra contiene 2000 genes. En la tabla 2 se muestran algunos de los resultados obtenidos.

Tabla 2. Ent.= Entrenamiento, Pba.= Prueba.

Genes	Precisión		Recall		f1-score	
	Ent.	Pba.	Ent.	Pba.	Ent.	Pba.
10	0.82	0.95	0.81	0.88	0.81	0.91
15	0.82	0.95	0.87	0.85	0.81	0.91
20	0.83	1.0	0.87	1.0	0.85	1.0
30	0.94	0.93	0.96	0.86	0.95	0.87

Después de concluir los experimentos, se observa que los mejores resultados se obtuvieron al usar 20 genes, obteniendo una precisión de $83\,\%$ en el conjunto de entrenamiento y $100\,\%$ en el conjunto de prueba.

■ La base *Prostate cancer* se compone de 102 muestras para entrenamiento (52 con tumor de próstata y 50 con no-tumor de próstata "normal") El conjunto de prueba contiene 25 muestras con tumor de próstata y 9 normal, cada muestra contiene 12,600 genes. Al realizar la partición se tienen 95 muestras para el entrenamiento y 41 para prueba. En la tabla 3 se muestran algunos de los resultados obtenidos.

Tabla 3. Ent.= Entrenamiento, Pba.= Prueba.

Genes	Precisión		Recall		f1-score	
	Ent.	Pba.	Ent.	Pba.	Ent.	Pba.
10	0.74	0.78	0.75	0.78	0.74	0.78
				0.85		
				0.89		
				0.90		
50	0.84	0.89	0.83	0.89	0.83	0.89

Después de concluir los experimentos, se observa que los mejores resultados se obtuvieron al usar 40 genes, obteniendo una precisión de 94 % en el conjunto de entrenamiento y 90 % en el conjunto de prueba.

6. Conclusiones

Los diversos experimentos permiten determinar el comportamiento de la metodología propuesta en la clasificación de microarreglos de DNA. Durante la primera etapa, se aplicó con éxito una reducción de dimensionalidad sobre los conjuntos de datos Leukemia benchmarck ALL-AML, Colon Tumor y Prostate cancer para seleccionar el conjunto de genes que mejor describe una enfermedad en particular, utilizando un algoritmo genético. El problema de reducción de dimensionalidad puede tratarse como un problema de optimización, debido a que la disminución dimensional de un microarreglo de DNA puede verse como un

problema combinatorio que trata de encontrar entre millones de genes los más relevantes.

Los resultados obtenidos utilizaron en algunos casos menos del uno por ciento de los genes para realizar una tarea de detección o clasificación.

En la segunda etapa, se evaluó el desempeño del GN, los resultados obtenidos mostraron que todo el conjunto de datos se resolvió con una buena precisión, por lo que el algoritmo genético es una buena técnica para entrenar un GN. Estos GN fueron entrenados usando el conjunto de genes propuestos en la primera etapa.

Finalmente, se puede concluir que el GN entrenado con la metodología propuesta es capaz de detectar, predecir y clasificar una enfermedad con una precisión aceptable.

Trabajos futuros incluirán la comparación del desempeño del neurón generalizado con otras arquitecturas de ANN.

Agradecimientos. Se agradece al posgrado de Ciencia e Ingeniería de la Computación de la UNAM, así como al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo brindado.

Referencias

- López, M., Mayorquin, P., Vega, M.: Aplicación de los microarrays y biochips en salud humana. Madrid: Fundación Española para el Desarrollo de la Investigación en Genómica y Proteomica (2005)
- Garro, B. A., Rodríguez, K., Vazquez, R.: Generalized neurons and its application in DNA microarray classification. In: Evolutionary Computation (CEC), 2016 IEEE Congress on, pp. 3110–3115, IEEE (2016)
- 3. Illana-Rico, E.: Análisis bioinformático de datos de expresión genética obtenidos mediante tecnología de microarrays. Jaén: Universidad de Jaén (2018)
- Cano Gutiérrez, C.: Extracción de conocimiento de microarrays y literatura biomédica para el estudio de la regulación genética. Tesis doctoral], Granada: Editorial de la Universidad de Granada (2010)
- 5. Huynh, Hieu Trung, Kim, Jung-Ja, Won, Yonggwan: Classification study on DNA micro array with feed forward neural network trained by singular value decomposition. International Journal of Bio-Science and Bio-Technology, Vol. 1, pp. 17–24 (2009)
- Plous, C.: Microarreglos de ADN y sus aplicaciones en investigaciones biomédicas. Revista CENIC. Ciencias Biológicas, Núm. 2, pp. 132–135 (2007)
- Rizwan, M., Jamil, M., Kothari, D.P.: Generalized neural network approach for global solar energy estimation in India. IEEE Transactions on Sustainable Energy, Vol. 3, pp. 576–584, IEEE (2012)
- 8. Kiran, R., Jetti, S. R., Venayagamoorthy, G. K.: Online training of a generalized neuron with particle swarm optimization. In: Neural Networks (IJCNN'06), International Joint Conference on, pp. 5088–5095, IEEE (2006)
- Kulkarni, R., Venayagamoorthy, G. K.: Generalized neuron: Feedforward and recurrent architectures. Neural Networks, Vol. 22, pp. 1011–1017 (2009)

- Peterson, L. E., Ozen, M., Erdem, H., Amini, A., Gomez, L., Nelson, C., Ittmann, M.: Artificial neural network analysis of DNA microarray-based prostate cancer recurrence. In: Computational Intelligence in Bioinformatics and Computational Biology, (CIBCB'05), Proceedings of the 2005 IEEE Symposium on, pp. 1–8, IEEE (2005)
- 11. Arranz de la Peña, J., Parra Truyol, A.: Algoritmos genéticos. Vol. 20, pp. 06–07 (2007)
- 12. Garro, B., Rodríguez, K., Vázquez, R.: Classification of DNA microarrays using artificial neural networks and ABC algorithm. Applied Soft Computing, Vol. 38, pp. 548–560 (2016)

Modelo de datos para el monitoreo de variables agroclimatológicas de huertos urbanos utilizando internet de las cosas y cómputo en la nube

Omar Jehovani López Orozco¹, Miguel González Mendoza², Juan Manuel Olvera Santoyo¹, Ariel Lucien García Gamboa²

Resumen. En huertos urbanos, así como en las formas comunes de producción, algunas milenarias como los campos de cultivo a cielo abierto, otras más recientes como los invernaderos, se ejecutan siguiendo métodos tradicionales transmitidos de generación en generación. A veces supervisado por un agrónomo, pero apenas respaldado por datos, ya que no se aprovechan herramientas de las Tecnologías de la Información y Comunicación disponibles en la actualidad. En su mayoría, las decisiones diarias de los agricultores se relacionan con el riego y el uso adecuado de pesticidas y fertilizantes. Una fertilización inadecuada causa efectos negativos en las plantas que eventualmente se traduce en pérdidas económicas y un impacto ambiental negativo al aumentar la huella ecológica. Es muy importante tener información de los cultivos, el agua, el suelo y el medio ambiente circundante. El presente trabajo propone un modelo de datos para el envío y almacenamiento de información en la nube FIWARE de variables agro-climatológicas en Huertos Urbanos utilizando dispositivos con conectividad a internet y acceso a una nube de datos.

Palabras clave: huertos urbanos, internet de las cosas, modelo de datos, FIWARE.

Data Model for Monitoring Agro-climatological Variables of Urban Gardens Using the Internet of Things and Cloud Computing

Abstract. In urban orchards, like in the traditional forms of agricultural production, some are millenary such as open-air cultivation fields, and in others more recent such as greenhouses, are executed following traditional techniques and methods transmitted from generation to generation. Sometimes supervised by an agronomist, but barely backed by data, since they do not take advantage

of tools of Information and Communication Technologies available today. Mostly farmers' daily decisions are related to irrigation and the proper use of pesticides and fertilizers. Inadequate fertilization causes negative effects on the plants that eventually results in economic losses and a negative environmental impact by increasing the ecological footprint. It is very important to have information from the crops, the water, the soil and the surrounding environment. This paper proposes a data model for sending and storing information in the FIWARE cloud of agro-climatological variables in Urban orchards using devices with Internet connectivity and access to a data cloud.

Keywords: urban orchards, data model, internet of things, FIWARE.

1. Introducción

1.1. Huertos urbanos

Los huertos urbanos o agricultura urbana es parte del plan estratégico de la seguridad alimentaria propuesto por la Organización de las Naciones Unidas para la Agricultura y la Alimentación (FAO) para referirse al cultivo de plantas en el interior y en los alrededores de las ciudades, denominado "La agricultura urbana y periurbana" que proporciona productos alimentarios de distintos tipos de cultivos (granos, raíces, hortalizas, hongos, frutas), así como productos no alimenticios (plantas aromáticas y medicinales, plantas ornamentales, productos de los árboles). La agricultura urbana y periurbana (AUP) incluye la silvicultura, para producir frutas y leña [8].

La característica de estos es que los puede realizar casi cualquier persona y con poco conocimiento de agricultura, lo mejor es que lo hace desde su hogar o espacios dentro o en la periferia de la ciudad sin necesidad de que la persona tenga que realizar grandes desplazamientos, así mismo se puede encontrar diferentes confecciones de material y adaptación de estos a los espacios. Los tipos de huertos clasificados de acuerdo con el espacio donde se ubican son: azoteas, balcones, plazas públicas, terrenos comunitarios, escuelas, jardines municipales, muros, entre otros. Aunado a que ocupan espacios dentro de las manchas urbanas, podemos destacar múltiples beneficios, entre ellos se encuentran [10]:

- Beneficos medioambientales. Se preocupan por reducir el uso de pesticidas, herbicidas y fertilizantes, se utiliza de forma racional el agua, se utilizan las compostas, se utilizan carcazas de refrigeradores y lavadoras además se reduce el efecto del calor en las zonas urbanas y reduce el uso de bolsas plásticas.
- Beneficios educativos. Se utiliza de manera que las personas reciban y promuevan la educación ambiental, aplicación de técnicas en ciencias naturalez, aplicación de resolución de problemas y practicas de tecnicas de cultivo y el intercambio de conocimiento generacional [2].
- Beneficios sociales y económicos. Ayuda a mejorar la salud y calidad de vida y lo más importante es que reduce los gastos familiares por frutas y verduras [4].

1.2. Aspectos a considerar al implementar un huerto urbano

Es importante la ubicación del huerto porque de ello dependerán las variables ambientales que influyen directamente en el desarrollo del producto, por supuesto el sustrato o tipo de suelo que contenga debe cumplir con las características indispensables que le permitan a la planta tener retención de agua y desarrollar una estabilidad del tallo , los contenedores pueden ser una alternativa para el cultivo en espacios confinados, la procedencia de la semilla , debemos considerar los puntos anteriores para que podamos descifrar los tiempos así mismo las frecuencias de riego y como último aspecto pero no menos importante será siempre considerar la utilización de macro y micro nutrientes esenciales para el desarrollo, reproducción y maduración del fruto.

1.3. Estadísticas

En los países en desarrollo del mundo el consumo diario de fruta y hortalizas es apenas del 20% al 50% de lo recomendado por la FAO y la Organización Mundial de la Salud (OMS). Para 2020, la proporción de la población urbana que vivirá en la pobreza llegará al 45%, o 1 400 millones de personas. Para entonces, el 85% de la población pobre de América Latina, y casi la mitad de la de África y Asia, se concentrará en las ciudades y los centros urbanos [11].

El acceso a alimentos nutritivos es una dimensión clave de la seguridad alimentaria. En África y Asia, las familias urbanas gastan hasta un 50% de su presupuesto para alimentos en productos preparados baratos, muchas veces carentes de las vitaminas y minerales esenciales para la salud [12].

Las familias urbanas pobres gastan hasta el 80% de sus ingresos en alimentos, lo que los vuelve muy vulnerables cuando los precios de los alimentos suben o sus ingresos disminuyen.

La agricultura urbana y periurbana está muy difundida En américa latina y el caribe donde se practica, por ejemplo, en un 40% de los hogares de Cuba y en el 20% de los de Guatemala y Santa Lucía [7].

Además, es Cuba la que promueve la HUP desde principios del decenio de 1990, este sector representa el 60% de la producción hortícola, y el consumo per cápita de los cubanos de fruta y hortalizas supera el mínimo recomendado por la FAO y la OMS, además, proporciona directamente unos 117,000 empleos.

En Ecuador, unos microhuertos de hortalizas en 54 centros para el desarrollo infantil alimentan a 2,500 niños y obtienen suficientes ganancias de sus ventas para ser autosustentables.

La HUP puede ser rentable incluso a muy pequeña escala. En Dakar, las mujeres mantenían para consumo de sus familias el 30% de las hortalizas de sus microhuertos, vendían los excedentes en los puestos de las familias y ganaban el equivalente a los salarios de los trabajadores [6].

Desafortunadamente la agricultura propiamente urbana es aún muy incipiente en la Ciudad de México, no está todavía extendida la idea de cultivar dentro de las zonas propiamente urbanas, y la alta densidad de edificación hace que los espacios verdes sean escasos o de poca extensión [7].

En cuanto a la tecnificación de cultivos, especialmente para el monitoreo y control de variables agroclimatológicas, es frecuente encontrar niveles más altos de adopción de tecnología en ambientes controlados como invernaderos. Lo anterior se debe a la necesidad de garantizar la inocuidad del producto, la frescura y las propiedades y calidad organolépticas. Además, los agricultores deben perseguir una alta productividad cuidando el costo de producción, los recursos naturales y el impacto ambiental.

La producción agrícola es un negocio multifactorial, y las dificultades asociadas para monitorear y controlar las muchas variables a lo largo de toda la cadena de suministro, desde la producción hasta el tenedor del consumidor, necesitan necesariamente sistemas de información y comunicación para obtener, registrar, analizar y entregar información útil y comprensible para cada parte interesada involucrada a través de las diferentes etapas. Las Tecnologías de la Información y la Comunicación, junto con Internet of Things, se pueden usar para monitorear aspectos importantes de la cadena de suministro de Agroalimentos y para gestionar de manera efectiva los procesos involucrados. La adopción de las TIC y la IoT no es sencilla, tiene muchos desafíos asociados.

2. Trabajos relacionados

2.1. Descripción de conceptos tecnológicos

Hay muchas definiciones de Internet de las Cosas conocidos también como dispositivos IoT, una de ellas es "IoT es un entorno de computación en red ambientado, invisible, global, construido a través de la comunicación continua de sensores inteligentes, cámaras, software, bases de datos y datos masivos. Se centra en un tejido de información que abarca todo el mundo". La idea básica de IoT es que prácticamente todos los objetos físicos están conectados a Internet a través de redes cableadas o inalámbricas.

Las aplicaciones web e Internet, junto con los desarrollos de IoT, actualmente lideran las últimas tendencias tecnológicas en muchos sectores productivos. Las tendencias tecnológicas actuales se centran en tecnologías habilitadoras como sensores inalámbricos, lectores de códigos de barras, antenas de identificación por radiofrecuencia y etiquetas. Sin embargo, estas tendencias están centradas en el hardware y en realidad no se enfocan tanto en cómo se obtiene y utiliza la información para conducir los procesos a nivel de la cadena de suministro [5].

En el trabajo de Brewster [1] clasifica los desafíos del IoT para el sector agroalimentario en dos tipos, estos son los estructurales y los tecnológicos. Cada clase incluye diferentes desafíos, tal como se detalla en la Tabla 1.

De los aspectos mostrados en la tabla 1 nos enfocaremos principalmente en detallar los aspectos tecnológicos.

Conectividad. Este es un aspecto clave para el monitoreo de huertos agrícolas mediante dispositivos IoT, ya que la conectividad no siempre está disponible en las áreas rurales. Hay algunas opciones: para las granjas cercanas a los centros urbanos, es posible tener acceso a Internet a través de WiFi; para las áreas rurales debe haber conexión a través de la red celular; y para esas zonas rurales muy remotas, la opción es la conectividad por satélite.

Capacidad de Procesamiento de Datos. Los sensores y actuadores IoT no tienen su propio sistema operativo y pueden tener capacidades de procesamiento y almacenamiento muy limitadas, por lo que deben conectarse a dispositivos o puertas de enlace para vincular, a través de internet, los dispositivos IoT y la nube.

Tabla 1. Retos para la adopción de dispositivos IoT para el sector agroalimentario, adaptado para el contexto de los huertos urbanos. Elaborado con información [1].

Aspectos Estructurales	Aspectos Tecnológicos
Heterogeneidad del sector Tamaños de los huertos y costos de inversión de capital en tecnología Modelos comerciales o financiamiento en huertos privados o públicos Confidencialidad de datos en huertos privados Aceptación social y de los usuarios	Interoperabilidad Conectividad Poder de Procesamiento de Datos Falta de Políticas de Gobernabilidad de los Datos Seguridad y Privacidad de los Datos en huertos privados Disponibilidad de los Datos en huertos públicos

Potencia energética. Los sensores y los actuadores generalmente funcionan sin su propia alimentación, la energía es suministrada por fuentes externas. Para granjas remotas, este es un problema para considerar.

Gobernanza de datos, seguridad y privacidad. El sistema de IoT o la plataforma desarrollada debe garantizar el control completo de los datos por parte del interesado que genera los datos. Por ejemplo, si los datos se generan en el huerto, por lo tanto, el agricultor será el propietario de estos datos. Esto proporciona a los interesados una sensación de control y, por lo tanto, también formalizará la privacidad de la información y permitiría a los agricultores ver los datos como una fuente potencial de ingresos [1].

Interoperabilidad. La interoperabilidad en arquitecturas que contemplan dispositivos IoT y están basadas en la nube, para agricultura en general, se pueden categorizar como una característica clave al realizar un intercambio de información sin problemas en uno o varios sistemas o aplicaciones [3]. El principal habilitador que respalda dicho intercambio continuo de información reside en el uso de **modelos de datos**. Definir modelos de datos para sistemas interoperables dentro del dominio agrícola requiere identificar el vocabulario común de las cosas y entidades, sus propiedades y relaciones y luego representar este conocimiento utilizando ontologías.

Las ontologías se pueden desarrollar utilizando lenguajes basados en web para permitir una integración e interoperabilidad más completa y, por lo tanto, para proporcionar un marco de referencia para obtener modelos de datos legibles por máquina.

Un modelo de datos se requiere para la obtención de información de forma estructurada y organizado de tal forma que la extracción de la información se realice de forma sencilla y práctica.

2.2. Monitoreo de un huerto urbano

Los aspectos por monitorear son básicos y se enlistan a continuación:

- El riego;
- La ubicación del almácigo, maceta o el sustrato donde se colocará la planta;
- El clima, se debe tener cuidado con el tipo de planta y la estación del año;
- Enfermedades y plagas, en caso de aparición debe utilizarse el manejo físico u aplicación de insecticidas orgánicos;
- Fertilización, cuando exista carencia o necesidad del cultivo de algún nutriente deberá realizar una dosificación correcta de acuerdo con el tipo de sustrato:
- La poda, cuando se requiera debe hacerse una poda a la parte superior de la planta buscando se enriquezca de follaje;
- Mantenimiento, se mantiene libre de hierbas que no pertenecen al cultivo, y
- Si la planta es una guía y necesita de un tutor deberá siempre en revisión con la finalidad de que no se quiebre la planta y no se pierda el producto.

2.3. Soluciones de tecnología disponibles para un buen manejo del huerto urbano

En la actualidad existen un sin fin de equipos, insumos, asesoría y herramientas que pueden facilitar el manejo de un huerto urbano, algunas de las opciones se muestran a continuación (la lista no es exhaustiva):

- Hidronova. http://hidronova.mx/compra
- Proain tecnología Agrícola: http://proain.com/
- Ecocán: https://www.ecocanjardineria.com.es
- SU HUERTA en casa: http://suhuertoencasa.com/
- PRIVA: http://www.priva-international.com
- Growinpallets: http://www.growinpallet.com/es

3. Modelo de datos para el monitoreo de un huerto urbano

El modelo de información de contexto de FIWARE NGSI, se basa en la noción de entidades y sus atributos. Las entidades representan objetos físicos arbitrarios, y los atributos representan las propiedades de estos objetos [13]. La comunidad de FIWARE ha propuesto modelos de datos disponibles en [14], dentro de los cuales se encuentran clasificados por dominio:

Edificio,
 Dispositivos,
 Medio ambiente,
 Parques y jardines,
 Puntos de interés,
 Alumbrado público,

Estacionamiento,
 Transporte,

Clima,
 Manejo de desechos.

En la figura 1 se ilustra el diagrama UML de los modelos de datos reutilizados como instancias para realizar la composición del modelo de datos específico para los

Huertos Urbanos. Los modelos de datos reutilizados se encuentran publicados en [9]. Los modelos de datos reutilizados son: Garden, Device Model, Location, Adress, Flower Bed, Orchad Type, Greenspace Record. A continuación, se muestra un ejemplo extraído del modelo de datos general para Huertos Urbanos que en la actualidad existen.

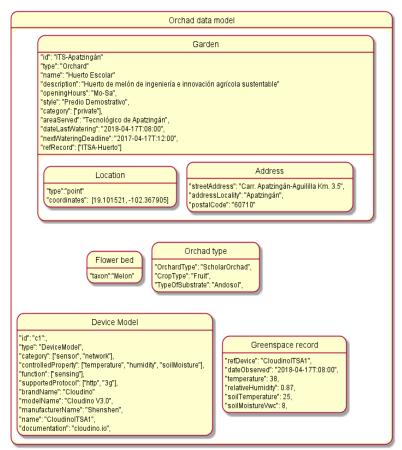


Fig. 1. Diagrama de los modelos de datos utilizados para obtener el modelo de datos para los Huertos Urbanos.

Orchad Data Model Example

```
{
    "id": "ITS-Apatzingán",
    "type": "Orchard",
    "name": "Huerto Escolar",
    "description": "Huerto de Melón de Ingeniería en Innovación
Agrícola Sustentable",
    "OrchardType": "ScholarOrchad",
    "CropType": "Fruit",
    "TypeOfSubstrate": "Andosol",
    "taxon": "Melon",
    "location": {
        "type": "Point",
        "type": "Point",
```

```
"coordinates": [19.101521, -102.367905]
        },
        "address": {
            "streetAddress": "Carr. Apatzingán-Aguililla Km. 3.5",
            "addressLocality": "Apatzingán",
            "postalCode": "60710"
        },
        "openingHours": "Mo-Sa",
        "style": "Predio Demostrativo",
       "category": ["private"],
        "areaServed": "Tecnológico de Apatzingán",
        "dateLastWatering": "2018-04-17T:08:00",
        "nextWateringDeadline": "2017-04-17T:12:00",
        "refRecord": ["ITSA-Huerto"]
        "device": {
            "id": "c1":,
            "type": "DeviceModel",
            "category": ["sensor", "network"],
            "controlledProperty":
                                                              "humidity",
                                    ["temperature",
"soilMoisture"],
            "function": ["sensing"],
            "supportedProtocol": ["http", "3g"],
            "brandName": "ArduinoUno"
            "modelName": "ArduinoUno V3.0",
            "manufacturerName": "Shenshen",
            "name": "ArUnoITSA1",
            "documentation": "https://store.arduino.cc/usa/arduino-uno-
rev3",
},
        "refDevice": "CloudinoITSA1",
        "dateObserved": "2018-04-17T:08:00",
        "temperature": 38,
        "relativeHumidity": 0.87,
        "soilMoistureVwc": 8,
    }
```

3.1. Validación del modelo de datos

El modelo de datos se validó utilizando un prototipo experimental como el que se ilustra en la Figura 2. Se utilizaron sensores de temperatura, humedad relativa y humedad del sustrato. Dichos sensores se conectaron a nodos compuestos de un Arduino nano y un módulo de bluetooth.

La comunicación entre los nodos identificados en la figura 2 como IoT-Device y el Gateway se realizaba vía Bluetooth. La comunicación entre el Gateway y el Context Broker de FIWARE se realizó mediante WiFi. El Orion Context Broker es un habilitador genérico para el desarrollo de aplicaciones en la nube, contiene información sobre el contexto actual. Sin embargo, la información de contexto evoluciona con el tiempo, creando un historial de contexto [15].

Para la visualización de los datos generados por los sensores se utilizó una APP desarrollada en Android por la empresa IMPRONTA Desarrollo en Sistemas Informáticos S.A. de C.V.

Modelo de datos para el monitoreo de variables agro-climatológicas de huertos urbanos...

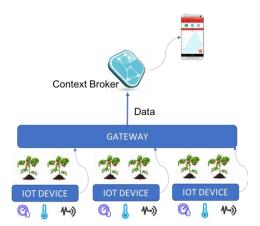


Fig. 2. Esquema general del prototipo utilizado para la validación del Modelo de Datos para el Monitoreo de Huertos Urbanos.

La Aplicación móvil muestra de forma gráfica los datos tomados de los sensores en tiempo real, o el histórico de datos.

4. Conclusiones

En el presente trabajo se describió la importancia de los huertos urbanos, los requerimientos para llevar un buen cuidado de los cultivos, así como los aspectos a considerar durante el desarrollo de un huerto.

Se resaltó la importancia de utilizar nuevas tecnologías que permitan mejorar el cuidado de los huertos, realizar mejores mediciones de las variables agroclimatológicas que influyen en el crecimiento y bienestar de las plantas.

Se propuso un modelo de datos reutilizando modelos de datos previamente desarrollados por la comunidad que colabora en FIWARE. Dicho modelo fue validado utilizando un prototipo compuesto de dispositivos IoT y uno de los Habilitadores Genéricos de FIWARE conocido como Context Broker.

Referencias

- 1. Brewster, C., Roussaki, I., Kalatzis, N., Doolong, K., Ellis, K.: IoT in Agriculture: Designing a Europe-Wide Large-Scale Pilot. IEEE Communications Magazine (2017)
- 2. Morán Alonso, N.: Huertos urbanos en tres ciudades europeas: Londres, Berlín, Madrid. Boletín CF+S, 47/48, pp. 75–124 (2011)
- 3. Murdock, P., et al.: Semantic Interoperability for the Web of Things. Research Gate (2016)
- 4. SAGARPA: El huerto familiar [FT-HF] Ficha técnica (1999)
- 5. Verdouw, C., Wolfert, J., Beulens, A., Rialland, A.: Virtualization of food supply chains with the internet of things. Journal of Food Engineering 176, pp. 128–136 (2016)
- 6. FAO Portal Web: Crear Ciudades más verdes. 19 pp. (2009)
- 7. FAO Portal Web: Ciudades verdes, pp. 9-20 (2014)
- 8. FAO Portal Web (2018)

ISSN 1870-4069

9. FIWARE Portal Web, https://www.fiware.org/developers/data-models/ (2018)

Omar Jehovani López Orozco, Miguel Gozález Mendoza, Juan Manuel Olvera Santoyo, et al.

- 10. Agrohuerto Página Web: Beneficios de los huertos urbanos para ti y tu ciudad (2018)
- 11. FAO: Portal web, Panorama general (2018)
- 12. FAO: Cuestiones de la agricultura urbana, Revista electrónica, Enfoques (1999)
- 13. FIWARE: NGSI Association (2018)
- 14. FIWARE Data Models (2018)
- 15. FIWARE Developers Catalogue (2018)

Algoritmo descentralizado de ruteo de tráfico vehicular con información local inspirado en el algoritmo de hormigas

Alejandro Maldonado-Navarro, Carlos Gershenson-García

Universidad Nacional Autónoma de México, Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas, México mna_17@uxmcc2.iimas.unam.mx, cgg@unam.mx

Resumen. Los métodos convencionales de ruteo siempre han encontrado dificultades para equilibrar el flujo óptimo de la red y del usuario, así como para optimizar los tiempos de viaje. Por lo tanto, es importante reflexionar sobre la necesidad de nuevos métodos para administrar el tráfico sobre una red vehicular que permitan reducir el tiempo de traslado y aumentar el flujo de la red, explotando así todos los recursos disponibles. En este artículo, se propone un modelo de tráfico vehicular basado en autómatas celulares, en el cual se explora un mecanismo de ruteo basado en información local, junto con un factor de penalización basado en el algoritmo de colonia de hormigas. Los resultados de los experimentos con diferentes densidades y diferentes parámetros son examinados, mostrando que el mecanismo aquí propuesto logra incrementar hasta en un 100 % el flujo de la red y el número de viajes que se pueden realizar en determinadas densidades.

Palabras clave: algoritmo de ruteo, autómatas celulares, tráfico vehicular, optimización de colonia de hormigas, autoorganización.

Decentralized Algorithm of Vehicular Traffic Routing with Local Information Inspired by Ant Algorithm

Abstract. Conventional routing methods have always found it difficult to balance the optimal flow of the network and the user, as well as to optimize travel times. Therefore, it is important to reflect on the need of new methods to manage the traffic on a vehicular network that allows to reduce the travel time and increase the flow of the network, exploiting all the available resources. In this article, a model of vehicular traffic based on cellular automata is proposed, in which a routing mechanism based on local information is explored with a penalty factor based on the ant colony algorithm. The results of the experiments with different densities and different parameters are examined, showing that, the mechanism proposed here increases up to 100 % the network flow and the number of trips that can be made in certain densities.

Keywords: routing algorithm, cellular automata, vehicular traffic, ant colony optimization, self-organization.

1. Introducción

El problema de movilidad en México crece junto al número de automóviles en las calles y reduce la disponibilidad que éstas tienen. Algunos estudios [12,15] revelan cómo el incremento del parque vehicular está aumentando más rápido que el incremento de la infraestructura que lo soporta. Es decir, cada día hay más coches y menos vías en donde puedan circular.

La perspectiva futura para México, es que esta situación se agrava ante el continuo crecimiento del parque vehicular, que se estima llegue a 70 millones de vehículos en el año 2030 [8]. Por este motivo, es importante hacer consciencia sobre de la gravedad de la situación.

La solución a este problema puede seguir tres diferentes enfoques [5]: a) reducir el número total de vehículos en las calles, b) incrementar la infraestructura carretera, c) incrementar el flujo del tráfico con la infraestructura carretera existente. En la Ciudad de México ya se han propuesto diferentes estrategias para reducir el número de coches en las calles. También se ha incrementado la infraestructura. Sin embargo, no hay resultados con gran impacto. Por lo tanto, mejorar la movilidad con el uso de la infraestructura actual es una opción que debe de explorarse.

En la literatura podemos encontrar diferentes formas de resolver el problema haciendo uso de la infraestructura, por ejemplo, optimizar el flujo por medio de semáforos autoorganizantes [6], mejorar el tráfico mediante mecanismos de ruteo [10,2,16], o utilizar autoorganización y comunicación entre vehículos [5,18], en este documento, nos centraremos principalmente en el segundo enfoque: mejorar el tráfico mediante mecanismos de ruteo, especialmente mecanismos inspirados en la autoorganización de las hormigas, tales mecanismos han demostrado ser eficientes. Por ejemplo [10] propone un sistema de tráfico inteligente distribuido (DITS), el cual incrementa la velocidad media de los vehículos, mientras que [2] propone un mecanismo de antiferomona que logra reducir los tiempos de viaje.

Es clara la necesidad de buscar nuevos mecanismos de ruteo que permitan reducir el tiempo y la distancia del recorrido de los usuarios y de toda la red. En este artículo se propone un modelo de tráfico vehicular basado en autómatas celulares, en el cual se explora un algoritmo de ruteo que usa información local, junto con un factor de penalización basado en el algoritmo de colonia de hormigas.

Para la elaboración de los experimentos se propuso un marco de trabajo, el cual se compone de dos elementos: el primero es un modelo de tráfico vehicular basado en autómatas celulares, en el cual se explora el algoritmo propuesto, este modelo se compone de un grupo de reglas que definen la dinámica entre los vehículos. El segundo elemento es el algoritmo de ruteo para los vehículos, compuesto de un grupo de reglas que definen la forma en que interactúan la

feromona, el escenario y los vehículos, además de un algoritmo voraz basado en una heurística y una función de penalización para encontrar una ruta subóptima.

El documento está ordenado de la siguiente forma: en la sección 2 se exponen algunos conceptos básicos. En la sección 3 se explica el marco de trabajo: primero se introduce el modelo de tráfico vehícular propuesto, luego de ello se expone el algoritmo que se va a utilizar. En la sección 4 se da una breve explicación de la metodología utilizada para realizar los experimentos. A continuación, en la sección 5 se muestran los resultados. Finalmente, en la última sección se discuten los resultados, se dan conclusiones y se muestran ideas sobre trabajos futuros.

2. Conceptos básicos

2.1. Algoritmo de hormigas

El algoritmo de optimización por colonia de hormigas, o por sus siglas en inglés ACO, fue introducido inicialmente como *Ant System* [3] o sistema de hormigas por Dorigo en el año de 1996, este algoritmo describe cómo las hormigas son capaces de encontrar la ruta más corta entre su fuente de alimento y su hogar.

En el artículo [3] se considera el ejemplo de la Fig. 1 , en el cual existe un camino donde las hormigas están caminando desde A hacia E (Fig. 1 (a)). A continuación, un obstáculo aparece en el camino y lo interrumpe. "Entonces en la posición B las hormigas que caminan de A hacia E (o de la posición D aquellas que caminan en dirección opuesta) tienen que decidir si girar a la derecha o a la izquierda (Fig. 1 (b)). La respuesta es influenciada por la intensidad de la feromona dejada por las hormigas anteriores. Un alto nivel de feromona en el camino correcto da a una hormiga un estímulo más fuerte y en consecuencia una probabilidad más alta de tomar el camino correcto" [3], de tal suerte que, la primera hormiga en llegar a B tiene la misma probabilidad de girar a la derecha o a la izquierda, ya que el camino BCD es más corto que BHD, la hormiga en llegar más rápido será la que tome el camino BCD, causando que la hormiga proveniente de D ahora escoja el camino CBA para llegar a A, en consecuencia este camino se va a fortalecer y por lo tanto las probabilidades de que este sea escogido por las siguientes hormigas se incrementan (Fig. 1 (c)).

En este artículo se propone un mecanismo de ruteo, parte del cual es inspirado en el algoritmo de hormigas anteriormente descrito, especialmente en el uso de la feromona. En el artículo original, esta feromona "da información sobre cuántas hormigas en el pasado han escogido el mismo camino". En este artículo, la feromona nos dará información de cuántos vehículos han escogido un camino.

En el modelo propuesto la red inicia con una cantidad de feromona. Al pasar por las vías los vehículos recogen feromona (a diferencia del algoritmo original en el cual se deja feromona), la feromona de cada vía se recupera en cierta cantidad en cada paso de tiempo, la dinámica de los vehículos en las intersecciones consiste en tomar la ruta con mayor cantidad de feromona y menor distancia. La implementación de este mecanismo es descrita más a detalle en la sección 3.

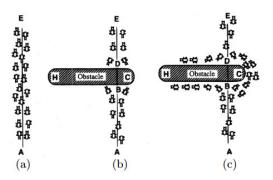


Fig. 1. "Un ejemplo con hormigas reales. (a) Las hormigas siguen el camino entre los puntos A y E. (b) Un obstáculo es interpuesto; las hormigas pueden escoger para ir alrededor de él siguiendo uno de los dos caminos con probabilidades iguales. (c) En el camino más corto se concentra más feromona" [3].

2.2. Tráfico vehicular

Velocidad. La velocidad v se puede apreciar como el escalar resultante de la distancia d que existe entre un punto A hacia un punto B dividido en el tiempo t que tarda en desplazarse, la fórmula se puede establecer en la ecuación (1):

$$v = \frac{d}{t},\tag{1}$$

donde v = velocidad, d = distancia y t el tiempo.

Densidad. Es el número de vehículos que se encuentran sobre un tramo carretero, es decir, vehículos por el tamaño de la sección carretera, usualmente se usan vehículos por milla o vehículos por kilómetro, se define mediante la ecuación (2):

$$k = \frac{veh}{d},\tag{2}$$

donde k es la densidad, veh es el número de vehículos y d es el tamaño de la sección carretera.

Flujo. Greenshields [7] estableció la relación entre densidad, velocidad y flujo, la cual se puede interpretar como la cantidad de vehículos que pasan por un tramo carretero en cierto tiempo. El flujo también se puede obtener a partir de la densidad y la velocidad (ecuación (3)):

$$q = \frac{veh}{t} = \frac{veh}{d}\frac{d}{t} = kv, \tag{3}$$

donde q es el flujo, veh es el número de vehículos, t es el tiempo y d es el tamaño de la sección carretera.

Diagrama fundamental. El término diagrama fundamental en sí mismo es históricamente basado en la hipótesis fundamental de Lighthill y Whitham [11], de la existencia de una curva de concentración de flujo (Fig. 2). Este enfoque nos permite observar la relación de las variables y estudiar los dos estados principales del tráfico, el flujo libre y el tráfico congestionado.

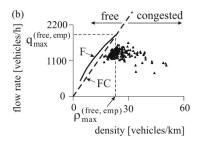


Fig. 2. Diagrama fundamental [9].

2.3. Modelos de tráfico vehicular

Los siguiente modelos de tráfico vehicular son basados en autómatas celulares (AC), los AC fueron usados inicialmente por von Neumann y Ulam [14]. Los AC son idealizaciones matemáticas de sistemas físicos. Un AC puede ser considerado unidimensional o bidimensional, esto según la dimensión del escenario. Esencialmente un AC evoluciona en espacios y tiempos discretos. Los valores de las variables principalmente se ven afectados por las variables de sus vecinos [19].

NaSch. El modelo NaSch fue propuesto en 1992 por Nagel y Schreckenberg [13]. Básicamente, describe la interacción entre vehículos sobre autopistas. Se considera estocástico debido a la desaceleración aleatoria y se basa principalmente en las siguientes cuatro reglas:

- 1. Aceleración: si la velocidad v de un vehículo es menor que v_{max} y si la distancia hacia el siguiente carro es mayor que v+1, la velocidad es incrementada en uno, v=v+1
- 2. Desaceleración (debido a otros carros): si un vehículo en el sitio i observa que el siguiente vehículo está en el sitio i+j (con $j \leq v$), este reduce su velocidad a j-1 (v=j-1).
- 3. Aleatoriedad: con una probabilidad de p, la velocidad de cada vehículo (si es mayor que cero) es decrementada por uno (v = v 1).
- 4. Movimiento del carro: cada vehículo avanza v lugares.

BML. Este fue un modelo propuesto en 1992 [1]. Es un modelo autoorganizado, considerado determinista (excepto por su inicialización aleatoria). Consiste en

una matriz en cuyas celdas es colocada una cantidad n de vehículos de forma aleatoria. Estos vehículos sólo pueden moverse en un único sentido, hacia arriba o hacia la derecha. Lo anterior ocurre en su versión asíncrona (o modelo I), siempre y cuando la siguiente celda no esté bloqueada. En el primer tiempo se mueven todos los de un sentido y a continuación, se mueven los del otro sentido, es decir, se establece un ciclo con dos etapas de turnos de movimientos.

3. Marco de trabajo

3.1. Modelo de tráfico vehicular

El modelo fue fuertemente inspirado por [13,1] el primero funge como modelo para autopistas, el segundo para ciudades, se tomaron como referencia [16,4] para la dinámica home to work y la topología Manhattan-like, también se tomaron ideas de [17] para la reducción de velocidad al acercarse a una intersección. La idea planteada en este trabajo se asemeja a la de [16] el cual describe la elección de rutas dinámicas entre la residencia y el trabajo.

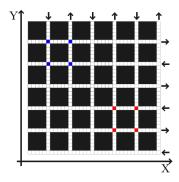


Fig. 3. Escenario o topología del modelo de tráfico vehicular, las flechas indican el sentido de las calles, los cuadros azules de la esquina superior izquierda denota los trabajos mientras que, en los cuadros rojos se establecen los hogares.

El escenario del modelo. Este se define en una cuadrícula de $n \times n$ celdas, donde se establecen $l \times l$ calles, las cuales se forman por conjuntos de celdas. A cada sentido de las calles le corresponde l/4, los sentidos de las calles son izquierda, derecha, arriba y abajo. Cada calle puede tener un sólo sentido. El escenario es considerado con fronteras cíclicas, dado que el escenario es como un toroide y los vehículos pueden pasar de un extremo a otro. La velocidad máxima permitida se establece en V_{max} .

En la inicialización de la simulación, los vehículos se ubican en el escenario de forma aleatoria. En cada paso de tiempo, estos tienen una sola dirección (norte, sur, este u oeste) y una velocidad $[0,1,...,V_{max}]$. En el cuadrante superior izquierdo se establecen los hogares, mientras que en el inferior derecho los trabajos (Fig. 3). Al inicio, a los vehículos se les asigna un lugar de trabajo y un hogar. Estos pueden ser compartidos, es decir, varios vehículos pueden tener el mismo trabajo y el mismo hogar. La dinámica principal de los vehículos es ir de su trabajo a su hogar y viceversa cuando llegan a uno de estos sitios.

La extensión de BML realizada, considera los cuatro sentidos posibles: iz-quierda L, derecha R, arriba U y abajo D. En el escenario, los vehículos pueden corresponder a uno de los cuatro vectores, esto según su orientación, de tal modo que en cada tiempo t existen cuatro vectores del tipo $X_t = \{x_{t,1}, x_{t,2}, ..., x_{t,n}\}$, donde t indica el tiempo, y $x_{t,n}$ hace referencia al vehículo n perteneciente al vector X en el tiempo t. En resumen, se tendrían los siguiente vectores: $L_t = \{l_{t,1}, l_{t,2}, ..., l_{t,n}\}, R_t = \{r_{t,1}, r_{t,2}, ..., r_{t,n}\}, U_t = \{u_{t,1}, u_{t,2}, ..., u_{t,n}\}, D_t = \{d_{t,1}, d_{t,2}, ..., d_{t,n}\}$

La función de transición se aplica de forma sincronizada a todos los elementos pertenecientes a cada vector, es decir, todos los vehículos del vector X cambian de posición al mismo tiempo. En contraste, el mecanismo se vuelve asíncrono cuando se va a ejecutar esta acción en cada vector, de tal modo que, primero se aplica la función de transición al vector L (todos los orientados a la izquierda se mueven de forma sincronizada) luego al R, al U y finalmente al D, cada una de estas cuatro tareas es llamada fase y, a la labor de realizar estas cuatro fases se le llamará ciclo asíncrono de movimientos (CAM).

En cada fase del CAM, sólo se contemplan los vehículos dentro de sus vectores originales. Es decir, si el vehículo $x_{t,1}$ cambia de orientación, solo podrá avanzar con su nueva orientación en t+1. Esto nos servirá de mucho, ya que a diferencia del BML original, este modelo permite las vueltas a la derecha y a la izquierda, siempre y cuando se respeten las direcciones de las calles y no haya obstáculos. Se considera una vuelta cuando un vehículo x_i que pertenece al vector con orientación X, termina su movimiento en una intersección con una nueva orientación Y. Por ejemplo, que el vehículo 3 del vector con orientación hacia la derecha termine en una intersección con orientación hacia arriba, es considerado una vuelta, cuando los vehículos caen en un escenario como el anterior, estos sólo pueden salir de la intersección hasta su correspondiente fase del siguiente CAM.

Finalmente el conjunto de reglas que definen la interacción entre los vehículos se describe de la siguiente manera:

- 1. Aceleración: si la velocidad v de un vehículo es menor que V_{max} y la distancia al vehículo de enfrente es mayor que v+1, la velocidad es incrementada en uno.
- 2. Aceleración por intersección: si el vehículo en la celda i observa una intersección en la celda i+1 y su velocidad es 0, entonces su velocidad se establece en 1.

- 3. Desaceleración (debido a otros coches): Si el vehículo en la celda i observa un vehículo en i + j con j menor o igual a v, entonces reduce su velocidad a j 1.
- 4. Desaceleración (por intersección): si el vehículo en la celda i observa una intersección en la celda i+j con j menor o igual que v, entonces reduce su velocidad a j 1.
- 5. Aleatoriedad: Con una probabilidad P, la velocidad de cada vehículo (si es mayor a 0), es decrementada en 1.
- 6. Movimiento del carro: Cada vehículo avanza v lugares.

Algoritmo de ruteo. El algoritmo propuesto pertenece a la clase de algoritmos voraces, es decir, elige la opción óptima en cada paso local. Su función de costo se compone principalmente de tres elementos: el primero es la distancia entre la ubicación actual y la siguiente intersección, el segundo es la heurística, definida como la distancia euclidiana entre la siguiente intersección y el objetivo, el último elemento es un factor penalización, el cual se determina usando la feromona del escenario.

El uso de la feromona se inspira principalmente en el algoritmo de colonia de hormigas, el cual fue explicado brevemente en la sección 2. La forma en que interactúan la feromona, los vehículos y el escenario se define de la siguiente manera:

- 1. Inicialización: Al inicializar el escenario, todas las celdas tienen un nivel de feromona P_{max} .
- 2. Incremento: Cada paso de tiempo t, la feromona de cada celda i incrementa en P_{inc} unidades, es decir $P_{i,t} = P_{i,t-1} + P_{inc}$ si $P_{i,t} > P_{max}$ entonces $P_{i,t} = P_{max}$.
- 3. Decremento: Cada vez que un vehículo pasa de la celda i hasta la celda j, la feromona P de las celdas que van de $i \to j$ decrementan en P_{dec} unidades, es decir $P_{x,t} = P_{x,t-1} P_{dec}$ si $P_{x,t} < 0$ entonces $P_{x,t} = 0$, $x \in i \to j$.

El algoritmo propuesto, usa una heurística basada en la distancia euclidiana de la siguiente intersección al objetivo, y un valor de penalización basado en el nivel de feromona. Los resultados que podemos extraer con él, pueden ser considerados subóptimos debido a que al hacer uso de información local, puede no alcanzar el resultado óptimo.

La Fig. 4 muestra brevemente la descripción de los datos usados por el algoritmo. Los vehículos usan este algoritmo al llegar a una intersección, permitiéndoles elegir la calle que en promedio tiene el mayor nivel de feromona y menor distancia hacia su destino. Esto se obtiene mediante la función de costo descrita en la ecuación (4). En esta se evalúa la distancia euclidiana d_{in} que existe desde su posición actual hacia la intersección, más la distancia euclidiana d_{nt} que hay desde la intersección evaluada hasta el objetivo. El resultado obtenido de la operación anterior es multiplicado por una función de penalización definida por el inverso del promedio de nivel de feromona $\hat{p}_{n,t}$ existente en las celdas del camino n que se está evaluando en el tiempo t (ecuación (5)), más uno . Con

esto se puede decir entonces que, a mayor feromona menor penalización y, a menor feromona mayor penalización (ecuación (4)).

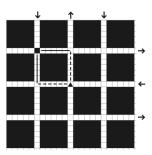


Fig. 4. Algoritmo propuesto. Las flechas en los costados del mapa indican el sentido de las calles, el triángulo en el centro representa al vehículo y su orientación. El objetivo es denotado por el cuadrado ubicado en la parte superior izquierda que yace en la intersección. El conjunto de celdas bajo la línea punteada son consideradas como la calle a evaluar. Dada la topología, siempre existen dos. La línea punteada marca la distancia de la ubicación actual a la siguiente intersección, mientras que la línea negra marca la distancia de la intersección al objetivo.

Por lo tanto, cuando $\hat{p} = 0$, es decir, la calle está completamente saturada, el valor de la ruta R_n se convierte sólo en el resultado de la operación $d_{in} + d_{nt}$, de tal modo que el valor máximo para R_n es igual a $d_{in} + d_{nt}$, y que, este valor será entonces la peor ruta a tomar:

$$R_n = (d_{in} + d_{nt}) * \frac{1}{\hat{p}_{n,t} + 1}, \tag{4}$$

de donde

$$\widehat{p}_{n,t} = \frac{1}{k} \sum_{i=0}^{k} p_{i,t}, \tag{5}$$

donde R_n es el coste del camino n, d_{in} es la distancia euclidiana de la ubicación actual hacia la siguiente intersección, d_{nt} es la distancia euclidiana de la siguiente intersección al objetivo (heuristica), y $\widehat{p}_{n,t}$ es el nivel de feromona p promedio en el grupo de celdas que conforman el camino n de tamaño k a evaluar, en el tiempo t.

Esta función de costo, es parecida a la función de costo usada en [16]. Sin embargo, en aquel trabajo se expone como factor de penalización el uso de la densidad elevado a un factor alfa. Esta función puede apreciarse en la ecuación (6):

$$P_n = (d_{in} + d_{nt})(1 + c_{in})^{\alpha}, \tag{6}$$

donde P_n es el costo del camino, d_{in} la distancia del nodo actual al nodo siguiente, d_{nt} la distancia del siguiente nodo al nodo objetivo, c_{in} representa la congestión o densidad del camino $i \to n$, y α es un factor de corrección [16].

En el caso de este trabajo, el uso de la feromona ayuda a conservar información del pasado, como si se tratara de una memoria a corto plazo, a diferencia de la densidad que nos da información únicamente de lo que está ocurriendo en determinado momento. Ésta es la ventaja que ofrece este nuevo enfoque respecto al anterior.

4. Metodología

Para realizar los experimentos se hizo uso de la herramienta NetLogo 6.0, la cual permite recrear escenarios mediante el uso de agentes y celdas (tortugas y parcelas). Se hicieron $N \times N \times M$ experimentos debido a que se buscó la mejor combinación del incremento P_{inc} y decremento P_{dec} de la feromona, diferentes valores desde 0 hasta 10 fueron probados, obteniendo un total de 121 resultados para los parámetros P_{inc} y P_{dec} .

Cada uno de estos 121 experimentos fue conformado por M sub experimentos, esto con la intención de explorar las diferentes densidades, que fueron desde 1 hasta M vehículos con ciertos parámetros P_{inc} y P_{dec} , estos sub experimentos fueron ejecutados con un periodo de 350 pasos o ciclos asíncronos de movimientos (CAM) cada uno, de tal modo que se elaboraron 121 experimentos que exploran las diferentes posibilidades de parámetros P_{inc} , P_{dec} y de 899 diferentes densidades, es decir se hicieron un total de 108,779 experimentos.

La mejor combinación se obtuvo a través del análisis del número de viajes realizados en cierto periodo de tiempo entre los trabajos y los hogares o viceversa (al los cuales llamaremos simplemente viajes o recorridos). El objetivo principal fue encontrar aquella combinación que maximiza el número de viajes realizados. Principalmente se extrajeron y estudiaron los siguientes datos:

(a) El número total de viajes que se pueden realizar en determinado periodo de tiempo. Es decir, cuántos vehículos pueden llegar de su origen a su destino. Este valor es denotado por n, el tamaño del vector de la ecuación (7):

$$\overline{X} = \{x_1, x_2, ..., x_n\},$$
 (7)

donde \overline{X} indica el vector de las distancias recorridas en cada viaje, y n el tamaño de este vector, es decir, la cantidad de viajes realizados.

(b) El promedio de los viajes realizados según los parámetros P_{inc} y P_{dec} , con uno hasta m vehículos, es denotado en la ecuación (8) por $\hat{J}_{P_{inc},P_{dec}}$:

$$\widehat{J}_{P_{inc}, P_{dec}} = \frac{L}{M},\tag{8}$$

donde L indica el total de viajes realizados en todas las densidades (ecuación (9)):

$$L = \sum_{m=1}^{M} n_m, \tag{9}$$

- donde n_m representa el número de viajes que se pudieron realizar con la densidad m, y M es igual al número total de vehículos que se pueden tener en el escenario.
- (c) El flujo promedio de los vehículos con una cantidad m de vehículos en el escenario, es definida por la ecuación (10), donde v_i es la velocidad del vehículo i en el paso 350, que es el último en la simulación, k_m representa la densidad con m vehículos, esto se expone en la ecuación (11), donde A indica el número de celdas total que son calle en el escenario, el cual se puede calcular mediante la ecuación (12):

$$\mu_m = \frac{1}{m} \sum_{i=1}^m v_i k_m, \tag{10}$$

$$k_m = \frac{m}{A},\tag{11}$$

$$A = n^2 + (n^2 2L), (12)$$

donde n es el número de calles que que hay en vertical y horizontal y l es el número de celdas por calle.

(d) El flujo promedio de los vehículos según los parámetros P_{inc} y P_{dec} determinado por $\widehat{\mu}_{P_{inc},P_{dec}}$ en la ecuación (13), donde M es igual al número máximo de vehículos que se puede tener en el escenario con:

$$\widehat{\mu}_{P_{inc}, P_{dec}} = \frac{1}{M} \sum_{m=1}^{M} \mu_m.$$
(13)

Los parámetros estáticos para el experimento se muestran en la tabla 1 : el tamaño de cada celda es considerado de 7.5 metros, debido a que es aproximadamente el espacio ocupado por un coche que no está en movimiento [13]. Para una calle se consideran 12 celdas, 90 metros, una aproximación realista a lo que mide una cuadra en ciudades compactas como Manhattan.

Tabla 1. Parámetros estáticos del mundo y el modelo.

Mundo	Tamaño de la calle	12 celdas (90 metros)
	Número de calles	6 calles
	Número máximo de vehículos	899
Modelo	Velocidad máxima	3 celdas por tick (40 km/h)
	Probabilidad de paro	.3

5. Resultados

La información obtenida puede ser resumida en los mapas de calor mostrados en la Fig. 5, donde podemos observar la variación de los resultados según los parámetros P_{inc} y P_{dec} . También es posible observar un patrón o tendencia del incremento de viajes e incremento del flujo. Para ambos mapas de calor es posible apreciar la necesidad de mantener un nivel de P_{inc} mayor que P_{dec} para obtener datos relevantes. La Fig. 5 (a) muestra entonces una tendencia al incremento de valores desde $P_{inc} = 6$ hasta $P_{inc} = 3$ (siempre y cuando $P_{inc} > P_{dec}$) y un ligero decremento desde $P_{inc} = 2$ hasta $P_{inc} = 1$. Casi el mismo patrón de incremento ocurre en el mapa de calor de la Fig. 5 (b).

Tabla 2. Las cinco mejores combinaciones de parámetros.

#	P_{inc}	P_{dec}	Viajes
1	2	3	318
2	1	2	315
3	2	4	312
4	3	5	312
5	3	4	311

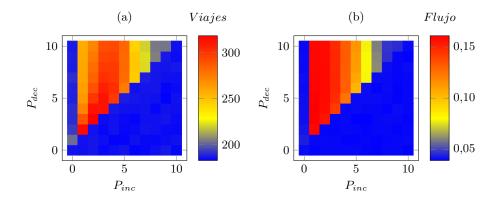


Fig. 5. Mapas de calor conforme a los parámetros P_{inc} y P_{dec} . Las zonas azules representan los valores más pequeños, las zonas naranjas valores intermedios, mientras que las zonas rojas los valores más altos. (a) El promedio de viajes que es posible realizar en 350 pasos o CAMs conforme a los parámetros, que se obtienen mediante la ecuación (8). (b) El flujo promedio conforme a los parámetros, que se obtienen mediante la ecuación (13).

Los resultados más relevantes son extraídos de los experimentos Exp00 ($P_{inc} = 0, P_{dec} = 0$) y Exp23 ($P_{inc} = 2, P_{dec} = 3$), se considera como línea base el expe-

rimento Exp00 ya que los niveles de feromona no van a cambiar y sería como no tenerla en cuenta. Por otra parte, el experimento Exp23 fue considerado ya que al observar la tabla 2, donde se establece una lista de los cinco mejores parámetros, es posible apreciar que los parámetros pertenecientes a este experimento tienen el mayor número de viajes.

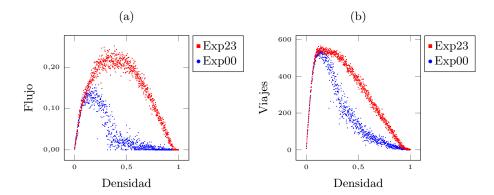


Fig. 6. Resultados obtenidos en las diferentes densidades, desde uno hasta 899 vehículos, para ambas figuras, donde Exp23 representa los resultados del experimento con los parámetros $P_{inc} = 2$, $P_{dec} = 3$, y Exp00 representa el experimento con los parámetros $P_{inc} = 0$, $P_{dec} = 0$. (a) Diagrama fundamental de densidad contra flujo. Es posible observar el comportamiento esperado y el cambio de fase con una densidad de 0.08. Este dato es calculado mediante la ecuación (10) el flujo promedio en cierta densidad. (b) Número de viajes que son posibles de realizar en un periodo de 350 pasos o CAMs. Este dato se calcula mediante la ecuación (7).

Las gráficas de la Fig. 6 muestran los resultados individuales de tales parámetros, el diagrama fundamental de la Fig. 6 (a) muestra de forma clara el incremento del flujo en toda la red con el uso de los parámetros 2 y 3, mientras que el experimento con parámetros en cero muestra un comportamiento típico de un diagrama fundamental, por otro lado es posible observar en la Fig. 6 (b), como el uso de los parámetros 2 y 3 incrementa de manera relevante el número de viajes que son posibles realizar, suavizando de forma significativa la caída de la curva.

6. Discusión

Se realizaron un total de 108 779 experimentos, los cuales permitieron indagar 121 diferentes combinaciones de parámetros de feromona, de este análisis se extrajo la mejor combinación, la cual fue comparada contra un experimento donde la feromona no tenía relevancia, de esta comparación se concluye que:

(a) El diagrama fundamental del experimento Exp00, demuestra un comportamiento típico del flujo de la red en sus diferentes densidades.

- (b) La mejor configuración de parámetros obtenida a través de la selección mediante la maximización de los viajes (Exp23), demuestra un incremento considerable de flujo de vehículos (de hasta el 100%) en sus diferentes densidades, esto se puede ver en la gráfica 6 (a), donde también es posible apreciar cómo ocurre la transición de fase de tráfico fluido a tráfico congestionado.
- (c) El número de viajes realizados en ambos experimentos tiende a ser el mismo hasta densidades cercanas a 0.2. Luego de esta densidad, los datos que describen la curva del experimento Exp23 superan en gran medida los del experimento Exp00, de tal modo que con el experimento Exp23 es posible observar un incremento en el número de viajes que se pueden realizar en determinado tiempo.
- (d) El algoritmo con el uso de la feromona incrementó hasta en un 100 % el flujo y el número de viajes que se pueden realizar en densidades entre 0.4 y 0.6, en comparación con un algoritmo de ruteo sin feromona.

Como trabajos futuros se plantea el estudio minucioso de diferentes tamaños del escenario. De igual forma, indagar sobre diferentes topologías, con el fin de analizar la relación que estos parámetros tienen con respecto a los parámetros de incremento y decremento de la feromona.

Agradecimientos. Este trabajo fue realizado gracias a los recursos del Centro de Ciencias de la Complejidad (C3) de la Universidad Nacional Autónoma de México, el primer autor agradece también a CONACYT-México por la beca otorgada para sus estudios de posgrado.

Referencias

- Biham, O., Middleton, A.A., Levine, D.: Self-organization and a dynamical transition in traffic-flow models. Physical Review A 46(10), R6124–R6127 (nov 1992)
- 2. Dallmeyer, J., Schumann, R., Lattner, A.D., Timm, I.J.: Don't go with the ant flow: Ant-inspired traffic routing in urban environments. Journal of Intelligent Transportation Systems 19(1), 78–88 (sep 2014)
- Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 26(1), 29–41 (1996)
- 4. Fang, J., Shi, J., Chen, X.Q., Qin, Z.: A two-dimensional CA traffic model with dynamic route choices vetween residence and workplace. International Journal of Modern Physics C 21(02), 221–237 (feb 2010)
- 5. Ferreira, M., Fernandes, R., Conceição, H., Viriyasitavat, W., Tonguz, O.K.: Self-organized traffic control. In: Proceedings of the seventh ACM international workshop on VehiculAr InterNETworking VANET 10. ACM Press (2010)
- Gershenson, C., Rosenblueth, D.A.: Self-organizing traffic lights at multiple-street intersections. Complexity 17(4), 23–39 (dec 2011)
- Greenshields, B.D., Thompson, J.T., Dickinson, H.C., Swinton, R.S.: The photographic method of studying traffic behavior. Proceedings of the Thirteenth Annual Meeting of the Highway Research Board Held at Washington, D.C. 13, 382–399 (dec 1934)

- 8. INE, C.: Analysis of the Automotive Industry in Mexico. México. Instituto Nacional de Ecología (National Institute of Ecology) Centro de Transporte Sostenible (Centre for Sustainable Transport) (2012)
- 9. kerner, B.: Introduction to Modern Traffic Flow Theory and Control. Springer Berlin Heidelberg (2009)
- Kponyo, J., Nwizege, K., Opare, K., Ahmed, A.R., Hamdoun, H., L.O.Akazua, Alshehri, S., Frank, H.: A distributed intelligent traffic system using ant colony optimization: A NetLogo modeling approach. In: 2016 International Conference on Systems Informatics, Modelling and Simulation (SIMS). IEEE (jun 2016)
- 11. Lighthill, M.J., Whitham, G.B.: On kinematic waves. II. a theory of traffic flow on long crowded roads. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 229(1178), 317–345 (may 1955)
- Medina Ramírez, S.: La importancia de reducción del uso del automóvil en México.
 Tendencias de motorización, del uso del automóvil y de sus impactos. México.
 Instituto de Políticas para el Transporte y el Desarrollo (2012)
- Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. Journal de Physique I 2(12), 2221–2229 (dec 1992)
- Neumann, J.V.: Theory of Self-Reproducing Automata. University of Illinois Press, Champaign, IL, USA (1966)
- 15. Perez Leon, M.A.: Modelado de trafico vehicular (2017), tesis de maestria
- Scellato, S., Fortuna, L., Frasca, M., Gómez-Gardeñes, J., Latora, V.: Traffic optimization in transport networks based on local routing. The European Physical Journal B 73(2), 303–308 (dec 2009)
- 17. Tonguz, O., Viriyasitavat, W., Bai, F.: Modeling urban traffic: A cellular automata approach. IEEE Communications Magazine 47(5), 142–150 (may 2009)
- Tonguz, O.K., Viriyasitavat, W.: A self-organizing network approach to priority management at intersections. IEEE Communications Magazine 54(6), 119–127 (jun 2016)
- Wolfram, S.: Statistical mechanics of cellular automata. Rev. Mod. Phys. 55(3), 601–644 (Jul 1983)

Middleware aplicado a la simulación de fluidos con técnicas de Realidad Aumentada

J.L. López¹, D. Villanueva^{2,3}, H. Estrada³, M. R. Zarate⁴

¹ Universidad Carlos III de Madrid, Computer Science Department, Madrid, Spain

² CONACyT (Consejo Nacional de Ciencia y Tecnología), Dirección de Cátedras, Mexico City, Mexico

³ INFOTEC (Center of Research and Innovation in Information Technology Communications), Aguascalientes, Ags., Mexico

> ⁴ Universidad Tecnológica del Centro de Veracruz, Cuitláhuac, Ver., México

Resumen. Actualmente, la tecnología de Realidad Aumentada se implementa en diferentes disciplinas que brindan múltiples soluciones al mundo real. Por lo tanto, un progreso constante para el desarrollo de herramientas y software en áreas de investigación que van desde el entretenimiento y educación hasta el campo de la medicina. Del mismo modo, el conocimiento y la aplicación de la tecnología de Realidad Aumentada se extiende a áreas que no están directamente relacionadas con la computación. En este artículo, se presenta un middleware como un puente entre un proyecto de simulación de fluidos y Realidad Aumentada. El trabajo facilita el manejo de la información sobre los fluidos. Además, permite la visualización de objetos con técnicas de Realidad Aumentada. El middleware integra módulos para la administración, el manejo de los fluidos y el uso de los modelos de Realidad Aumentada. Finalmente, su aplicación sirve como una herramienta experimental para un proceso de enseñanza-aprendizaje en educación, específicamente en el área de las ciencias químicas.

Palabras clave: middleware, simulación de fluidos, realidad aumentada.

Middleware Applied to the Simulation of Fluids with Augmented Reality Techniques

Abstract. Nowadays, Augmented Reality technology is implemented in different disciplines, which provides multiple solutions to the real world. Therefore, a constant progress for the development of tools and software in research areas ranging from entertainment and education to the field of medicine. In the same way, the knowledge and application of Augmented

Reality technology extends to areas that are not directly related to computing. In this article, we present a middleware which as a bridge between a fluid simulation project and the Augmented Reality. The work facilitates the handling of information about fluids. In addition, it allows the visualization of objects with Augmented Reality techniques. Besides, the middleware integrates modules for administration, fluid management and the use of Augmented Reality models. Finally, its application serves as an experimental tool for a teaching-learning process in education, specifically in the area of chemical sciences.

Keywords: middleware, fluid simulation, augmented reality.

1. Introducción

Mediante la implementación de técnicas de Realidad Aumentada (RA) se han incrementado considerablemente el desarrollo de aplicaciones para diferentes áreas de aplicación, que van desde la medicina [1], hasta el entretenimiento [2]. Asimismo, en la explotación en otras áreas que no se encuentran relacionadas directamente con la informática. Por ejemplo, en áreas de la educación [3].

El desarrollo de aplicaciones que utilizan esta tecnología resulta de interés para su uso en áreas como la educación, ya que permiten simular procesos que proporcionan una visión distinta, en consecuencia, ratifican el entendimiento sobre cómo se comportan algunos elementos en diferentes entornos y situaciones. Por un lado, la RA permite identificar nuevos usos o estrategias para el uso de distintos elementos en diferentes campos de aplicación. Por otro lado, la simulación de fluidos se aplica en ramas cuyo propósito es representar fluidos de diferentes entornos y situaciones, esto puede ir desde una explosión en un videojuego, hasta simulaciones de reacciones químicas [4] y [5].

En este contexto, la simulación se emplea en diferentes áreas, tales como, la electrónica [6], robótica [7], nanotecnología [8], mecánica [9], sistemas computacionales [10], entre otros. Sin embargo, la química es una de las disciplinas que se emplea en los procesos físicos, matemáticos y de mecánica de los fluidos, ya que se pueden representar las características de los fluidos bajo diferentes situaciones [11]. Por lo tanto, la RA puede ser empleada en el área de las ciencias químicas para enseñar a estudiantes sin la necesidad de un laboratorio físico y, sobre todo, sin el peligro que pueden llegar a representar el uso de fluidos.

El presente trabajo de investigación propone la creación de un middleware, el cual, combina el uso de un método de simulación de fluidos y un marco de trabajo de RA con modelos 3D. La integración permite la simulación de diferentes tipos de fluidos y sus distintas características. Finalmente, se ha desarrollado una aplicación para el análisis del comportamiento de los fluidos en un entorno de RA, la herramienta impacta en el área de la educación, específicamente en estudiantes a nivel secundaria en la materia de química.

2. Trabajo previo

2.1. Realidad Aumentada

El término realidad aumentada fue considerado por dos ingenieros en 1962 [12]. Sin embargo, el término adquiere presencia en el mundo científico a principios de los años 90 cuando la tecnología basada en ordenadores de procesamiento rápido, técnicas de renderizado de gráficos en tiempo real y sistemas de seguimiento de precisión portables, permitieron implementar la combinación de imágenes generadas por el ordenador sobre la visión del mundo real que tiene un usuario [13].

Actualmente, existen muchas aplicaciones que se encuentran asociadas a objetos del mundo real. Por ejemplo, Tom y David en la empresa Boeing propusieron el uso de esta tecnología para mejorar la eficiencia de las tareas que realizaban los operadores asociados a la fabricación de aviones [14] o quizá una de las aplicaciones más conocidas de la RA en la educación sea el proyecto Magic Book del grupo activo HIT de Nueva Zelanda donde el alumno lee un libro real a través de un visualizador de mano y ve sobre las páginas reales contenidos virtuales. De esta manera cuando el alumno ve una escena de RA que le gusta puede introducirse dentro de la escena y experimentarla en un entorno virtual inmersivo [15]. Ronald presenta un estudio de aplicaciones en diferentes áreas utilizando técnicas de RA [16]. Por un lado, la RA se encarga de estudiar las técnicas y tecnologías que permiten integrar en tiempo real el contenido digital con el mundo real [17].

A diferencia de la realidad virtual, la realidad aumentada permite generar capas de información que, a su vez, se encargan de alinear la información virtual para crear una integración de los datos. No obstante, el término no debe ser confundido con el de realidad mixta, pues integra realidad aumentada, realidad virtual y virtualización aumentada superponiendo la interacción de objetos reales junto a objetos virtuales [18] y [16]. Azuma menciona que la realidad aumentada debe cumplir tres características: combinar el mundo real y virtual, interactuar en tiempo real y la alineación en 3D [3]. Por otro lado, la RA se usa para describir la combinación de tecnologías que permiten mezclar en tiempo real, el contenido generado por computadora con video en vivo [19]. Además, se utiliza en múltiples escenarios para la simulación, construcción, diseño, entre otros. La RA mantiene una diferencia con la realidad virtual, ya que la RA es un ambiente completamente virtual y sintético, todo ocurre dentro de la computadora sin interacción con el exterior y la realidad aumentada hace una combinación de elementos del mundo real con elementos virtuales, además en la realidad virtual todo es un ambiente completamente virtual [20].

2.2. Simulación de fluidos

La simulación de fluidos es una herramienta muy utilizada para la generación de gráficos por computadora que permiten generar animaciones realistas, tales como, agua, humo, explosiones, y fenómenos relacionados [21]. Dada alguna configuración de entrada de un fluido y una geometría de escenario, un simulador de fluidos realiza

la evolución del movimiento del fluido hacia adelante en el tiempo, haciendo uso de las (posiblemente muy simplificadas) ecuaciones de Navier-Stokes que describen la física de los fluidos [22].

En los gráficos por computadora, tales simulaciones varían en complejidad desde animaciones de alta calidad que consumen una cantidad notable de tiempo para efectos visuales y de películas, hasta sistemas simples de partícula en tiempo real usados en juegos modernos. Para poder simular fluidos en dispositivos electrónicos como lo son computadoras y celular se desarrollaron métodos de simulación que permiten el trabajo sobre GPU (Graphics Process Unity) en este caso SPH que es un método de simulación de fluidos en tiempo real sobre procesadores gráficos [23].

La simulación de fluidos permite la creación de simulaciones gráficas y al mismo tiempo, la representación de diferentes tipos de fluidos; estos pueden ir desde una gota de agua hasta una explosión, este tipo de simulaciones son muy utilizadas sobre todo en las ramas de la física y en la creación de videojuegos donde se genera un mayor realismo [24]. Asimismo, la simulación de fluidos conlleva varios aspectos, uno de estos es la precisión que se requiere cuando se desea interactuar con dichos fluidos, actualmente existen varios métodos pertenecientes a la rama de mecánica de los fluidos (dinámica de fluidos) que permite una simulación de fluidos en tiempo real [25].

Actualmente, los laboratorios en el área de la química requieren de equipos e instrumentos para la realización de diferentes pruebas de experimentación sobre distintos tipos de reacciones químicas, tales como, matraces, pipetas, probetas, etc. Por un lado, la obtención de estos instrumentos a largo plazo genera gastos, por consiguiente, el hecho de que las personas que realizan dichas pruebas se expongan a riesgos inherentes al manejo de diferentes substancias. Por lo tanto, el presente trabajo propone el desarrollo un middleware que trabaje en conjunto con la RA y Simulación de Fluidos, que sirvan como una herramienta de experimentación para el manejo de diferentes elementos y fluidos líquidos en un área de enseñanza en la educación.

2.3. Middleware

Un middleware se define como un software de conectividad que consiste en un conjunto de servicios que permiten interactuar a múltiples procesos que se ejecutan en distintas máquinas a través de una red. Además, oculta la heterogeneidad y provee de un modelo de programación conveniente para los desarrolladores de aplicaciones [26]. Asimismo, es un software que puede incrementar significativamente la reusabilidad mediante soluciones utilizables rápidamente y basadas en estándares aplicables a problemas y tareas comunes en programación. Al mismo tiempo, permite concentrarse en asuntos propios de la aplicación y olvidarse de problemas comunes, estructurales o no, ya resueltos previamente de forma elegante y satisfactoria [27].

También un middleware es una clase de tecnología que posee una abstracción de programación común diseñada para ayudar a gestionar la complejidad y heterogeneidad de los sistemas distribuidos [28], generalmente es una aplicación de software que asiste a una aplicación para poder comunicarse con otras aplicaciones.

2.4. Marcadores

Un marcador visual es una imagen que contiene algún código preestablecido, el cual puede ser decodificado por un dispositivo para el reconocimiento de objetos o transferencia de información [29]. La RA usa marcadores cuya principal característica es el uso de imágenes llamadas marcadores. Por lo tanto, la cámara reconoce los marcadores, busca la imagen asociada y la coloca sobre el marcador. El marcador puede estar impreso en una hoja de papel para que la cámara logre reconocer el marcador, sin verse afectado por el brillo y contraste que existe en el ambiente [30] y [31].

3. Arquitectura de software propuesta

La arquitectura de software integra un conjunto de estructuras, propiedades y relaciones que conforman el "edificio" que dará soporte a la solución software que se va a desarrollar y, además su implementación es de vital importancia, básicamente por tres razones claves: la primera, facilita la comunicación entre las partes interesadas en el desarrollo del software. La segunda, puede tomar decisiones tempranas y evitar un impacto mayor en una etapa posterior del desarrollo del proyecto. La tercera razón es porque permite comprender fácilmente la estructura y el flujo de trabajo de sus componentes. Además, la arquitectura de software es considerada un puente entre la fase de diseño y la ingeniería de requerimientos, debido a que tiene una relación directa entre decisiones de arquitectura y los requisitos.

Se ha identificado que, a pesar de que la mayoría de los sistemas son distintos entre sí, existen algunas similitudes entre las arquitecturas de los mismos [32]. Sommerville, también coincide en que los patrones son una forma de presentar, compartir y, sobre todo, reutilizar el conocimiento sobre los sistemas de software. Además, agrega que un patrón arquitectónico es una descripción abstracta estilizada de buena práctica, que se ensayó y se puso a prueba en diferentes sistemas y entornos, de tal forma que se pueda demostrar que el estilo arquitectónico propuesto tuvo éxito en sistemas anteriores. Asimismo, agrega que los patrones no son aptos para todos los problemas y es por eso que a la hora de definirlo, debe dejarse en claro qué problema resuelve, cuándo es conveniente utilizarlo y cuáles son sus fortalezas y debilidades [33]. Para comprender la arquitectura se hace uso de diagramas que son la representación visual del software y la forma en que ha sido implementado, mismos que otorgan una perspectiva clara y detallada de su estructura.

3.1. Vista física del software propuesto

La vista física de la arquitectura del sistema muestra la interacción entre los elementos que la componen, por un lado la aplicación de escritorio que se encarga de gestionar los fluidos y la generación del marcador personalizado y por otra parte, la aplicación móvil que hace uso de los marcadores, para la representación visual de los

elementos. Además, el flujo de la información entre los diferentes componentes del software que tiene como actor principal al middleware, ver Fig. 1.

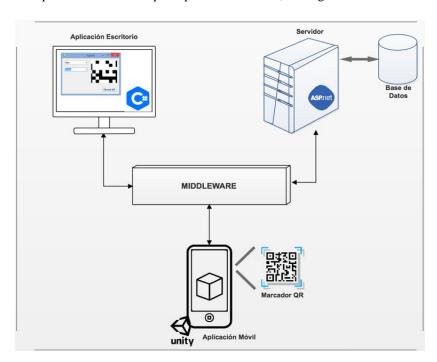


Fig. 1. Vista física del sistema implementando el middleware.

Una descripción de los componentes del software se realiza a continuación:

Aplicación de escritorio. Tiene como objetivo la configuración de un fluido y la generación de un marcador personalizado, que posteriormente podrá ser utilizado para su representación mediante RA.

Servidor & Base de Datos. Se encargan de controlar las peticiones y respuestas de datos y el almacenamiento de la información relacionada de los fluidos y los marcadores.

Aplicación móvil. Realiza las peticiones hacia la coincidencia encontrada entre el marcador identificado y su representación al modelo de RA.

Middleware. Es el intermediario que administra la comunicación entre los diferentes módulos del sistema, una de sus funciones principales para este trabajo, es asegurar que las peticiones hacia los datos obtengan una respuesta correcta. En el caso de estudio propuesto, asegura la conexión entre aplicaciones con tecnologías distintas.

Con la implementación del middleware se limita la cantidad de trabajo y proceso que el dispositivo móvil tiene que llevar a cabo, ya que la carga de configuración del fluido y la generación del Marcador, se deja a la aplicación de escritorio. Por lo tanto, la única tarea que procesa el móvil es la representación de los elementos, en este caso, de los fluidos.

3.2. Arquitectura del software

La aplicación de escritorio se desarrolla bajo el patrón arquitectural MVC (Modelo, Vista, Controlador), cuyo propósito es separar las capas de la aplicación, es decir, si un elemento de la capa vista requiere la interacción con un elemento de la capa del modelo (base de datos), esta petición deberá ser atendida por el controlador, ya que es el encargado de controlar las peticiones entre la aplicación [34] y [35]. Al patrón MVC implementado se le añade la comunicación con una capa middleware para que en conjunto provea de la funcionalidad requerida. Unity es un motor para crear videojuegos multiplataforma creado por Unity Technologies. Es una plataforma de desarrollo para Windows y OS X, además permite la creación de videojuegos para Windows, OS X, Linux, Xbox 360, Play Station 3, Wii, Wii U, iPad, iPhone, Android y Windows Phone [36] y [37]. En este trabajo, Unity se implementa como una herramienta para el diseño y creación de los modelos 3D, ya que su motor de videojuegos permite hacer uso de esos modelos y exportarlos.

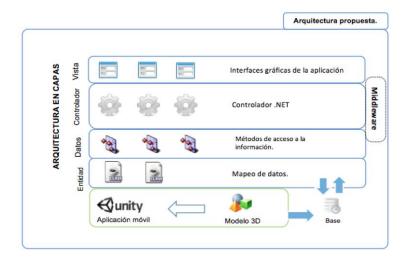


Fig. 2. Arquitectura lógica del software.

La arquitectura lógica del software propuesta para el simulador de fluidos, se implementa bajo el patrón arquitectural MVC, la cual permite agregar la capa del middleware. La Fig. 2, muestra las capas y los componentes que las conforman, las cuales se describen a continuación:

- Capa vista: se encarga de desplegar la interfaz gráfica, que permiten la interacción con el usuario.
- Capa controlador: esta capa funciona como un intermediario entre las diferentes capas de la aplicación, cualquier petición que se realice por parte de usuario o almacenar en la base de datos tendrá que ser el controlador el que realice las llamadas a los métodos o instancias necesarias para hacer dichas tarea.

- Capa entidad y datos: en esta capa se encuentran los componentes necesarios para manejo y aseguramiento de la integridad de la información, por ejemplos los métodos de consulta, inserción, modificación y eliminación de la información que se maneja en el sistema.
- Middleware: se visualiza como parte de la arquitectura, para conocer que existe una comunicación con la aplicación de escritorio e integración con el sistema.

En este trabajo se hace uso del software Unity 3D como una herramienta para la manipulación completa de objetos 3D y, se hace uso de Vuforia como el plug-in que se agrega a Unity, el cual, implementa la aplicación que permite visualizar los objetos 3D con RA.

4. Evaluación y discusión

La implementación de un middleware aplicado a la simulación de fluidos permite la comunicación entre los diferentes módulos del sistema utilizando técnicas de RA, los resultados sobre el desarrollo e implementación se describen a continuación. Por un lado, las propiedades que se consideran para de cada fluido son, nombre, viscosidad, masa molar, densidad, color y un código. En cuanto al código, es asignado de manera automática por el sistema, los datos de cada fluido son almacenados en base datos para su posterior consulta. Por otro lado, las características para la simulación de fluidos de elementos con características distintas requieren de cálculos que permiten obtener los nuevos valores. La mezcla se realiza con base a los criterios; masa molar, viscosidad y densidad, siendo estas características las que permiten que la simulación de fluidos se visualice gráficamente de forma distinta. Los cálculos de la masa molar se obtienen a partir de la sumatoria de la masa molar de los elementos a mezclar, la unidad que representa la masa molar es *g/mol*, revisar ecuación (1)

$$M = m_1 + m_2, (1)$$

donde:

M = masa molar,

 $m_I =$ masa molar del elemento 1,

 m_2 = masa molar del elemento 2.

Para la densidad se hace uso de los valores, cantidad y densidad de los elementos que se requieren mezclar, tal y como se observa en la ecuación (2), obteniéndose en g/cm^3 .

$$\rho = \frac{((c_1 * d_1) + (c_2 * d_2))}{(c_1 + c_2)},\tag{2}$$

donde:

 ρ = densidad,

 c_1 = cantidad del elemento 1,

 d_1 = densidad del elemento 1.

 c_2 = cantidad del elemento 2,

 $d_2 = densidad \ del \ elemento \ 2.$

La viscosidad se obtiene mediante la cantidad y viscosidad de los elementos que se requieren mezclar, ver ecuación (3).

$$v = \frac{((c_1 * v_1) + (c_2 * v_2))}{(c_1 + c_2)},\tag{3}$$

donde:

v = viscosidad,

 c_1 = cantidad del elemento 1,

 v_1 = densidad del elemento 1,

 c_2 = cantidad del elemento 2,

 v_2 = densidad del elemento 2.

Finalmente, la cantidad total se calcula a partir de la cantidad de cada elemento, tal y como se observa en la ecuación (4), la cual, se expresa en mililitros *ml*.

$$c = c_1 + c_2, \tag{4}$$

donde:

c = Cantidad,

 c_1 = Cantidad del elemento 1,

 c_2 = Cantidad del elemento 2.

La interfaz de usuario del simulador de fluidos permite seleccionar los elementos, en este caso fluidos: los cuales se basan en el tipo, color, y cantidad, (ver Fig. 3). Además, la interfaz permite por un lado, seleccionar un elemento previamente almacenado. Por otro lado, asignar un color e indicar la cantidad para pasar a la elección del contenedor y la generación del marcador.

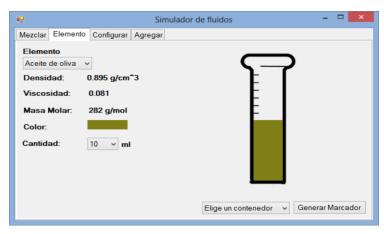


Fig. 3. Interfaz de la aplicación de escritorio.

Una vez configurado cada elemento se procede a la generación del marcador correspondiente, el cual contiene las características específicas de cada elemento, además de una clave única para cada fluido. La Fig. 4, muestra un ejemplo de los marcadores generados.

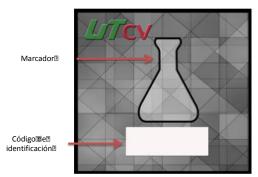




Fig. 4. Marcador personalizado para los fluidos.

El prototipo de visualización de los elementos previamente generados en el simulador de fluidos se visualiza en la Fig. 5. El proceso para la visualización requiere que la cámara busque y reconozca tanto el marcador como el código de identificación el cual se asociará al elemento 3D.



Fig. 5. Visualización de elementos mediante la aplicación móvil.

5. Conclusiones y trabajo futuro

La implementación de un middleware integra conceptos de simulación básica de fluidos, un modelo de visualización de objetos 3D y una aplicación móvil. Mediante técnicas de RA facilitan el proceso de enseñanza aprendizaje en ciencias como la química. Las pruebas realizadas ofrecen la posibilidad al usuario de una experiencia sobre RA para manipular herramientas de medición, tales como, probeta graduada, matraz de Erlenmeyer, pipeta, que, en conjunto de sustancias, tales como el agua, el vinagre, el mercurio, entre otros, generan un aprendizaje a través de la observación en las reacciones generadas en un aspecto visual, kinestésico y auditivo.

Asimismo, el middleware ayuda a disminuir la carga de trabajo soportada por el dispositivo móvil, ya que las tareas de configuración de fluidos y creación de

marcadores se realiza desde la aplicación de escritorio. Además, la adaptación del patrón arquitectural MVC sobre la estructura de la aplicación móvil y su comunicación con el middleware facilita la construcción del software.

Finalmente, el patrón MVC provee de una lógica de tal forma que requiere que las capas de la aplicación se encuentren separadas, lo que permite conseguir una ventaja de extensibilidad y mantenibilidad para el software.

Referencias

- 1. Ploder, O., Wagner, A., Enislidis, G., Ewers, R.: Computer-assisted intraoperative visualization of dental implants. Augmented reality in medicine. Der Radiologe, 35(9), pp. 569–572 (1995)
- 2. Stapleton, C., Hughes, C., Moshell, M., Micikevicius, P., Altman, M.: Applying mixed reality to entertainment. Computer (Long. Beach. Calif), 35(12), pp. 122–124 (2002)
- 3. Azuma, R. T. A survey of augmented reality Presence Teleoperators virtual Environments, 6(4), pp. 355–385 (1997)
- 4. Nee A., S. Ong, and G. Chryssolouris: Augmented reality applications in design and manufacturing. CIRP Annals-manufacturing technology, 61(2), pp. 657–679 (2012)
- Marescaux, J., Clément, J. M., Tassetti, V., Koehl, C., Cotin, S., Russier, Y., Ayache, N.: Virtual reality applied to hepatic surgery simulation: the next revolution. Annals of surgery, 228(5), p. 627 (1998)
- 6. Eckmiller, R.: Electronic Simulation of the Vertebrate Retina. IEEE Trans. Biomed. Eng., vol. BME-22, no. 4, pp. pp. 305–311 (1975)
- 7. Echeverria, G., Lassabe, N., Degroote, A., Lemaignan, S.: Modular open robots simulation engine: MORSE. In: IEEE International Conference on Robotics and Automation, 2011, pp. 46–51, (2011)
- 8. Qiao, W., McLennan, M., Kennell, R., Ebert, D., Klimeck, G.: Hub-based Simulation and Graphics Hardware Accelerated Visualization for Nanotechnology Applications. IEEE Trans. Vis. Comput. Graph. 12(5), pp. 1061–1068 (2006)
- 9. De, S. K., Aluru, N. R.: A chemo-electro-mechanical mathematical model for simulation of pH sensitive hydrogels. Mech. Mater. 36(5), pp. 395–410 (2004)
- Magnusson P. S. et al.: Simics: A full system simulation platform. Computer (Long. Beach. Calif) 35(2,) pp. 50–58 (2002)
- 11. Greenspan, H. P.: On fluid-mechanical simulations of cell division and movement. J. Theor. Biol. 70(1), pp. 125–134 (1978)
- 12. Caudell, T., Mizell, D.: Augmented reality: An application of heads-up display technology to manual manufacturing processes. In: 1992. Proc. Twenty-Fifth (1992)
- 13. Basogain, X., Olabe, M., Espinosa, K., Rouèche, C., Olabe, J. C.: Realidad Aumentada en la Educación: una tecnología emergente. Escuela Superior de Ingeniería de Bilbao, EHU (2018)
- Ribo, M., Pinz, A., Fuhrmann, A. L.: A new optical tracking system for virtual and augmented reality applications. In: IMTC Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics (Cat. No.01CH 37188), vol. 3, pp. 1932–1936 (2001)
- 15. The Human Interface Technology Laboratory New Zealand (HIT Lab NZ): Systems and undefined (2011)
- 16. Billinghurst, M., Clark, A., Lee, G.: A survey of augmented reality. Foundations and Trends® in Human–Computer Interaction, 8(2-3), pp. 73-272 (2015)
- 17. Billinghurst, M., Kato, H.: Collaborative augmented reality. Commun. ACM 45(7), pp. 64–70 (2002)

- Olmedo, H., Augusto, J.: From augmented reality to mixed reality. In: Proceedings of the 13th International Conference on Interacción Persona-Ordenador (p. 36). ACM.'12, pp. 1–2 (2012)
- 19. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent advances in augmented reality. IEEE Comput. Graph. Appl., 21(6), pp. 34–47 (2001)
- Goldsmith, D., Liarokapis, F., Malone, G., Kemp, J.: Augmented Reality Environmental Monitoring Using Wireless Sensor Networks. In: 12th International Conference Information Visualisation 2008, pp. 539–544 (2008)
- 21. Chen, J. X., Lobo, N. D. V., Hughes, C. E., Moshell, J. M.: Real-time fluid simulation in a dynamic virtual environment. IEEE Comput. Graph. Appl. 17(3), pp. 52–61 (1997)
- Li, B., Fleming, C. A.: Three-Dimensional Model of Navier-Stokes Equations for Water Waves. J. Waterw. Port, Coastal, Ocean Eng. 127(1), pp. 16–25 (2001)
- 23. Fay, J.: Introduction to fluid mechanics. MIT Press (1994)
- Vennard, J. K., Street, R. L., Flores, J. Barrios: Elementos de mecánica de fluidos: Versión SI. Compañía Editorial Continental (1979)
- 25. Pashley, D. H.: Dentin: a dynamic substrate--a review. Scanning Microsc. 3(1), pp. 161–74–6 (1989)
- Pietzuch, P. R., Bacon, J. M.: Hermes: A distributed event-based middleware architecture.
 In: Distributed Computing Systems Workshops 2002. Proceedings. 22nd International Conference IEEE, pp. 611-618 (2002)
- 27. Hadim, S., Mohamed, N.: Middleware: Middleware challenges and approaches for wireless sensor networks. IEEE distributed systems online 7(3), p. 1 (2006)
- 28. Bakken D. E.: Middleware, Encyclopedia of Distributed Computing (2002)
- Peña-Cabrera, M., Juarez, I. L., Cabrera, R. R., Osorio, R.: Un Proceso de Aprendizaje para Reconocimiento de Objetos en Línea en Tareas Robotizadas. In: 3ª Conferencia Iberoamericana en Sistemas, Cibernética e Informática, CISCI (2004)
- 30. Pence, H. E., Williams, A. J., Belford, R. E.: New tools and challenges for chemical education: Mobile learning, augmented reality, and distributed cognition in the dawn of the social and semantic web. Chemistry education: Best practices, opportunities and trends. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany (2015)
- 31. Lee, K.: Augmented reality in education and training. Springer TechTrends 56(2), pp. 13–21 (2012)
- 32. Morales-Chaparro, R., Linaje, M., Preciado, J. C., Sánchez-Figueroa, F.: MVC web design patterns and rich internet applications. Proceedings of the Jornadas de Ingenieria del Software y Bases de Datos, pp. 39–46 (2007)
- 33. Pressman, R. S., Troya, J. M: Ingeniería del software (1988)
- 34. Sommerville, I., Sawyer, P. Sawyer: Requirements engineering: a good practice guide. Wiley (1999)
- 35. Ferre, X., Juristo, N., Moreno, A., Sánchez, I.: A Software Architectural View of Usability Patterns. In: 2nd Workshop on Software and Usability Cross-Pollination (at INTERACT'03) Zurich, Switzerland (2003)
- 36. García, A. S.: Desarrollo de aplicaciones para múltiples plataformas y uso de realidad virtual con Unity 3D. Tesis Doctoral (2017)
- 37. González Mora, C.: Videojuego con Realidad Virtual (2017)

Ambiente inteligente de aprendizaje de depuración de errores en programas Java

María Lucia Barrón-Estrada, Ramón Zatarain-Cabada, Minerva Valencia-Rodríguez, Gloria E. Peralta-Peñuñuri

Tecnológico Nacional de México — Instituto Tecnológico de Culiacán, División de Estudios de Posgrado e Investigación, Culiacán, Sinaloa, México {lbarron, rzatarain, mvalencia, gperalta}@itculiacan.edu.mx

Resumen. La tarea de depurar los errores de un programa es una de las más complejas y difíciles para un estudiante que se inicia en el proceso de aprender programación. Los compiladores ofrecen ayudas para realizar dicha tarea, pero ésta se enfoca más a apoyar a programadores expertos que a novatos. En este trabajo se presenta un ambiente inteligente para aprender a depurar errores en el código de programas escritos en el lenguaje de programación Java. La implementación del sistema experto del ambiente se basó en la teoría de seguimiento de ejemplos, para el proceso de aprendizaje se usó un agente pedagógico y técnicas de gamificación. El ambiente fue evaluado exitosamente con estudiantes de nivel básico de ingeniería en sistemas computacionales e ingeniería en tecnologías de la información y comunicaciones.

Palabras clave: java, depuración de errores, ejemplos erróneos, seguimiento de ejemplos, ambiente de aprendizaje, agente pedagógico, gamificación.

Intelligent Learning Environment for Debugging Errors in Java Programs

Abstract. One of the most complex and difficult tasks for a novice programmer as student is to debug a program. Compilers offers tools to carry out this task, which nevertheless focuses more on supporting expert programmers than novices. In this paper we present an intelligent environment to help students learn how to debug code in the Java programming language. Example-tracing theory was used to implement the expert system. In addition, the learning process was supported by a pedagogical agent and gamification techniques. The environment was successfully evaluated with freshmen students of computer systems engineering and engineering in information and communication technologies.

Keywords: java, debugging, incorrect examples, example tracing, learning environment, pedagogical agent, gamification.

1. Introducción

Aprender a desarrollar programas para una computadora, es una tarea compleja que involucra diversas habilidades requeridas durante los pasos de la metodología de desarrollo, que van desde la comprensión del problema y el planteamiento de la solución, la selección del lenguaje de programación y codificación, las pruebas y depuración de errores, finalizando con la ejecución y mantenimiento del programa. La complejidad que representa enseñar y aprender programación [1] ha tomado notoriedad desde sus inicios. Cualquier persona que aprenda a programar debe enfrentarse a la tarea de corregir los diferentes tipos de errores que pueden presentarse en un programa: sintácticos, semánticos, de lógica, etc. Una deficiente habilidad para depurar errores en programas produce frustración en el estudiante y además puede ocasionar la introducción de nuevos errores [2]. La enseñanza a través de ejemplos erróneos [4] se ha utilizado en diferentes áreas como: Matemáticas, Física y Medicina, entre otras, aportando buenos resultados en el aprendizaje de los estudiantes

En el área de Computación, la enseñanza de la programación se centra principalmente en el desarrollo de la lógica algorítmica y el correcto uso de la sintaxis y semántica del lenguaje de programación usado para la codificación; la depuración de errores, pocas veces o nunca es utilizada por los maestros, libros de programación y software educativo como estrategia central de enseñanza. Un enfoque basado en errores fue desarrollado por Ginat [5], para un curso introductorio a la programación orientada a objetos.

La motivación juega un papel importante en los procesos de aprendizaje y actualmente, en el ámbito de los ambientes de aprendizaje, se utilizan diversas técnicas de gamificación para motivar a los estudiantes. La gamificación se refiere a la inclusión de elementos presentes en los videojuegos, en el diseño y desarrollo de aplicaciones correspondientes a un contexto educativo, con la intención de involucrar, motivar y mejorar el aprendizaje. Las mecánicas de juego se refieren a las reglas que se deben cumplir para producir cambios en el sistema. Generalmente se trata de recompensas virtuales [8], como por ejemplo ganar medallas, puntos, insignias o monedas, o bien, subir de nivel o categoría, así como visualizaciones de tablero de jugadores, entre otras.

En el estudio de Stott y sus colegas [9] hablan sobre las dinámicas subyacentes que hacen que los juegos sean atractivos y cómo éstas han sido utilizadas en prácticas pedagógicas modernas.

Por otra parte y en relación a este trabajo, existen metodologías para la representación del conocimiento de un experto que tienen un enfoque cognitivo, el cual se sustenta en la identificación de los procesos mentales durante el aprendizaje. Una de estas metodologías es llamada seguimiento de ejemplos (*example-tracing*). Este método evalúa el comportamiento del estudiante comparándolo flexiblemente contra ejemplos generalizados de comportamiento de resolución de problemas [6].

En seguimiento de ejemplos, los ejemplos generalizados, contienen las rutas de solución aceptables para un problema en particular (ruta óptima y sub-óptimas) además, también es posible registrar comportamiento erróneo (correspondiente a errores comunes) para ofrecer tutoría contextualizada a un tipo de error en particular y tienden a ser más fáciles de crear que las reglas de producción o las restricciones. Los

tutores basados en el método seguimiento de ejemplos pueden exhibir un comportamiento sofisticado de tutoría, ya que es posible proporcionar orientación a nivel paso, es decir, orientación durante la resolución de problemas complejos y no solo orientación hasta la finalización del ejercicio lo que hace que se cumpla con el criterio mínimo para considerase como un Sistema Tutor Inteligente (STI) según Vanlehn [7].

En este artículo, se presenta un ambiente inteligente de aprendizaje de depuración de errores en programas llamado *Find Error Java*, el cual utiliza un agente pedagógico llamado Lucy [3] para presentar al estudiante ejemplos erróneos con el fin de ayudarlo a comprender el error que se presenta en el código del programa y la forma cómo debe resolverlo. El ambiente está apoyado por un sistema experto el cual se construyó usando la metodología seguimiento de ejemplos y técnicas de gamificación para motivar al estudiante en su proceso de aprendizaje.

Este artículo está estructurado de la siguiente forma: en la Sección 2 se presentan los trabajos relacionados en las áreas de depuración de errores, sistemas tutores inteligentes y ambientes de aprendizaje, gamificación y agentes pedagógicos; en la Sección 3 se muestra el ambiente inteligente de aprendizaje desarrollado. La Sección 4 expone las pruebas realizadas y el análisis de las mismas y para finalizar, en la sección 5 se presentan las conclusiones y trabajos futuros.

2. Trabajos relacionados

En esta sección se abordan los temas relacionados con el trabajo que se presenta, tales como: el proceso de depuración de errores en programas, los sistemas tutores inteligentes y ambientes de aprendizaje, la gamificación y los agentes pedagógicos, los cuales que constituyen teoría y estado del arte.

2.1. Depuración de errores

Diversas investigaciones abordan el tema de la depuración de errores de programación partiendo de enfoques diferentes como: identificar el tipo de errores que comúnmente cometen los programadores novatos [10], visualizar dinámicamente la ejecución de programas [11], pronosticar la navegación en la web para la depuración de programas con el objetivo de ampliar herramientas que ofrecen los entornos de desarrollo [12], o abordar factores afectivos como frustración ante la depuración [2]. En estos trabajos se considera a la depuración de errores como una actividad de recuperación, posterior a la codificación y no como un tema central para la enseñanza.

Ahmadzadeh y sus colegas [9] proponen que la comprensión de patrones de depuración en errores de compilación y lógicos ofrece elementos que permiten mejorar el método de enseñanza, mientras que Murphy y otros [2] plantean que las habilidades de depuración consisten en aplicar simultáneamente la comprensión del funcionamiento previsto del programa, entender la ejecución real defectuosa del programa, tener experiencia en programación en general, comprender el lenguaje de programación y el dominio de la aplicación y tener conocimiento de errores y métodos de depuración.

Jackson, Cobb y Carver [13] lograron identificar los errores comunes que comenten los programadores novatos en lenguaje Java, y mencionan que durante su revisión a la literatura encontraron que los métodos más usados para la identificación de errores comunes de programación son el uso de la experiencia, la aplicación de encuestas a maestros, el conteo manual y categorización de errores o bien, el uso de las compilaciones proporcionadas por los mismos alumnos.

McCall y Kölling [14] proponen un método de categorización de errores comunes cometidos por programadores novatos al utilizar el lenguaje de programación Java, basado en los mensajes de diagnóstico que produce el compilador. Los investigadores clasificaron en 10 categorías todos los errores, relacionando los mensajes de error con el código que lo produjo.

En otras investigaciones [4] se ha demostrado que el uso adecuado de ejemplos erróneos conduce a buenos resultados. Aplicarlos en un entorno tecnológico educativo incluye, entre otros aspectos, el diseño de la instrucción, que se refiere al proceso o método de enseñanza, tal como la definición de las habilidades que deben ser fortalecidas, la decisión sobre la manera y el momento para dar retroalimentación al estudiante y la elección o innovación de los recursos pedagógicos-tecnológicos tales como el uso de agentes pedagógicos o recursos de gamificación.

2.2. Sistemas tutores inteligentes y ambientes de aprendizaje

No existe una definición universal aceptada del concepto Sistema Tutor Inteligente. Puede decirse que los Sistemas Tutores Inteligentes (STI) son aquellos que ofrecen una enseñanza diferencial, aquellos que adaptan su respuesta de enseñanza después de realizar un razonamiento sobre las necesidades de los estudiantes y el conocimiento del dominio [15].

Los STI se caracterizan por poseer mecanismos de inteligencia artificial que permiten adaptar la tutoría, pero generalmente no brindan al estudiante la posibilidad de elegir los temas o ejercicios del dominio. Los ambientes de aprendizaje a diferencia de los STI, ponen a disposición del estudiante todo el contenido educativo ofreciéndole la posibilidad de elegir los temas y secuencia de sus actividades de estudio, sin embargo, no cuentan con características de inteligencia artificial con el fin de guiar la tutoría.

Los Entornos Inteligentes de Aprendizaje (EIA) incorporan características de los ambientes de aprendizaje (flexibilidad de navegabilidad en los temas de instrucción) y los Sistemas Tutores Inteligentes (adaptabilidad de las experiencias de tutoría conforme al modelo del estudiante). Estas características que anteriormente eran consideradas como contradictorias hoy se consideran como complementarias [16].

2.3. Gamificación y agentes pedagógicos

Los videojuegos se han vuelto muy populares en los últimos años. Aunque el término gamificación, surgió en el ámbito digital y tuvo una aceptación generalizada después del 2010 [17], es importante mencionar que desde los años ochenta los investigadores han estudiado los beneficios de los enfoques basados en juegos en la educación [18,19].

Por otra parte, uno de los elementos importantes en los sistemas educativos es el tutor o profesor que determina la estrategia de enseñanza a utilizar para que los

estudiantes adquieran el conocimiento. En los sistemas usados en el aprendizaje electrónico, los tutores son llamados agentes pedagógicos (AP) y frecuentemente se representan a través de personajes que pueden ser reales o animados; ellos se encargan de establecer la comunicación con los estudiantes interactuando de diferentes formas para proporcionarles el apoyo cognitivo que requieren.

3. Ambiente inteligente de aprendizaje

3.1. Descripción general

Find Error Java es un ambiente inteligente de aprendizaje en el que se provee al estudiante una forma sencilla y ágil de aprender a evitar o superar errores comunes que se cometen durante las primeras etapas de aprendizaje de programación Java. El sistema está diseñado para ser usado por estudiantes novatos de nivel escolar bachillerato o universidad que se encuentren estudiando cursos introductorios de programación Java. Básicamente el estudiante debe identificar, entender y corregir el error sintáctico, semántico o lógico que contiene un pequeño programa o fragmento de código presentado.

Para desarrollar el ambiente de aprendizaje *Find Error Java* fue necesario establecer los elementos clave que permitirán su correcto funcionamiento para proporcionar asistencia a los estudiantes al momento de resolver los ejercicios planteados y además poder medir su aceptación y eficacia.

La Figura 1 muestra los 5 elementos clave definidos para el desarrollo del sistema.

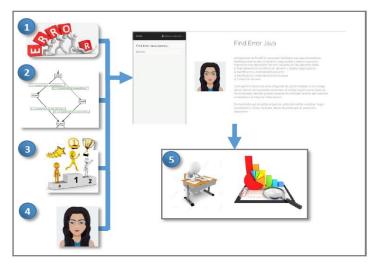


Fig. 1. Elementos que fundamentan el desarrollo de Find Error Java.

A continuación, se presenta una breve descripción de los elementos clave.

1. Identificación de los errores comunes que cometen los estudiantes novatos al aprender a programar y creación de banco de ejercicios.

- 2. Estrategia de instrucción de depuración de errores mediante la adaptación a la técnica de seguimiento de ejemplos (*example-tracing*).
- 3. Dinámicas y mecánicas de gamificación que ayuden a motivar al estudiante a la reflexión.
- 4. Agente pedagógico Lucy [3] integrado a la estrategia de instrucción basada en ejemplos erróneos.
- 5. Plan de pruebas experimentales mediante evaluaciones de usabilidad y ganancias de aprendizaje.

3.2. Proceso de instrucción

Partiendo de los errores comunes identificados, se realizó un Análisis de Tareas Cognitivas (CTA por sus siglas en inglés) que permitió identificar los diferentes recursos cognitivos que un experto utiliza consciente o inconscientemente durante la depuración de estos errores, con el fin de definir los lineamientos y el proceso de instrucción para la depuración.

Algunos de los lineamientos son:

- 1. Presentar conceptos básicos específicos para entender el funcionamiento del lenguaje o un error en particular, con el fin de ofrecer una solución.
- 2. Orientar al estudiante sobre el significado de los mensajes que arroja el compilador y la máquina virtual de Java según el tipo de error encontrado.
- 3. Para errores lógicos, ofrecer al estudiante orientación que le permita entender o interpretar el error con base en las entradas o salidas del programa.
- 4. Durante la realización de todos los ejercicios ofrecer al estudiante orientación sobre la diferencia entre errores sintácticos, semánticos o lógicos.

El proceso de instrucción para la depuración se definió con base a los principios cognitivos, los errores comunes detectados y los lineamientos, con el fin de promover el aprendizaje meta cognitivo al propiciar que el estudiante realice un razonamiento consciente sobre cada una de las fases del proceso de depuración. Además se utilizó un enfoque constructivista en el sentido que los ejercicios consideran conocimiento previo el cual pudiera ser impreciso.

El proceso de instrucción consta de 4 pasos:

- 1. Notificación del error (instrucciones del ejercicio).
- 2. Identificación del error en el código (línea o fragmento de código erróneo).
- 3. Entendimiento de la causa (conceptos, reglas o principios relacionados).
- 4. Corrección del error (instrucciones correctas).

3.3. Arquitectura del ambiente inteligente

Para el diseño del sistema se eligió el modelo arquitectónico en capas que permite una organización clara e independiente de los componentes (ver figura 2).

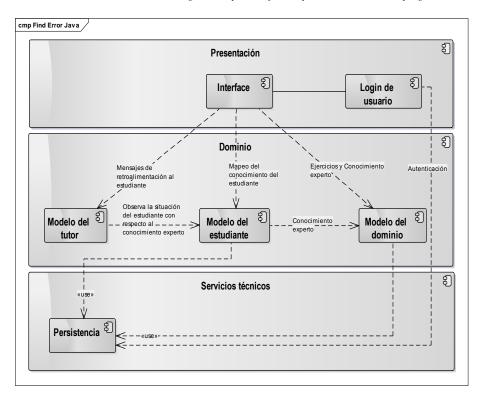


Fig. 2. Arquitectura lógica del sistema Find Error Java.

Capa de presentación. Contiene las interfaces con las que el usuario va a interactuar con el sistema, tales como registro de usuario, iniciar sesión, menú principal, realización de ejercicios y consulta de puntuación.

Capa de dominio. Integrada por 3 componentes principales: Modelo del tutor, Modelo del estudiante y Modelo del Dominio. Este último contiene las representaciones del conocimiento experto respecto a los ejercicios, estrategia de depuración, conceptos y procesos de pensamiento que deben transferirse al estudiante. Este componente tiene la responsabilidad de implementar una adaptación de la técnica de seguimiento de ejemplos (example-tracing) para procesar un grafo de comportamiento que incluye acciones incorrectas que pudiera cometer un estudiante durante el proceso de instrucción en depuración; así como también comportamiento óptimo, mismo que se pretende transferir al estudiante. Las acciones o comportamiento sub-óptimo de la técnica original, no fueron consideradas ya que el objetivo es mostrar un ejemplo de depuración de errores de manera óptima. El grafo de comportamiento experto del modelo del dominio contiene la estrategia de instrucción en la que se conduce al estudiante a un razonamiento consciente sobre el entendimiento de la notificación del error (instrucciones), identificación del error en el código, entendimiento de la causa y corrección del error. La figura 3 presenta el grafo de comportamiento experto utilizado por el modelo del dominio.

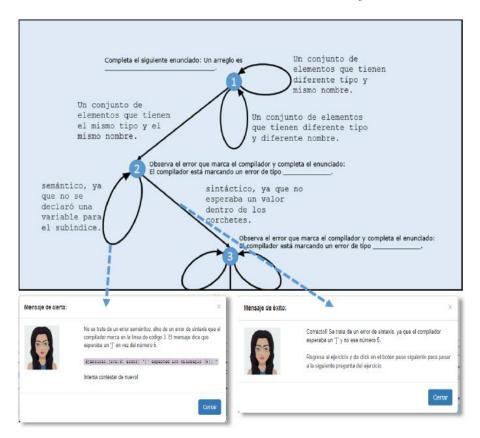


Fig. 3. Grafo de comportamiento experto.

Capa de servicios técnicos. Se encarga de realizar la lectura y almacenamiento de la información. Está conformada por modelos que el ORM de laravel (llamado Eloquent) permite utilizar. Los modelos tienen una correspondencia con cada una de las tablas que integran la base de datos de *FindErrorJava* y son los encargados de realizar la conexión a la base de datos y ejecutar las consultas de lectura y actualización.

3.4. Principales componentes del ambiente inteligente

En esta sección se describen los principales componentes con los que interactúa el usuario en el ambiente de aprendizaje FindErrorJava.

Interfaz de inicio. En la Figura 4 se presenta la interfaz de inicio, donde el agente pedagógico Lucy explica brevemente el proceso de instrucción que siguen los ejercicios. Se presenta el menú principal, con las ligas a los ejercicios que el estudiante puede realizar en orden ascendente, de menor a mayor grado de complejidad.



Fig. 4. Interfaz de inicio.

Interfaz de ejercicios. Esta es la interfaz principal del sistema donde el estudiante realiza los ejercicios de depuración de errores de programación y donde se combinan diferentes técnicas de instrucción con el objetivo de darle tutoría al estudiante para que no solo aprenda a corregir el tipo de error que contiene el código del ejercicio sino también que adquiera conceptos, principios, razonamientos y habilidades que le servirán incluso para resolver otro tipo de errores. Un ejercicio puede ser resuelto en *n* pasos, donde *n* puede ser mayor o igual que 1.

Un ejercicio está conformado por:

- a) Título del ejercicio
- b) Indicaciones
- c) Código del ejercicio
- d) Mensajes del compilador
- e) Mensajes de la Máquina Virtual de Java (MVJ)
- f) Pregunta o petición de interacción
- g) Acciones optativas

En la Figura 5 se puede observar la correspondencia entre los atributos descritos y la interfaz de ejercicios.

Agente pedagógico. El agente pedagógico Lucy se utilizó para realizar intervenciones con el estudiante durante la realización de los ejercicios. Las intervenciones del agente pedagógico pueden ser de 3 tipos:

- 1) Lucy proporciona pistas bajo demanda.
- 2) Retroalimentación a nivel paso: Lucy ofrece una retroalimentación inmediatamente después de que el estudiante comete algún error, con el fin de intentar corregir la idea errónea que haya provocado su respuesta incorrecta (ver Figura 6) en el momento justo cuando ocurrió

3) Intervención por notificación: Cuando el agente pedagógico considera que el estudiante no está interesado en contestar correctamente, le envía una notificación cuyo objetivo es motivarlo a leer las pistas y realizar reflexiones más profundas. Los mensajes de notificación enviados no siempre son los mismos.

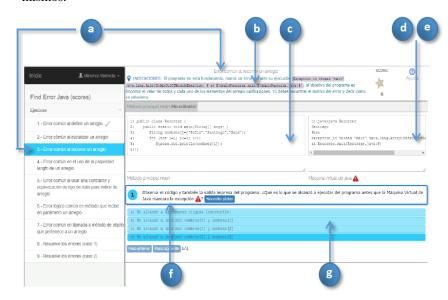


Fig. 5. Interfaz de ejercicios.



Fig. 6. Retroalimentación inmediata respuesta incorrecta del estudiante.

4. Pruebas

En esta sección se presentan las evaluaciones realizadas al sistema con el fin de dictaminar su eficacia y medir el grado de aceptación por parte de los usuarios.

4.1. Población y planificación de las pruebas

El estudio se realizó en el Instituto Tecnológico de Culiacán, y participaron 46 estudiantes (9 mujeres y 37 varones) en un rango de 18-21 años, pertenecientes a las carreras Ingeniería en Sistemas Computacionales e Ingeniería en Tecnologías de la Información y Comunicaciones.

La lista de actividades de este estudio se muestra en la tabla 1.

Tabla 1. Listado de actividades para la realización del estudio.

Orden	Actividad
1	Selección de la muestra (estudiantes que participan en el estudio).
2	Distribución de la muestra en dos grupos: experimental y control.
3	Aplicación de examen Pre-Test a ambos grupos.
4	Utilización de Find Error Java en grupo experimental.
5	Aplicación de encuesta TAM a grupo experimental.
6	Aplicación de examen Post-Test a ambos grupos.

4.2. Efectividad y ganancias de aprendizaje significativo en depuración de errores

Para la evaluación de la efectividad en ganancia de aprendizaje, se decidió abordar con *Find Error Java* el tema de arreglos, ya que este tópico es generalmente un problema para los estudiantes durante los cursos básicos de programación, persistiendo estos problemas incluso en cursos posteriores a los básicos.

Para medir y comparar la ganancia de aprendizaje entre alumnos que usan *Find Error Java* y los que no lo usan, la muestra de 46 estudiantes se dividió en dos grupos, el primero denominado grupo experimental y el segundo denominado grupo de control.

A los 46 estudiantes, previo a la utilización de *Find Error Java*, se les aplicó una primera evaluación denominada Pre-Test relacionada a la depuración de errores. Posteriormente, se realizó una sesión de 45 minutos con el grupo experimental para que resolvieran los ejercicios de depuración de errores utilizando *Find Error Java* y para finalizar, se aplicó a ambos grupos una segunda evaluación denominada Post-Test.

En la Figura 7 se presenta un comparativo de las calificaciones obtenidas en el Pre-Test y Post-Test correspondientes al grupo experimental, el cual utilizó *Find Error Java*. La mayoría de los estudiantes (78.2%) del grupo experimental, presentaron un incremento en su calificación en el examen posterior al uso de la herramienta.

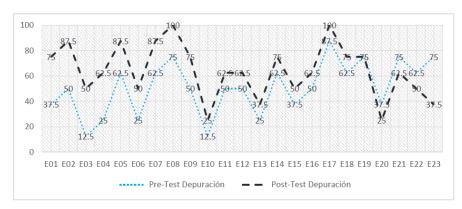


Fig. 7. Comparativo grupo experimental en relación a la depuración.

En la Figura 8 se presenta un comparativo de las calificaciones obtenidas en el Pre-Test y Post-Test correspondientes al grupo de control, el cual no utilizó *Find Error Java*. En este caso fue el 26% del grupo de control, el que presentó un incremento en su calificación.

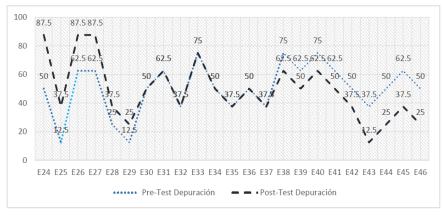


Fig. 8. Comparativo grupo de control en relación a la depuración.

5. Conclusiones y trabajo a futuro

Los errores de programación pueden utilizarse naturalmente como ejemplos erróneos, los cuales a su vez ofrecen una gran oportunidad de aprendizaje cuando son bien utilizados en el desarrollo de temas y actividades de aprendizaje. El uso de ejemplos erróneos genera conflicto cognitivo y siembran curiosidad en el estudiante (animan a la reflexión e investigación).

Find Error Java implementa un ambiente de aprendizaje para la depuración de errores que integra a la estrategia de instrucción de un agente pedagógico y elementos de gamificación. El objetivo de la gamificación fue motivar al estudiante para que intentara hacer reflexiones profundas antes de responder cada paso de los ejercicios;

además se requería propiciar un ambiente de competencia con la finalidad de ofrecer diversión.

Los resultados obtenidos en los experimentos donde se midió la efectividad en ganancias de aprendizaje mediante evaluaciones de conocimiento Pre-Test y Post-Test demuestran que el uso de una herramienta como *Find Error Java*, puede ser de gran utilidad para que los estudiantes de programación aprendan a depurar los errores del código que se generan cuando se diseña un programa para resolver un problema.

Como trabajo futuro se considera realizar más evaluaciones con temas diferentes del lenguaje Java. También buscaremos implementar el ambiente con otros lenguajes como Python y otros lenguajes en el campo de pensamiento computacional donde el enfoque es hacia lenguajes visuales orientados a niños y jóvenes.

Referencias

- Jenkins, T.: On the difficulty of learning to program. In: Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences, 4(2002), pp. 53– 58 (2002)
- Murphy, L., Lewandowski, G., McCauley, R., Simon, B., Thomas, L., Zander, C.: Debugging: the good, the bad, and the quirky - a qualitative analysis of novices' strategies. In: ACM SIGCSE Bulletin 40(1), pp. 163–167 (2008)
- Sosa, C.: Agente Pedagógico con estrategia "Aprendiz Cognoscitivo" para un ambiente de programación colaborativo. Tesis de maestría en Instituto Tecnológico de Culiacán (2016)
- 4. Borasi, R.: Capitalizing on errors as "springboards for inquiry": A teaching experiment. Journal for Research in Mathematics Education, pp. 166–208 (1994)
- 5. Ginat, D., Shmalo, R.: Constructive use of errors in teaching CS1. In: Proceeding of the 44th ACM Technical Symposium on Computer Science Education, pp. 353–358 (2013)
- Aleven, V., Mclaren, B., Sewall, J., Koedinger, K.: A New Paradigm for Intelligent Tutoring Systems: Example-Tracing Tutors. International Journal of Artificial Intelligence in Education 19(2), pp. 105–154 (2009)
- 7. Vanlehn, K.: The behavior of tutoring systems. International Journal of Artificial Intelligence in Education 16(3), pp. 227–265 (2006)
- 8. Chorney, A.: Taking the Game Out of Gamification. Dalhousie Journal of Interdisciplinary Management, 8(1) (2012)
- 9. Stott, A., Neustaedter, C.: Analysis of gamification in education. Surrey, BC, Canada, 8(36) (2013)
- 10. Ahmadzadeh, M., Elliman, D., Higgins, C.: An analysis of patterns of debugging among novice computer science students. In: ACM SIGCSE bulletin 37(3), pp. 84–88 (2005)
- 11. Cross II, J. H., Hendrix, T. D., Barowski, L. A.: Combining dynamic program viewing and testing in early computing courses. In: Proceeding of IEEE 35th Annual Computer Software and Applications Conference (COMPSAC), pp. 184–192 (2011)
- 12. Lawrance, J., Bogart, C., Burnett, M., Bellamy, R., Rector, K., Fleming, S. D.: How programmers debug, revisited: An information foraging theory perspective. IEEE Transactions on Software Engineering 39(2), 197–215 (2013)
- Jackson, J., Cobb, M., Carver, C.: Identifying Top Java Errors for Novice Programmers.
 In: Proceedings of 35th Annual Conference Frontiers in Education (FIE 2005), pp. 24–27 (2005)
- McCall, D., Kölling, M.: Meaningful Categorisation of Novice Programmer Errors. In: Proceedings of 2014 Frontiers in Education Conference (FIE 2014), pp. 2589–2596 (2014)

María Lucia Barrón-Estrada, Ramón Zatarain-Cabada, Minerva Valencia-Rodríguez, et al.

- 15. Woolf, B. P.: Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning. Morgan Kaufmann (2010)
- 16. Brusilovsky, P., Pesin, L., Zyryanov, M.: Towards an adaptive hypermedia component for an Intelligent Learning Environment. In: Proceedings of International Conference on Human-Computer Interaction, Springer, Berlin, Heidelberg. pp. 348–358 (1993)
- 17. Deterding, S., Dixon, D., Khaled, R., Nacke, L.: From game design elements to gamefulness: defining "gamification". In: Proceedings of the 15th International Academic MindTrek Conference: Envisioning future media environments, pp. 9–15 (2011)
- 18. Malone, T.: Making Learning Fun: A taxonomy of intrinsic motivations for learning. Aptitude, learning, and instruction 3, pp. 223–253 (1987)
- 19. Gee, J.: What Video Games Have to Teach Us About Learning and Literacy? Computers in Entertainment, 2nd ed., Palgrave Macmillan, USA (2007)

Optimización basados en caos para la búsqueda de objetivos

Karla I. Fernandez-Ramirez, Arturo Baltazar

CINVESTAV Unidad-Saltillo, Coahuila, México k.ivonne.fernandez.rmz@gmail.com, arturo.baltazar@cinvestav.edu.mx

Resumen. En este trabajo se discute un algoritmo propuesto para la búsqueda óptima de un objetivo basado en la teoría del caos. La propuesta se basa en dos etapas; la primera etapa realiza una búsqueda en toda el área de trabajo mediante secuencias caóticas para encontrar el mayor número posible de mínimos o máximos locales. Existen diferentes sistemas que generan secuencias caóticas, en este trabajo se utilizaron los mapas: logístico y de Kent. Los resultados obtenidos muestran un mejor desempeño con el mapa de Kent. La segunda etapa utiliza el último valor óptimo detectado en la primera etapa como punto inicial para comenzar la búsqueda del óptimo global utilizando un camino caótico. El algoritmo fue probado en diferentes funciones conteniendo múltiples mínimos locales. Se realizó una combinación de secuencias aleatorias y caóticas, observando que el mejor desempeño es utilizando un camino caótico en la segunda etapa. Los resultados muestran que el esquema propuesto puede ubicar correctamente la posición del óptimo global con un tiempo de cálculo menor que los otros métodos de optimización estudiados.

Palabras clave: optimización, caos, búsqueda.

On Chaos Optimization for Target Search

Abstract. In this paper we discuss a proposed algorithm for the optimal target search based on chaos theory. The proposal is based on two stages; the first stage performs a search in the entire work area using chaotic sequences to find as many minimums or maximums local as possible. There are different systems that generate chaotic sequences, in this work the maps were used: logistic and Kent. The obtained results show a better performance with the Kent map. The second stage uses the last optimum value detected in the first stage as an initial point to begin the search for the global optimum using a chaotic sequence. The algorithm was tested in different functions containing multiple local minimums. A combination of random and chaotic sequences was performed, observing that the best performance is using a chaotic path in the second stage. The results show that the proposed scheme can correctly position the global optimum position with a short calculation time than other deterministic, stochastic and chaotic optimization methods.

Keywords: optimization, chaos, search.

1. Introducción

La teoría de cómo encontrar una estrategia para la búsqueda de un objetivo o una fuente de perturbación, fija o en movimiento, ha sido estudiada por científicos durante más de 50 años. En [1], se presenta la teoría de búsqueda desarrollada durante la Segunda Guerra Mundial para la Marina de los Estados Unidos. En esa teoría, el escaneado sistemático se utiliza comúnmente como una técnica de búsqueda [2]. Se han encontrado aplicaciones manuales o automáticas de esta técnica, por ejemplo, en astronomía, en la industria, medicina, biología y exploración minera.

Se puede ver que, si la resolución del paso de búsqueda es lo suficientemente pequeña y se conoce el espacio de trabajo, es probable que se alcance una búsqueda completa del área de trabajo. Sin embargo, tal esquema puede consumir mucho tiempo y potencia computacional. El problema de buscar un objetivo estacionario o no estacionario por un robot móvil también ha ganado considerable atención por parte de la comunidad científica para su posible aplicación en tareas de búsqueda como en operaciones militares, de vigilancia o de rescate [3]. La dinámica del Caos ha sido recientemente reportada como una herramienta para explorar zonas de vigilancia con una cobertura completa y rápida del espacio de trabajo [4] utilizando sistemas robóticos. El problema de búsqueda puede interpretarse como un problema de optimización en el que se debe encontrar el máximo o mínimo en un mapa de distribución.

Los algoritmos de optimización basados en caos (COA, por sus siglas en inglés) se basan en dos etapas, la primera se utiliza secuencias caóticas para producir una variable de búsqueda global. La segunda etapa realiza una búsqueda refinada considerando que la localización inicial de búsqueda se encuentra cerca del óptimo global. El COA se ha estudiado utilizando secuencias numéricas generadas por un mapa caótico [5]. En [6] demostraron que las propiedades estadísticas de los mapas caóticos afectan la eficiencia de los algoritmos en la búsqueda global, mientras que en [7] muestran que el éxito del algoritmo depende de la segunda etapa.

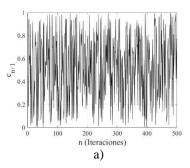
En este trabajo, desarrollamos un esquema de búsqueda de objetivos que utiliza la teoría del caos para su optimización. Los objetivos son los siguientes; primero se estudiará el efecto de utilizar una distribución caótica y una distribución aleatoria uniformemente distribuida en la primera etapa; segundo, se propone una modificación de la segunda etapa del COA, utilizando el camino caótico utilizado en [8]. El algoritmo propuesto se probará numéricamente usando funciones analíticas con más de un mínimo local. El objetivo es estudiar la robustez y rapidez del algoritmo propuesto para localizar el óptimo global.

2. Optimización aleatoria y caótica

Los algoritmos de optimización estocástica no dependen de una propiedad matemática estricta, como continuidad, diferenciabilidad o una descripción matemática exacta de las funciones objetivo y las condiciones de restricción, y por lo tanto sus procedimientos de optimización son fáciles de implementar [6]. Sin embargo, estos

algoritmos generalmente muestran una convergencia prematura y una capacidad de explotación débil, lo que lleva a un óptimo local en lugar de un óptimo global y, por lo tanto, una convergencia lenta. Alternativamente, el caos se ha propuesto como un algoritmo de optimización debido a sus propiedades topológicas y estadísticas. El caos ocurre en muchos sistemas no lineales, donde su evolución en el tiempo es impredecible. Para clasificar el comportamiento de un sistema como caótico, debe tener ciertas propiedades: sensibilidad a las condiciones iniciales, pseudoaleatoriedad y ergodicidad que implica que una secuencia caótica puede visitar todo el estado del atractor extraño [9]. Estas propiedades que se encuentran en el algoritmo de optimización del caos podrían tener un mejor rendimiento que los algoritmos estocásticos [10].

La Fig. 1 muestra la secuencia de una variable aleatoria uniformemente distribuida y su respectivo histograma para los primeros 500 valores. Los resultados de esta pequeña muestra evidencian la naturaleza de esta distribución.



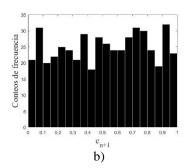


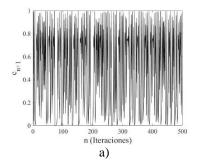
Fig. 1. a) Distribución aleatoria uniforme; b) histograma de la secuencia aleatoria.

Un mapa de una dimensión (1D) que puede generar una secuencia caótica es el mapa logístico. Este es un mapa polinomial que puede conducir a fenómenos complicados de dinámica caótica y se puede representar como se muestra en la ecuación (1):

$$c_{n+1} = \mu c_n (1 - c_n), \quad 0 < \mu \le 4, \quad c_n \in (0,1),$$
 (1)

donde c_n es el n-ésimo número caótico, n representa el número de iteraciones y μ es el parámetro de control. El sistema (1) tiene diferentes características dinámicas para diferentes valores de μ . Cuando el valor de $\mu=4$, se genera la secuencia caótica entre valores de (0,1). Los resultados de la simulación utilizando $\mu=4$ se muestran en la Fig. 2.

La función de densidad de probabilidad (PDF) es una distribución de Chebyshev [11], donde el centro de la función muestra una distribución uniforme, como se puede observar en el histograma de la secuencia caótica (Fig. 2 b).



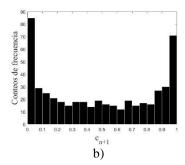
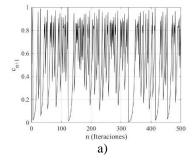


Fig. 2. a) Secuencias caóticas generadas por el mapa logístico con $\mu=4$; b) histograma de la secuencia caótica.

Se ha reportado que el mapa logístico puede tener algunas limitaciones debido a su distribución no uniforme [11]. Una alternativa es el mapa de Kent, que se ha sido utilizado en aplicaciones como el cifrado seguro y tiene una distribución teórica uniforme PDF. El mapa definido en la ecuación (2):

$$c_{n+1} = \begin{cases} \frac{c_n}{m} & 0 < c_n \le m \\ \frac{1-x_n}{1-m} & m < c_n \le 1 \end{cases}$$
 (2)

donde m es un parámetro de control con un intervalo de 0 < m < 1, y la secuencia está acotada entre (0,1) [11]. La Fig. 3 muestra un ejemplo de la secuencias generadas cuando m=.8, también se muestra el histograma correspondiente. La distribución obtenida exhibe un patrón que se asemeja a una distribución uniforme.



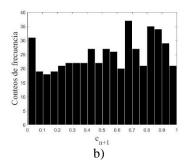


Fig. 3. a) Secuencia caótica generada por el mapa de Kent; b) histograma de la secuencia caótica.

Las secuencias caóticas en un espacio de trabajo en 2D de los dos mapas presentados se comparan con una distribución uniforme aleatoria. En las Fig. 4 y Fig. 5 muestran los resultados para 500 y 1000 iteraciones, respectivamente.

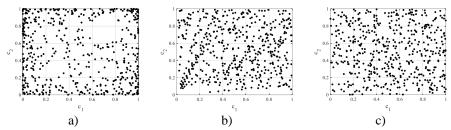


Fig. 4. Distribución en dos dimensiones con 500 iteraciones; a) mapa logístico; b) mapa de Kent; c) distribución uniforme aleatoria.

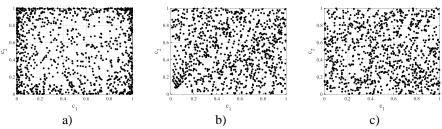


Fig. 5. Distribución en dos dimensiones con 1000 iteraciones; a) mapa logístico; b) mapa de Kent; c) distribución uniforme aleatoria.

Claramente, las distribuciones obtenidas usando mapas caóticos proporcionan una distribución casi uniforme similar a la distribución aleatoria uniforme con la ventaja de que, de acuerdo con las propiedades topológicas de la teoría del caos, todos los conjuntos abiertos en el mapa pueden ser visitados [11]. Más adelante, se probarán estos mapas con funciones seleccionadas para estudiar el desempeño para la búsqueda del máximo global.

3. Algoritmo de optimización propuesto basado en la teoría del caos

El algoritmo de optimización de caos propuesto se basa en dos etapas que se describe de manera general en la Fig. 6.



Fig. 6. Diagrama del algoritmo propuesto.

El algoritmo se describe a continuación:

ISSN 1870-4069

1. Se define los parámetros de inicialización: número de iteraciones n, límite inferior L_i y superior U_i donde i representa las dimensiones de la función a optimizar. Se

define un valor óptimo $f^* = \infty$ (si se busca el mínimo global). Se establecen los valores iniciales c_i^k (k = 1, ... n) para generar las secuencias caóticas o aleatorias.

2. Las variables caóticas c_i^k (acotadas entre (0,1)) se mapean en el rango de los límites establecidos (variables de optimización x_i^k) mediante la ecuación (3)

$$\chi_i^k = L_i + (U_i - L_i)c_i^k. \tag{3}$$

3. La función es evaluada con las variables de optimización $f(\vec{x}^k)$. Si el valor de la función $f(\vec{x}^k) < f^*$ entonces $f^* = f(\vec{x}^k)$ y la solución óptima $\vec{x}^* = \vec{x}^k$. La siguiente variable es generada c_i^{k+1} hasta k > n. A continuación, se muestra el seudocódigo de la primera etapa.

01	Inicialización n, L_i , U_i , f^* , $c_i^{k=1}$
02	Mientras $n \ge k$ entonces
03	$c_i^k = \mu c_i^{k-1} \left(1 - c_i^{k-1}\right)$
04	$x_i^k = L_i + (U_i - L_i)$
05	Si $f(\vec{x}^k) < f^*$ entonces
06	$f^*=f(\vec{x}^k)$
07	$ec{x}^* = ec{x}^k$
08	Fin
09	k = k + 1
10	Fin de la primera etapa

- 4. La segunda etapa inicializa con los últimos valores óptimo encontrado en la primera etapa. Se define el número de iteraciones n y los valores iniciales $\theta_i^{k=1}$ y $\zeta^{k=1}$.
- 5. En esta etapa se utiliza la ecuación de Pearson (4) definida por:

$$x_i^k = \sum_{k=1}^n \zeta^k e^{i\theta_i^k},\tag{4}$$

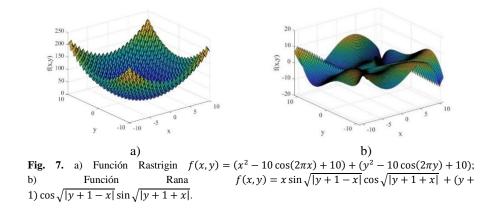
donde x_i^k es la nueva posición, que se da como una suma de n vectores de dos dimensiones con un tamaño de paso (ζ^k) de la iteración k y orientación (θ_i^k) . El valor de ζ^k y θ_i^k están dado por el mapa caótico. Los valores de θ_i^k son mapeados dentro del rango $(0,2\pi)$ utilizando la ecuación (3).

- 6. Se ajustan las variables dentro del rango, si $x_i^k < L_i$ entonces $x_i^k = L_i$; $x_i^k > U_i$ entonces $x_i^k = U_i$.
- 7. La función es evaluada con las variables de optimización $f(\vec{x}^k)$. Si el valor de la función $f(\vec{x}^k) < f^*$ entonces $f^* = f(\vec{x}^k)$ y la solución óptima $\vec{x}^* = \vec{x}^k$. La siguiente variable es generada c_i^{k+1} hasta k > n.

```
01 Inicialización n, L_i, U_i, f^*, \theta_i^{k=1}, \zeta^{k=1}
02 Mientras n \ge k entonces
03
        \theta_i^k = \mu \theta_i^k (1 - \theta_i^{k-1})
        \zeta^k = \mu \zeta^{k-1} (1 - \zeta^{k-1})
04
         x_i^k = \sum_{k=1}^N \zeta^k \, e^{i\theta_i^k}
05
06
         Si x_i^k < L_i entonces
07
             x_i^k = L_i
08
09
         Si x_i^k < U_i entonces
10
             x_i^k = U_i
11
         Si f(\vec{x}^k) < f^* entonces
12
13
             f = f(\vec{x}^k)
             x*
14
15
        Fin
16
       k = k + 1
17 Fin de la segunda etapa
```

4. Resultados numéricos

Para probar la eficiencia y el comportamiento del algoritmo propuesto, se evaluó utilizando funciones conocidas: la función Rastrigin (Fig. 7 a)) y la función Rana (Fig. 7 b)). La función Rastrigin tiene varios mínimos locales y un mínimo global en la posición $x^* = (0,0)$, and $f^* = 0$. La función Rana tiene varios mínimos locales y un mínimo global en el punto $x^* = (11,10)$, and $f^* = -11$.



En la Fig. 8 muestra un ejemplo de las etapas del algoritmo propuesto en la función Rastrigin, utilizando 200 iteraciones en la primera etapa y 100 en la segunda. El valor óptimo encontrado fue 1.03, tomando un tiempo computacional de .0313 seg.

297

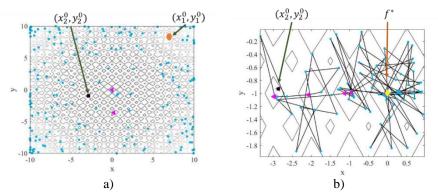


Fig. 8. Proyección de la función Rastrigin (plano x-y) donde (x_1^0, y_1^0) es el inicio de la primera etapa y (x_2^0, y_2^0) es el inicio de la segunda; a) Primera etapa; b) Segunda etapa.

La Tabla 1 muestra los resultados (tiempo computacional y valor óptimo) después de implementar el algoritmo propuesto. Los resultados muestran que los dos casos estudiados dan un valor óptimo general muy cercano al mínimo global verdadero de la función con solo una ligera mejoría para el caso 1.

Para comparar el algoritmo propuesto, se realizó una comparación con el algoritmo de gradiente descendiente y con el algoritmo de perturbación simultánea con aproximación estocástica (SPSA) partiendo de la misma posición y con el mismo número de iteraciones. Los resultados obtenidos muestran un mejor desempeño de estos dos algoritmos comparado con el caso tres del algoritmo propuesto, pero menor que los dos primeros casos. El algoritmo del gradiente puede mejorar el desempeño modificando el paso que se utilizó o usando un número mayor de iteraciones. El algoritmo del SPSA se basa en variables aleatorias, por lo que puede encontrar el valor óptimo incrementando el número de iteraciones.

Tabla 1. Resultados de la simulación de la función Rastrigin con el mapa logístico (N=300).

# C	Al	goritmo	\$7-1 (4*	TD:	
# Casos	1er etapa 2da etapa		Valor óptimo	Tiempo (seg.)	
1	Caos	Camino caótico	1.01	.0313	
2	Aleatorio	Camino caótico	1.57	.0313	
3	Caos	Camino aleatorio	9.03	.0156	
4	Gradiente (h=.025)		6.60	.0312	
5	SPSA		5.13	.0156	

Los tres casos propuestos se corrieron 50 veces para adquirir las métricas estadísticas mostradas en la Tabla 2 y en la Tabla 3 para las funciones Rastrigin y Rana, respectivamente. En promedio, el algoritmo propuesto usando secuencias caóticas en las dos etapas da mejores resultados que cuando se combina con secuencias aleatorias.

En la Tabla 4 muestra los resultados de implementar el algoritmo propuesto en la función de Rastrigin utilizando el mapa de Kent para generar la secuencia caótica. Los resultados muestran que los dos casos estudiados dan un valor óptimo general muy

cercano al verdadero mínimo global de la función con solo una ligera mejoría para el caso 1.

Tabla 2. Resultados de simulación de la función Rastrigin con el mapa logístico.

# Casos	V	alor ópti	mo	Tiempo computacional		
	Mejor	Peor	Promedio	Mejor	Peor	Promedio
1	4.03E-06	0.0009	0.0005	0.093	0.281	0.184
2	2.92E-06	2.16	0.5142	0.015	0.093	0.05
3	6.87E-06	0.0291	0.0043	0.046	0.187	0.092

Tabla 3. Resultados de simulación de la función Rana con el mapa logístico.

# C	Valor óptimo			Tiempo computacional		
# Casos	Mejor	Peor 1	Promedio	Mejor	Peor	Promedio
1	-11	-11	-11	.0156	.1093	.0409
2	-11	-11	-11	.0156	.125	.0596
3	-11	-10	-10	.0156	.125	.0415

Tabla 4. Resultados de la simulación de la función Rastrigin con el mapa Kent (N=300).

# Casos	Al	goritmo	Volon ándino	Tiempo (seg.)	
	1er etapa	2da etapa	Valor óptimo		
1	Caos	Camino caótico	.0063	.1250	
2	Aleatorio	Camino caótico	1.82	.0313	
3	Caos	Camino aleatorio	2.12	.0313	

5. Conclusiones

En el presente artículo, se propuso un algoritmo de optimización utilizando secuencias caóticas. El algoritmo se basa en dos etapas. En la primera etapa, se lleva a cabo una búsqueda de los posibles valores óptimos (máximos o mínimos). La segunda etapa comienza en el último óptimo encontrado durante la primera etapa y se realiza una búsqueda utilizando la caminata caótica.

La metodología propuesta se comparó con diferentes esquemas, combinando secuencias aleatorias y caóticas en las dos funciones. El mapa logístico y el mapa de Kent se usaron para generar las secuencias caóticas, que tienen diferentes distribuciones. Los resultados muestran que hay una ligera mejora en el tiempo computacional para el esquema propuesto. Los resultados presentados solo consideran funciones de dos dimensiones; sin embargo, como trabajo futuro se tiene contemplado atacar el problema en *n*-dimensiones y la implementación del esquema de optimización propuesto para la búsqueda de un objetivo.

Agradecimientos. Los autores agradecen al CONACYT por el apoyo económico para Karla I. Fernandez-Ramirez (estudiante de doctorado) y a través del proyecto CB-286907.

Referencias

- 1. Koopman, B.: Search and Screening: General principles with historical applications. Pergamon Press, New York (1980)
- 2. Gertner, I., Zeevi, Y. Y.: Scanning strategies for target detection. In: Proc. SPIE, vol. 1470, pp. 1–19. Data Structures and Target Classification (1991)
- 3. Benkoski, S., Monticino, M., Weisinger, J.: A survey of the search theory literature. Naval Research Logistic (39), pp. 469–494 (1991)
- 4. Martins-Filho, L. S., Macau, E. E.: Autonomous Robots and Agents, Springer Berlin Heidelberg, Berlín, Heidelberg (2007)
- 5. Bing, L., Weisun, J.: Chaos optimization method and its application. Control Theory and Application, pp. 613–615 (1997)
- 6. Yang, D., Liu, Z., Zhou, J.: Chaos optimization algorithms based on chaotic maps with different probability distribution and search speed for global optimization. Communications in Nonlinear Science and Numerical Simulation (19), pp. 1129–1246 (2014)
- 7. Velásquez, J. D.: An introduction to chaos-based algorithms for numerical optimization. Recista Avance en Sistema e Informática (8), pp. 51–60 (2011)
- 8. Baltazar, A., Fernandez-Ramirez, K.I., Aranda-Sanchez, J.I.: A study of chaotic searching for their application in an ultrasonic scanner. Engineering Applications of Artificial Intelligence, Unplublished (2018)
- 9. Yan, X. F., Chen, D. Z., Hu, S. X.: Chaos-genetic algorithms for optimizing the operating conditions based on RBF–PLS model. Computers & Chemical Engineering (27), pp. 1393–1404 (2003)
- 10. Yang, D., Li, G., Cheng, G.: On the efficiency of chaos optimization algorithms for global optimization. Chaos, Solitons Fractals (34), pp. 1366–1375 (2007)
- 11. Baker, G., Gollud, J.: Chaotic dynamics: An introduction, 2nd edn. Cambridge University Press, New York (1996)

Upper Confidence Bound o Upper Confidence Bound Tuned para General Game Playing: Un estudio empírico

Iván Francisco-Valencia, José Raymundo Marcial-Romero, Rosa María Valdovinos-Rosas

Universidad Autónoma del Estado de México, Facultad de Ingeniería, Toluca, Estado de México, México

ifranciscov196@alumno.uaemex.mx, {jrmarcialr,rvaldovinos}@uaemex.mx

Resumen. En este artículo se muestra un análisis comparativo de dos políticas de selección: Upper Confidence Bound (UCB) y Upper Confidence Bound Tuned (UCB-Tuned), para el método Monte Carlo Tree Search (MCTS), en el contexto de General Game Playing (GGP). Este análisis tiene como objetivo determinar cual de las políticas tiene el mejor rendimiento, en términos de victorias. Para el comparativo se desarrollaron dos agentes basados en estas políticas de selección usando el framework GGP-Base. Así mismo los agentes se limitaron a 100 simulaciones de juego por cada iteración del método MCTS. El análisis se realizó en tres etapas, en cada una de las cuales, los agentes compitieron en 1000 juegos de tablero Breakthrough y 1000 de Knightthrough. Los resultados muestran que el agente con mejor rendimiento en GGP, es el agente basado en la política UCB-Tuned, ya que en promedio, resulta ganar el 60,8 % de las veces en Breakthrough y el 78,1 % en Knightthrough.

Palabras clave: jugar juegos en general, problema del bandido multiarmado, políticas de selección.

Upper Confidence Bound or Upper Confidence Bound Tuned for General Game Playing: An Empirical Study

Abstract. This paper shows a comparative analysis of two selection policies: Upper Confidence Bound (UCB) and Upper Confidence Bound Tuned (UCB-Tuned), for the Monte Carlo Tree Search method (MCTS), in the context of General Game Playing (GGP). This analysis aims to determine which of the policies has the best performance, in terms of victories. For the comparison two agents were developed based on these selection policies using the GGP-Base framework. The agents were limited to 100 simulations of game for each iteration of MCTS method. The analysis has been done on three stages, in each of which the agents play in 1000 Breakthrough games and 1000 Knighthrough games. The results show that the agent with the best performance in GGP, is the agent based on UCB-Tuned police, since in average, it turns out to win

the 60,8% of times in Breakthrough and 78,1% in Knightthrough.

Keywords: general game playing, multi-armed bandit problem, selection policies.

1. Introducción

Desde sus inicios la Inteligencia Artificial (IA) ha tenido como uno de sus retos el desarrollo de agentes inteligentes capaces de jugar juegos de tablero al mismo nivel (o superior) que el ser humano [4]. El ánimo de desarrollar agentes jugadores se debe a que éstos presentan características propias de la inteligencia humana como deducción, razonamiento, resolución de problemas, búsqueda inteligente, representación del conocimiento, planificación, aprendizaje, creatividad, percepción y procesamiento de lenguaje natural, entre otros [20]. En este sentido, la IA ha producido agentes capaces de jugar a nivel de campeones humanos en juegos específicos como Chinook para Damas [18](juego para el cual ya existe una estrategia que permite ganar sin importar las jugadas que realice el contrincante [17]), Deep Blue para el ajedrez [6] y Dark Knight para Banqi o ajedrez medio chino [13].

El área de la IA que se ha enfocado en desarrollar agentes inteligentes capaces de jugar cualquier juego, es el área de General Game Playing (GGP). En GGP los agentes deben ser autónomos capaces de desarrollar sus propias estrategias sin intervención humana, sin haber jugado el juego con anterioridad y partiendo únicamente de las reglas que le son suministrados momentos antes de jugar (generalmente se utiliza Game Description Languaje) [4,11,12,20]. Aunque los agentes en GGP no pueden compararse en rendimiento y eficiencia a agentes especializados, su utilidad es mayor ya que éstos pueden desempeñarse en diferentes dominios, e incluso las técnicas desarrolladas en GGP pueden ser usadas en otras áreas como gestión de procesos de negocios, comercio electrónico, operaciones militares, entre otros [11,12].

Un avance importante en el área de GGP ha sido la implementación del método Monte Carlo Tree Search (MCTS), idea propuesta por primera vez en [10] y usada en el agente CadiaPlayer, el cual ganó tres veces la competencia internacional de GGP, definiendo el estado del arte del desarrollo de agentes en GGP [11]. MCTS es un método que se guía por simulaciones Monte Carlo [5], usando los resultados de exploraciones previas para estimar con mejor precisión los valores de los movimientos más prometedores [5,9].

Los algoritmos basados en MCTS hacen uso de un árbol donde cada arista representa un movimiento posible y cada nodo representa un estado del juego. A cada nodo se le asocia una serie de estadísticas como el número de victorias y el número de visitas al nodo. En cada iteración realiza cuatro pasos [9]: selección, expansión, simulación y propagación hacia atrás. En el paso de selección se recorre el árbol desde la raíz hasta un nodo que aún tenga nodos hijos por agregar al árbol, el recorrido es guiado por una política de selección que determina qué

nodo debe visitarse en cada nivel. En el paso de expansión un nodo se agrega al árbol. En el paso de simulación, se realiza una ejecución del juego partiendo del nodo hijo, en esta simulación el agente realiza los movimientos de los jugadores siguiendo una política de simulación hasta terminar el juego. En el paso de propagación hacia atrás, el resultado de la simulación es propagado en todos los nodos visitados actualizando sus estadísticas.

Cada vez que el algoritmo MCTS realiza el paso de selección se enfrenta con el siguiente dilema: ¿qué nodo se debe visitar, aquel que parece ser el mejor hasta el momento o nodos menos prometedores que quizás resulten ser mejores en iteraciones posteriores?. Este dilema es conocido como el dilema de exploración-explotación, al cual pertenece el problema denominado Multi-Armed Bandid Problem (MABP) [15,19,21]. El MABP consiste en lo siguiente, dado un conjunto de K máquinas traga monedas, donde cada una de ellas tiene cierta probabilidad de dar una recompensa, la pregunta es; en qué máquina debe jugarse si se desea maximizar la recompensa, ¿la que ha dado la mayor recompensa hasta el momento ó se debe probar otra máquina con la esperanza de que se obtenga una mejor recompensa?. Upper Confidence Bound (UCB), propuesto en [3], es el algoritmo más popular para MABP, y se ha usado en MCTS como política de selección con buenos resultados [10], a esta combinación se le conoce como Upper Confidence Bounds Applied to Trees (UCT).

UCB debido a sus resultados en juegos específicos, es la política más utilizada en GGP [7,8], principalmente a que es independiente del dominio, es decir no requiere de conocer el juego que se aborde sino cómo es el comportamiento de la distribución de recompensas del juego. Desde la concepción de UCB se han desarrollado nuevas políticas para MABP como UCB-Tuned [10], UCB-V, PAC-UCB [2], UCB-Minimal [14], Minimax Optimal Strategy (MOSS) [1], entre otros [5]. Hasta el momento no se cuenta con información que reporte la aplicación de estas políticas en GGP, incluso la política PAC-UCB no se ha probado en MCTS [5], esto debido quizás a los resultados obtenidos por UCB. Sin embargo, estas políticas también son independientes del dominio por lo cual se pueden aplicar a GGP pudiendo presentar mejores resultados que UCB.

Perick et al. [16] hace un comparativo de estas políticas en MCTS con el juego de Tron donde UCB-Tuned es quien tiene el mejor desempeño. Si estas políticas se aplican al dominio GGP, y debido a la variedad de juegos que éstos abordan, es posible que el resultado obtenido por Perick et al. varie.

En este artículo se muestra un estudio comparativo del rendimiento de las políticas UCB y UCB-Tuned en GGP usando los juegos Breakthrough y Knightthrough. La idea de este comparativo es analizar el comportamiento de cada una de estas políticas y su rendimiento en el contexto de GGP.

2. Monte Carlo Tree Search

El método Monte Carlo Tree Search hace uso de un árbol de juego, en el cual cada nodo representa un estado del juego y cada nodo un movimiento, en la Figura 1, se puede observar una parte del árbol del juego para el juego del gato.

Los nodos adicionalmente de representar los estados del juego, llevan registro del número de victorias y de visitas de sus nodos hijos.

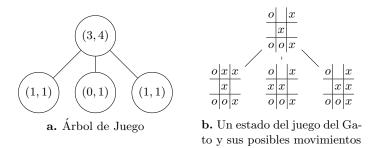


Fig. 1. Árbol del juego para el juego del Gato.

El método Monte Carlo Tree Search requiere de un árbol de juego y consta de cuatro pasos [5]:

Selección En este paso el método recorre el árbol de juego hasta un nodo hoja, en cada nivel el método debe decidir que nodo debe de expandirse, esta decisión es guiada por una política de selección que se basada en el número de victorias y de visitas de los nodos. En un inicio se usaba la tasa entre el número de victorias y las visitas, sin embargo popularmente se usa UCB debido a sus buenos resultados.

Expansión Una vez seleccionado el nodo hoja, este se expande por medio de los movimientos posibles del juego en el estado representando por el nodo hoja.

Simulación Partiendo del estado del juego del nodo hoja seleccionado, el método realiza los movimientos de los dos jugadores de manera aleatoria hasta lograr un resultado: victoria o derrota. Una desventaja de usar movimientos aleatorios radica en que el método no refleja a un jugador inteligente, por ejemplo en la Figura 1b el método podría elegir los movimientos de la izquierda o el de en medio, sin embargo, un jugador inteligente elegiría el movimiento de la derecha. Cazenave [7,8] ha mostrado que usando movimientos de simulaciones anteriores, en las que se lograron victorias, en lugar de movimientos aleatorios se puede lograr un mejor rendimiento del método.

Retro-propagación En este paso el resultado de la simulación es propagado en todos los nodos que se recorrieron en el paso de selección, incrementado el número de visitas y de vitorias

Estos pasos se repiten hasta que se cumple una condición de paro, como cierta cantidad de tiempo, de memoria o cierto número de simulaciones. Finalmente se escoge el movimiento del nodo del primer nivel del árbol con la mejor relación entre victorias y visitas.

2.1. Políticas de selección

En cada iteración del MCTS se decide que nodo del árbol debe visitarse en cada nivel, esta selección es guiada por una política de selección que debe ser capaz de balancear la explotación y la exploración, MABP es un problema que se enfrenta al mismo dilema. Dado un conjunto de máquinas traga-monedas que pueda dar o no una recompensa, y cuya distribución es desconocida e independiente del resto, el problema MABP consiste en elegir la máquina para activar en cada momento de tal manera que se obtenga la mayor recompensa posible

La selección de nodos en MCTS es equivalente a la selección de máquinas en MABP, cada nodo puede verse como una máquina, la cual da una recompensa cuando obtiene una victoria en su simulación, y cada visita equivale a una activación. Por lo anterior se pueden utilizar los mismos métodos de MABP en MCTS.

El proceso de selección de la máquina traga-monedas se conoce como política de selección. En un inicio la política de selección consistía en elegir aquella máquina que en promedio da mas veces una recompensa, sin embargo, se han propuesto nuevas políticas de selección en las que destacan UCB y UCB-T

Upper confidence bounds. Propuesta por Auer [3] para el Multi-Armed Bandit Problem, establece que la selección de la máquina que debe jugarse es aquella que maximice la ecuación (1):

$$\bar{x}_j + c\sqrt{\frac{\ln n}{n_j}},\tag{1}$$

donde $c = \sqrt{2}$, \bar{x} es el promedio de las recompensas de la máquina j, n_j es el número de veces que la máquina j ha sido jugada hasta el momento, y n el número total veces que se han jugado todas las máquinas.

UCB puede ser divido en dos términos:

$$\alpha = \bar{x}_i, \tag{2}$$

$$\beta = c\sqrt{\frac{\ln n}{n_j}}. (3)$$

La ecuación (2) se encarga del proceso de explotación y la ecuación (3) del proceso de exploración. Durante la ejecución se puede ver que conforme una máquina es jugada más veces, el término de exploración deja de influir y el de explotación incrementa su influencia, lo que permite que UCB explore todas las máquinas.

Upper confidence bounds tuned. Auer [3] propone UCB-Tuned como una mejora a UCB, en la cual la máquina a jugar es aquella que tenga el valor máximo:

Iván Francisco-Valencia, José Raymundo Marcial-Romero, Rosa María Valdovinos-Rosas

$$\bar{x}_j + \sqrt{\frac{\ln n}{n_j} \min\left\{\frac{1}{4}, V_j(n_j)\right\}},\tag{4}$$

$$V_{j}(s) = \left(\frac{1}{s} \sum_{\tau=1}^{s} x_{j,\tau}^{2}\right) - \bar{x}_{j,s}^{2} + \sqrt{\frac{2 \ln t}{s}},$$
 (5)

donde s es el número de veces que se ha jugado la máquina, t el número de veces que alguna máquina ha sido jugada. Se debe observar que en la ecuación (5) se hace uso de la varianza muestral y la constante $\frac{1}{4}$ en la ecuación (4) surge del límite superior de la varianza de una variable aleatoria Bernoulli.

3. Escenario de pruebas

Para realizar el comparativo se hizo uso del framework GGP Base¹ y los juegos para dos personas por turnos: Breakthrough y Knightthrough (dos de los juegos usados en [7,8]).

3.1. GGP-base

GGP-Base es un framework, escrito en Java, que permite el desarrollo de agentes para GGP, además de incorporar un Servidor que permite realizar encuentros entre agentes. GGP-Base puede conectarse a servidores proveedores de juegos lo que permite probar los agentes desarrollados en distintos escenarios.

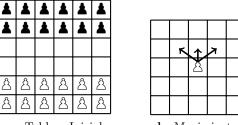
3.2. Breakthrough

El juego hace uso de un tablero cuadriculado de nxm. En un inicio las dos primeras filas de cada jugador contienen peones (ver Figura 2). El objetivo radica en llevar un peón a la primera fila del contrincante para ello el peón puede moverse una casilla hacia adelante o en diagonal siempre hacia adelante (ver Figura 2b). El peón puede capturar solamente en diagonal.

Para la investigación aquí mostrada se uso un tablero de 6×6 disponible en GGP-Base.

¹ http://github.com/ggp-org/ggp-base

Upper Confidence Bound o Upper Cofidence Bound Tuned para General Game Playing...



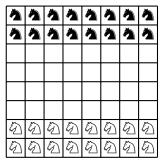
a. Tablero Inicial

b. Movimientos

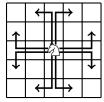
Fig. 2. Breakthrough 6×6 .

3.3. Knightthrough

Similar a Breakthrough con la diferencia de que en lugar de usar peones se usan caballos que se mueven de igual manera que en el ajedrez (ver Figura 3). El objetivo es llevar un caballo a la primera fila del contrincante. Para la investigación aquí mostrada se usó un tablero de 8×8 , de igual manera, disponible en GGP-Base.



a. Tablero Inicial



b. Movimientos

Fig. 3. Knightthrough 8×8 .

4. Ejecución y resultados

Para comparar las políticas se crearon dos agentes, uno basado en UBC y otro basado en UCB-Tuned. Para MCTS en el paso de simulación se estableció que se usarán 100 simulaciones por iteración. Para disminuir el sesgo aleatorio, el análisis se realizó en tres etapas, cada una de las cuales los agentes compitieron en 1000 juegos de Breakthrough y 1000 juegos Knightthrough, registrando el número de vitorias de cada uno de los agentes.

De lo anterior se obtuvieron los resultados mostrados en la Tabla 1. De los resultados se puede observar que el agente que tiene el mejor rendimiento,

en términos de victorias, es el basado en UCB-Tuned, ganando en promedio el 60,86% de las veces en el juego Breakthrough y 78,1% en el juego Knightthrough.

Etapa Breakthrough Knightthrough 1 3922172 397 220 UCB 3 385223 391,333 \bar{x} 220,0 σ 6,0277 3.0 1 608 786 2 780 603 UCB-TUNED 3 615 777 \bar{x} 608,666 7814,5825 6,0277

Tabla 1. Resultados.

5. Conclusiones y trabajo futuro

En este artículo se mostró un análisis comparativo de dos políticas de selección, UCB y UCB-Tuned, en el contexto de GGP, esto con la finalidad de determinar, qué política tiene el mejor rendimiento en términos del número de victorias. Tomando los resultados obtenidos del análisis comparativo se puede concluir que la política de selección que tiene mejor rendimiento en GGP, es UCT-Tuned. El motivo de que UCB-Tuned tenga mejor rendimiento que UCB puede deberse a que UCB-Tuned hace uso de la varianza muestral para determinar qué máquina debe jugarse, en tanto que UCB solo asume una distribución arbitraria.

Se deja como trabajo futuro el uso de otros juegos para determinar si el comportamiento de UCB-Tuned se mantiene estable en éstos. Así mismo se propone realizar un análisis utilizando las políticas UCB-V, PAC-UCB, UCB-Minimal y Minimax Optimal Strategy, con diversas cantidades de simulaciones en MCTS.

Referencias

- Audibert, J.Y., Bubeck, S.: Minimax policies for adversarial and stochastic bandits.
 In: Conference on Learning Theory. pp. 217–226 (2009)
- 2. Audibert, J.Y., Munos, R., Szepesvári, C.: Tuning bandit algorithms in stochastic environments. In: International conference on Algorithmic Learning Theory. vol. 4754, pp. 150–165. Springer (2007)
- 3. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine learning 47(2-3), 235–256 (2002)
- Björnsson, Y., Schiffel, S.: Comparison of GDL reasoners. In: Proceedings of the IJCAI-13 Workshop on General Game Playing (GIGA'13). pp. 55–62 (2013)

- Browne, C.B., Powley, E., Whitehouse, D., Lucas, S.M., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., Colton, S.: A survey of monte carlo tree search methods. IEEE Transactions on Computational Intelligence and AI in games 4(1), 1–43 (2012)
- Campbell, M., Hoane, A., hsiung Hsu, F.: Deep blue. Artificial Intelligence 134(1), 57–83 (2002)
- 7. Cazenave, T.: Playout policy adaptation for games. In: Advances in Computer Games. pp. 20–28. Springer (2015)
- 8. Cazenave, T.: Playout policy adaptation with move features. Theoretical Computer Science 644, 43–52 (2016)
- 9. Chaslot, G.: Monte-carlo tree search. Maastricht: Universiteit Maastricht (2010)
- Finnsson, H., Björnsson, Y.: Simulation-based approach to general game playing. In: AAAI. vol. 8, pp. 259–264 (2008)
- Genesereth, M., Björnsson, Y.: The international general game playing competition. AI Magazine 34(2), 107 (2013)
- 12. Genesereth, M., Love, N., Pell, B.: General game playing: Overview of the AAAI competition. AI magazine 26(2), 62 (2005)
- 13. Hsueh, C.H., Wu, I.C., Tseng, W.J., Yen, S.J., Chen, J.C.: Strength improvement and analysis for an mcts-based chinese dark chess program. In: Advances in Computer Games. pp. 29–40. Springer (2015)
- Maes, F., Wehenkel, L., Ernst, D.: Automatic discovery of ranking formulas for playing with multi-armed bandits. In: European Workshop on Reinforcement Learning. pp. 5–17 (2011)
- 15. Munos, R., et al.: From bandits to monte-carlo tree search: The optimistic principle applied to optimization and planning. Foundations and Trends in Machine Learning 7(1), 1–129 (2014)
- Perick, P., St-Pierre, D.L., Maes, F., Ernst, D.: Comparison of different selection strategies in monte-carlo tree search for the game of tron. In: Computational Intelligence and Games (CIG), 2012 IEEE Conference on. pp. 242–249 (2012)
- 17. Schaeffer, J., Burch, N., Björnsson, Y., Kishimoto, A., Müller, M., Lake, R., Lu, P., Sutphen, S.: Checkers is solved. science 317(5844), 1518–1522 (2007)
- 18. Schaeffer, J., Lake, R., Lu, P., Bryant, M.: Chinook the world man-machine checkers champion. AI Magazine 17(1), 21 (1996)
- Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction, vol. 1. MIT press Cambridge (1998)
- 20. Świechowski, M., Mańdziuk, J.: Specialized vs. multi-game approaches to AI in games. In: Intelligent Systems' 2014, pp. 243–254. Springer (2015)
- Zhou, L.: A survey on contextual multi-armed bandits. CoRR abs/1508.03326 (2015)

C-PSO: Optimización por cúmulo de partículas incrustando la ley de Coulomb

Ricardo Solano Monje¹, Nayeli Joaquinita Meléndez Acosta², Cosijopii García García², Homero Vladimir Ríos Figueroa³

¹ Instituto Tecnológico Superior de Cuidad Serdán, Cd. Serdán, Puebla, México

² Universidad del Istmo campus Ixtepec, Oaxaca, México

³ Universidad Veracruzana, Xalapa, Veracruz, México

rsolano@tecserdan.edu.mx,

{nayelimelendez,cosijopiigarci}@gmail.com, hrios@uv.com.mx

Resumen. Este artículo presenta una comparación del algoritmo PSO (Particle Swarm Optimization) contra una propuesta original a la cual hemos llamado C-PSO, esta versión hace uso de la Ley de Coulomb inspirados en el hecho que las aves tienen moléculas que les permiten detectar el campo magnético de la tierra y guiándolos hacia su destino. Por tanto cuando las aves migratorias realizan su recorrido utilizan su brújula interna. Saber esto nos llevó a pensar que quizás el algoritmo está incompleto y que era necesario agregar otra característica que guiara la dirección de la partícula mejorando el rendimiento del algoritmo PSO, así que recreamos la Optimización por enjambre de partículas incrustando la Ley de Coulomb. En la pruebas se ha incluido también al Algoritmo de Búsqueda Gravitacional, debido a su semejanza con nuestro algoritmo. Los algoritmos analizados y el propuesto fueron probados en 10 funciones de optimización con 1000 iteraciones para 10, 30, 50 y 100 dimensiones. Los resultados obtenidos muestran que el algoritmo propuesto C-PSO presenta mejores resultados en comparación con PSO y en algunos casos supera GSA.

Palabras clave: cúmulo de partículas, optimización por cúmulo de partículas PSO, ley de Coulomb, algoritmo de búsqueda gravitacional GSA.

C -PSO: Particle Swarm Optimization by Embedding Coulomb's Law

Abstract. This article presents a comparison of the classical PSO algorithm (Particle Swarm Optimization) and a novel proposal we have called C-PSO, this new approach makes use of the Coulomb Law inspired by the fact that birds have molecules in their eyes allowing them to detect the magnetic field of the earth then guiding them towards their destination. Therefore when the migratory birds make their journey they use their internal compass. Knowing this led us to think that perhaps the algorithm was incomplete and that it was

necessary to add another feature that would guide the direction of the particle then improving the performance of the PSO algorithm, so we recrafted Particle Swarm Optimization by embedding the Coulomb Law (Coulomb Particle Swarm Optimization C-PSO). The Gravitational Search Algorithm (GSA) has also been included in the tests due to its similarity to our algorithm. The algorithms were tested on 10 optimization functions with 1000 iterations for 10, 30, 50 and 100 dimensions. The results obtained show that the proposed algorithm C-PSO gives better results compared to PSO and in some cases surpasses GSA.

Keywords: particle swarm, particle swarm optimization (PSO), Coulomb's law, gravitational search algorithm (GSA).

1. Introducción

La optimización es el proceso usado para encontrar la mejor solución a un problema determinado. Existen dos tipos de técnicas de optimización las estocásticas y las determinísticas. Los métodos determinísticos conducen a enormes esfuerzos computacionales en contra parte con los algoritmos estocásticos, quieres presentan eficiencia computacional.

Los algoritmos inspirados en la naturaleza son métodos heurísticos, no determinísticos, que simulan el comportamiento de sistemas naturales y tienen la capacidad de resolver problemas complejos a partir de condiciones y reglas iniciales, con algún o ningún conocimiento del espacio de búsqueda [1].

Tres son los elementos más importantes en el diseño de un algoritmo bioinspirado, elegir la representación apropiada del problema, la función de aptitud que medirá la calidad de las soluciones y la definición de operadores para generar nuevas soluciones.

La optimización usando algoritmos bioinspirados tiene dos categorías principales: algoritmos basados en cúmulos de partículas y algoritmo evolutivos. El algoritmo basado en la optimización por cúmulo de partículas (PSO, Particle Swarm Optimization) emula el comportamiento colectivo de los animales y los algoritmos evolutivos emulan la evolución natural [2].

Exploración es la capacidad de un algoritmo de búsqueda, para explorar diferentes regiones del espacio de búsqueda con el fin de localizar una solución óptima. La explotación, por otra parte, es la capacidad de concentrar la búsqueda en un área prometedora para refinar una solución candidata [3].

En los problemas de optimización se buscan técnicas heurísticas y meta heurísticas que proporcionen buenos resultados en tiempo de cómputo razonables.

Cuando las aves migran viajan largas distancias, suelen viajar al mismo lugar año tras año, es decir son muy precisos en su navegación. Al principio se pensaba que se guiaban por el sol, pero entonces no podrían navegar en días nublados y sobre el océano, así que descubrieron que las aves pueden ver el campo magnético y orientarse en consecuencia [4,5], pero no solo en las aves, también los reptiles como las tortugas [6] y animales marinos como focas, leones marinos y elefantes marinos [7].

Saber esto nos llevó a buscar y pensar que quizás el algoritmo PSO estaba incompleto y que era necesaria agregar otra característica que guiara la dirección de la partícula, de esta forma es que creamos C-PSO, el algoritmo de Optimización por Cúmulo de Partículas usando la Ley de Coulomb.

Este artículo presenta una comparación del algoritmo PSO y una nueva versión llamado C-PSO, los cuales permiten resolver problemas numéricos de optimización. El objetivo del artículo es probar el desempeño del algoritmo bio-inspirado llamado C-PSO utilizando un conjunto de funciones para su optimización. Adicionalmente en las pruebas se agregado el algoritmo GSA. El contenido del artículo se encuentra organizado de la siguiente manera, primero en la segunda sección se describe el PSO básico, luego en la tercera sección se describe la nueva versión C-PSO, posteriormente en la cuarta sección muestran las pruebas y resultados realizados. Finalmente, en la quinta sección se presentan las conclusiones.

2. Algoritmo optimización por cúmulo de partículas (PSO)

El algoritmo basado en la optimización por cúmulo de partículas fue inventado por el Dr. Russell C. Eberhart y Dr. James Kennedy en 1995. Esté algoritmo emula el comportamiento de las sociedades de animales que no tienen ningún líder en su cúmulo, como en el caso de las bandadas de aves y los bancos de peces [3].

Los miembros del cúmulo se comunican para informase quien tiene la mejor solución y todas las partículas se mueven hacia ese lugar. Esto sucede siempre que se encuentre una mejor fuente de alimento (o solución).

El algoritmo consiste en un cúmulo de partículas que se colocan en el espacio de búsqueda de algún problema o función, donde cada partícula representa una solución potencial al problema. La solución optimizada para cada partícula es determinada por la función de aptitud. Cada partícula determina su movimiento a través de la combinación de dos capacidades esenciales: la memoria de su mejor posición (local best, lbest) y el conocimiento de la mejor posición encontrada por todas las partículas, es decir lo mejor del cúmulo (global best, gbest). La siguiente iteración tiene lugar después de que todas las partículas se han movido. El cúmulo colectivamente busca la mejor solución, así como las aves buscan su comida

Cada particular en el cúmulo está compuesta por tres vectores: la posición actual x_i , la mejor posición anterior *lbest* y la velocidad v_i . La posición es un conjunto de coordenadas que describen un punto en el espacio. En cada iteración del algoritmo, la posición actual se evalúa como una solución de problema. Si esa posición es mejor que la que se ha encontrado hasta ese momento, entonces las coordenadas se almacenan en el segundo vector *lbest*. El valor de la mejor solución encontrada por el cúmulo hasta ese momento se almacena en *gbest*, este también se actualiza si la solución es mejor. El objetivo del algoritmo es seguir encontrando mejores soluciones y actualizar *lbest* y *gbest*.

La velocidad de la partícula se modifica en cada iteración de la ecuación (1) utilizando la velocidad anterior multiplicada por una constante w, un componente cognitivo y un componente social. El componente cognitivo representa la distancia entre la posición actual y la mejor posición conocida por esa partícula (lbest) y el

componente social representa la distancia entre la posición actual y la mejor posición del cúmulo (gbest) [8].

$$v_{i+1} = w \cdot v_i + c_l \cdot r_l \cdot (lBest_i - x_i) + c_a \cdot r_a \cdot (gBest_d - x_i). \tag{1}$$

Una partícula se mueve desde una posición del espacio de búsqueda hasta otra ecuación (2), añadiendo al vector posición x_i el vector velocidad v_i para obtener un nuevo vector posición de la siguiente manera:

$$x_i = x_{i-1} + v_i, (2)$$

donde x_i es la posición de la partícula i, v_i es la velocidad de la partícula i.

2.1. Algoritmo de búsqueda gravitacional (GSA)

Rashedi en 2009 propuso el Algoritmo de Búsqueda Gravitacional (Gravitational Search Algorithm, GSA), está basado en la ley de gravitación universal que considera que dos objetos se atraen entre sí por la fuerza de la gravedad y esta fuerza provoca un movimiento global en todos los objetos, las masa más pesadas que corresponden a las mejores soluciones se mueven más lentamente que las masas ligeras para garantizar la exploración del algoritmo [9].

3. Optimización por cúmulo de partículas usando la ley de Coulomb (C-PSO)

C-PSO es una nueva versión de PSO que hace uso de la Ley de Coulomb, el uso de esta ley es meramente una decisión de cierta forma arbitraria, pero no del todo, ya que la idea surgió del objetivo de guiar la dirección que toma cada partícula, pensando en el hecho de saber que "los pájaros pueden ver el campo magnético de la tierra". Los ojos de las aves tienen moléculas que les permiten detectar el campo magnético de la tierra. Por tanto, cuando las aves migratorias realizan su recorrido utilizan su brújula interna [5].

Esto nos llevó a pensar que quizás el algoritmo está incompleto y que era necesaria una característica que guiara la dirección de la partícula mejorando el algoritmo de PSO, de esta forma es que creamos el algoritmo de Optimización por Cúmulo de Partículas usando la Ley de Coulomb C-PSO (Coulomb Particle Swarm Optimizaton).

3.1. Ley de Coulomb

En 1784, el físico francés Charles Augustín Coulomb, descubrió la ley cuantitativa de las fuerzas entre dos cargas puntuales, ecuación (3). Cargas puntuales son aquellas cuyas dimensiones geométricas son despreciables comparadas con las distancias de separación entre ellas [10].

Suponga que dos cargas puntuales, m_1 y m_2 , están separadas a una distancia d en el vacío. Si m_1 y m_2 tienen el mismo signo, las dos cargas se repelen mutuamente; si poseen signos opuestos, se atraen una a la otra. La fuerza que experimenta una carga

debido a la otra se conoce como fuerza de Coulomb o eléctrica y está dada por la Ley de Coulomb:

$$F = k \frac{m_1 m_2}{d^2},\tag{3}$$

donde:

- F es la fuerza magnética,
- m₁ y m₂ son las masas magnéticas de cada partícula, en este caso se utilizan los costos de las partículas,
- k es la constante de permisividad eléctrica, la cual fue remplazada por la permeabilidad magnética absoluta μ la cual se define como:

 $\mu=\mu_r\mu_0$ Donde u_r y u_0 son constantes $\mu=5000\cdot 4\pi\times 10^{-7}$ quedando la fórmula final como $F=\mu\frac{m_1m_2}{d^2}$.

Hemos usado la Ley de Coulomb porque nos hemos imaginado que las partículas son dos cargas puntuales. Ahora para calcular la distancia entre las dos cargas ecuación (4) se hace uso de la distancia entre dos puntos:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. (4)$$

Utilizando la fórmula anterior, pero con dos valores:

$$d = \sqrt{(lBestPos_1 - gBestPos_1)^2 + (lBestPos_2 - gBestPos_2)^2}.$$
 (5)

La ecuación (6) para n par de valores:

$$D = \sqrt{\sum_{i=1}^{n} (lBestPos_i - gBestPos_i)^2}.$$
 (6)

Agregando esta característica a la fórmula del PSO original queda como se observa en la ecuación (7) de la siguiente manera:

$$v_{i+1} = w \cdot v_i + c_p \cdot r_p \cdot (pBest_i - x_i) + c_q \cdot r_q \cdot (gBest_d - x_i), \tag{7}$$

donde w ecuación (8) se define de la siguiente manera:

$$w = \begin{cases} \mu \frac{gBestCost*lBestCost}{d^2}, & d \le 0.5\\ 0, & d > 0.5 \end{cases}$$
 (2)

donde d es la distancia entre la posición del lBest de cada partícula y el gBest utilizando la ecuación (6), donde gBestCost es el mejor costo global y lBestCost es el mejor costo local.

La posición de la partícula es influenciada por la velocidad. Esta posición cambia añadiendo una velocidad, a la posición actual [1]. PSO es más sencillo que otros algoritmos de optimización, puesto que sólo existe un operador de mutación, la velocidad de la partícula.

4. Pruebas y resultados

Para probar el rendimiento de la nueva versión de algoritmo C-PSO implementado en este trabajo se han seleccionado 10 funciones de optimización ecuaciones (9-18), las cuales fueron seleccionadas de [11]. A continuación, se muestran las funciones utilizadas:

Función Ackley

$$f(x) = 20 \exp\left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^{d} x_i^2}\right) - \exp\left(-0.2 \frac{1}{d} \sum_{i=1}^{d} \cos(2\pi x_i)\right) + 20 + \exp(1), \tag{3}$$

$$-32 \le x_i \le 32$$
,
 $\min(f(x)) = f(0 \dots 0) = 0$.

• Función Sphere

$$f(x) = \sum_{i=1}^{d} x_i^2,$$
 (4)

$$-100 \le x_i \le 100,$$

 $\min(f(x)) = f(0 \dots 0) = 0.$

Función Step

$$f(x) = \sum_{i=1}^{d} (\lfloor x_i + 0.5 \rfloor)^2,$$

$$-100 \le x_i \le 100,$$

$$\min(f(x)) = f(0 \dots 0) = 0.$$
(5)

• Función Rastringin

$$f(x) = \sum_{i=1}^{d} (x_i^2 - 10\cos(2\pi x_i) + 10), \tag{6}$$

$$-5.12 \le x_i \le 5.12$$
,
 $\min(f(x)) = f(0 \dots 0) = 0$.

• Función Schwefel 2.22

$$f(x) = \sum_{i=1}^{d} |x_i| + \prod_{i=1}^{d} |x_i|,$$

$$-10 \le x_i \le 10,$$

$$\min(f(x)) = f(0 \dots 0) = 0.$$
(7)

• Función Rosembrock

$$f(x) = \sum_{i=1}^{d-1} \left| 100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right|,$$

$$-30 \le x_i \le 30,$$

$$\min(f(x)) = f(1 \dots 1) = 0.$$
(8)

• Función Griewank

$$f(x) = \frac{1}{4000} \sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$
 (9)

C-PSO: Optimización por cúmulo de partículas incrustando la ley de Coulomb

$$-600 \le x_i \le 600,$$

 $\min(f(x)) = f(0 \dots 0) = 0.$

Función Quartic

$$f(x) = \sum_{i=1}^{d} i x_i^4, \tag{10}$$

$$-1.28 \le x_i \le 1.28,$$

 $\min(f(x)) = f(0 \dots 0) = 0.$

• Función Salomon

$$f(x) = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^{d} x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^{d} x_i^2},\tag{11}$$

$$-100 \le x_i \le 100,$$

 $\min(f(x)) = f(0 \dots 0) = 0.$

• Función general penalizada

$$f(x) = 0.1 \times \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{d-1} (x_i - 1)^2 \left[1 + \sin^2(3\pi x_{i+1}) \right] + (x_d - 1)^2 \left[1 + \sin^2(2\pi x_n) \right] \right\} + \sum_{i=1}^{d} u(x_i, a, k, m),$$
(12)

donde

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a, \\ k(-x_i - a)^m, & x_i < -a \end{cases}$$
$$a = 5, k = 100, m = 4,$$
$$-50 \le x_i \le 50,$$
$$\min(f(x)) = f(1 \dots 1) = 0.$$

En las pruebas se han ejecutado tres algoritmos PSO, GSA y la nueva versión propuesta C-PSO. Estas pruebas han sido diseñadas bajo las competencias del CEC [12] con el objetivo de comparar el rendimiento de los algoritmos bajo un conjunto de prueba más grande y complejo. Las configuraciones de los parámetros para realizar las pruebas se muestran en la tabla 1.

Los algoritmos fueron probados con 10 funciones de optimización corriendo cada algoritmo 50 veces para 10, 30, 50 y 100 dimensiones (véase tablas 2-5). En las tablas se presenta el mejor, el peor, el promedio, la mediana y la desviación estándar. Los resultados muestran que a C-PSO le fue mucho mejor que a PSO, también nos dimos cuenta que C-PSO y GSA son casi iguales, siendo mejor C-PSO en algunas funciones.

Una ventaja del algoritmo C-PSO es que al añadir la Ley de Coulomb no se estanca mucho en los mínimos locales, además también es más rápido en dimensiones grandes comparado con su versión normal PSO.

Tabla 1. Parámetros de configuración.

Método	Pa	rámetro	Valor		
	Condición de paro	No. Máximo de Iteraciones	1000		
PSO C-PSO	Condicion de paro	Error	r < 10e - 8		
	Coeficiente d	e decisión personal	1.8		
	Coeficiente	1.8			
	Coeficiente de	1			
	Población: nú	100			
	Espacio	Indicado en la formula			
	Condición de paro	No. Máximo de Iteraciones	1000		
	Condicion de paro	Error	r < 10e - 8		
GSA	Constante gr	Constante gravitacional inicial			
GSA	Po	100			
	Espacio	Indicado en la formula			

Una limitación de nuestro algoritmo es que en espacios de búsqueda muy grandes el error disminuye cada vez más lento conforme se va acercando al objetivo, esto se compensa con el hecho que hay muy poco estancamiento en mínimos locales.

Tabla 2. 10 dimensiones.

Función	Método	Peor	Promedio	Mediana	Mejor	Desviación Estándar
	PSO	4.6434	4.1835	4.2823	2.8901	3.3688e-01
F1	GSA	9.1844e-10	7.1027e-10	7.0143e-10	5.2529e-10	9.0299e-11
	C-PSO	1.3973	2.7947e-02	8.1978e-09	4.7481e-09	1.9761e-01
	PSO	7.2494e+01	4.9548e+01	4.8874e+01	2.5178e+01	1.1167e+01
F2	GSA	4.1972e-09	2.5243e-09	2.5670e-09	8.6872e-10	6.9139e-10
	C-PSO	9.9438e-09	6.0263e-09	5.9261e-09	1.3679e-09	2.0151e-09
	PSO	96	6.4440e+01	6.5500e+01	20	1.8031e+01
F3	GSA	0	0	0	0	0
	C-PSO	0	0	0	0	0
	PSO	2.5666e+01	1.8389e+01	1.8552e+01	1.2017e+01	2.7875
F4	GSA	2.9849	5.5718e-01	0	0	8.5660e-01
	C-PSO	6.6891e-01	1.7328e-01	1.2931e-01	0	1.9470e-01
	PSO	2.5393	1.9580	1.9622	1.4372	2.7298e-01
F5	GSA	9.9871e-09	8.7683e-09	8.9258e-09	6.3721e-09	9.8703e-10
	C-PSO	9.8754e-09	8.4571e-09	8.5493e-09	5.7628e-09	1.0619e-09
	PSO	2.7844e+05	6.9823e+04	5.7794e+04	8.2261e+03	4.7123e+04
F6	GSA	1.9798e+02	1.8520e+01	5.4451	5.1949	4.1633e+01
	C-PSO	6.0333e+02	3.3051e+01	9	7.4515	9.3606e+01
	PSO	8.1878e-01	6.3663e-01	6.5501e-01	4.7698e-01	8.6228e-02
F7	GSA	9.8573e-03	9.3676e-04	8.2554e-09	5.3021e-09	2.5825e-03
	C-PSO	1.3527e-01	5.1029e-02	5.1650e-02	0	3.3510e-02
F8	PSO	2.7773e-04	1.2729e-04	1.1703e-04	1.6621e-05	5.7889e-05

Función	Método	Peor	Promedio	Mediana	Mejor	Desviación Estándar
	GSA	9.9541e-09	7.0221e-09	7.5196e-09	6.4431e-10	2.4099e-09
	C-PSO	9.9528e-09	5.0270e-09	5.2708e-09	6.9464e-10	2.8921e-09
	PSO	1.3137	1.0519	1.0957	7.6368e-01	1.3454e-01
F9	GSA	1.9994e-01	1.0809e-01	1.0050e-01	9.9873e-02	2.3646e-02
	C-PSO	1.9987e-01	9.9901e-02	9.9873e-02	0	6.6964e-02
	PSO	3.4320	2.2167	2.0974	1.0840	6.0645e-01
F10	GSA	9.8203e-09	7.5572e-09	8.3043e-09	2.6701e-09	1.6825e-09
	C-PSO	1.0988e-02	4.4254e-04	6.8807e-09	1.8090e-09	2.1745e-03

Tabla 3. 30 dimensiones.

Función	Método	Peor	Promedio	Mediana	Mejor	Desviación Estándar
	PSO	9.0590	8.5251	8.5657	7.7965	0.3014
F1	GSA	9.9785e-09	9.2255e-09	9.1271e-09	8.2675e-09	5.1678e-10
	C-PSO	2.4895	0.4284	9.3258e-09	6.3903e-09	0.8165
	PSO	1.6459e+03	1.2974e+03	1.3051e+03	915.8696	147.8678
F2	GSA	9.9943e-09	8.8652e-09	9.0403e-09	5.8492e-09	9.0711e-10
	C-PSO	5.6686	0.2138	7.6648e-09	9.8626e-11	1.0602
	PSO	1592	1.3031e+03	1.3315e+03	758	170.8434
F3	GSA	0	0	0	0	0
	C-PSO	1	0.0200	0	0	0.1414
	PSO	221.8564	179.4923	180.1052	152.2287	13.5775
F4	GSA	11.9395	7.1438	6.9647	2.9849	2.1148
	C-PSO	47.6264	29.0552	29.0450	0	10.4963
	PSO	17.5114	15.2100	15.4030	12.0971	1.1574
F5	GSA	1.2879e-08	1.0228e-08	9.8634e-09	8.3128e-09	1.0127e-09
	C-PSO	3.3713	1.7896	1.9736	0	0.9532
	PSO	2.0607e+07	1.2343e+07	1.2673e+07	5.1112e+06	3.3507e+06
F6	GSA	354.2852	52.6966	26.0432	25.7199	76.3510
	C-PSO	492.2854	130.6455	100.2836	29	111.4637
	PSO	1.3891	1.3181	1.3258	1.1634	0.0494
F7	GSA	0.0221	0.0015	9.0379e-09	5.2854e-09	0.0047
	C-PSO	0.0441	0.0085	0.0037	0	0.0107
	PSO	0.0730	0.0522	0.0520	0.0207	0.0121
F8	GSA	9.9521e-09	8.0200e-09	8.2608e-09	4.8328e-09	1.3746e-09
	C-PSO	9.8256e-09	6.9951e-09	7.1661e-09	1.6977e-09	1.8737e-09
	PSO	4.4008	4.0113	4.0323	3.5050	0.1960
F9	GSA	0.7999	0.4931	0.4999	0.3410	0.1026
	C-PSO	0.5999	0.3379	0.3999	0	0.1307
	PSO	9.8178e+03	2.8002e+03	2.2841e+03	59.9781	2.4078e+03
F10	GSA	0.0108	4.2203e-04	8.7322e-09	5.3589e-09	0.0019
	C-PSO	0.0974	0.0054	8.0324e-09	5.0897e-13	0.0154

Tabla 4. 50 dimensiones.

Función	Método	Peor	Promedio	Mediana	Mejor	Desviación Estándar
F1	PSO	10.6956	10.1705	10.2193	9.2502	0.2617

	GSA	9.9817e-09	9.5161e-09	9.6041e-09	8.7679e-09	3.1137e-10
	C-PSO	3.2808	0.5174	9.8090e-09	5.8261e-09	1.0096
	PSO	4.2407e+03	3.5573e+03	3.6241e+03	2.5155e+03	419.4516
F2	GSA	9.9635e-09	8.9518e-09	9.0464e-09	7.0667e-09	7.3516e-10
	C-PSO	2.3243	0.0465	5.3584e-09	1.0093e-11	0.3287
	PSO	4310	3.7190e+03	3764	2950	316.6848
F3	GSA	0	0	0	0	0
	C-PSO	1	0.1000	0	0	0.3030
	PSO	449.5038	372.1244	373.2912	330.7323	19.9571
F4	GSA	29.8488	15.6408	15.4219	6.9647	4.2501
	C-PSO	89.4245	51.6254	50.5354	20.9171	14.3382
	PSO	37.7666	34.4328	34.4231	29.8963	1.7293
F5	GSA	2.7137e-08	2.0071e-08	2.0071e-08	1.5849e-08	2.0431e-09
	C-PSO	11.3297	3.0373	2.3185	1.0268	2.0830
	PSO	1.0428e+08	7.3024e+07	7.2840e+07	4.6470e+07	1.2407e+07
F6	GSA	381.8985	75.7824	46.3782	44.8286	72.9732
	C-PSO	828.0664	219.5308	183.5675	49	164.3921
	PSO	2.0316	1.9181	1.9304	1.7586	0.0753
F7	GSA	0.0148	0.0014	9.2736e-09	7.1085e-09	0.0036
	C-PSO	0.1331	0.0135	8.7100e-09	0	0.0242
	PSO	0.7144	0.4612	0.4527	0.2953	0.0879
F8	GSA	9.9410e-09	8.2150e-09	8.5915e-09	2.8050e-09	1.4767e-09
	C-PSO	9.9674e-09	7.8372e-09	8.5355e-09	1.6058e-09	2.0493e-09
	PSO	7.2225	6.6977	6.6966	6.2010	0.2421
F9	GSA	2.3044	1.7802	1.7103	1.0027	0.2954
	C-PSO	0.7999	0.5400	0.5999	0	0.1979
	PSO	4.4740e+05	1.4408e+05	1.4149e+05	2.5241e+04	6.5484e+04
F10	GSA	0.4084	0.0103	9.2555e-09	7.1621e-09	0.0576
	C-PSO	0.0440	0.0042	5.7271e-09	2.0569e-11	0.0077

Tabla 5. 100 dimensiones.

Función	Método	Peor	Promedio	Mediana	Mejor	Desviación Estándar
F1	PSO	1.2824e+01	1.2225e+01	1.2197e+01	1.1767e+01	2.5636e-01
	GSA	2.7213e-09	2.3306e-09	2.3404e-09	1.8929e-09	1.7055e-10
	C-PSO	2.5943	1.2989	1.5372	7.1054e-15	8.5017e-01
F2	PSO	1.3481e+04	1.1526e+04	1.1532e+04	1.0151e+04	7.3413e+02
	GSA	2.4663e-07	1.7871e-07	1.7669e-07	1.3286e-07	2.6073e-08
	C-PSO	1.1100e-02	2.2200e-04	3.7623e-09	6.3007e-12	1.5698e-03
F3	PSO	15212	1.3467e+04	1.3621e+04	11259	8.8689e+02
	GSA	94	3.0100e+01	2.6500e+01	7	1.6134e+01
	C-PSO	6	1.2400	1	0	1.3486
F4	PSO	9.5977e+02	8.4649e+02	8.4605e+02	7.7342e+02	3.3430e+01
	GSA	3.3829e+01	2.5949e+01	2.5869e+01	1.3929e+01	4.5254
	C-PSO	2.2528e+02	1.0938e+02	1.0688e+02	5.2293e+01	3.0030e+01
F5	PSO	1.1095e+02	9.6681e+01	9.6231e+01	8.3235e+01	5.3801e+00
	GSA	5.1731e-01	4.3299e-02	2.2635e-03	4.7920e-08	9.1704e-02
	C-PSO	2.7166e+01	5.7383	3.9847	2.0030	5.1023
F6	PSO	8.2480e+08	5.3770e+08	5.3109e+08	3.3714e+08	9.7281e+07
	GSA	1.8937e+05	4.6546e+04	3.8344e+04	9.4960e+03	3.4632e+04

Función	Método	Peor	Promedio	Mediana	Mejor	Desviación Estándar
	C-PSO	1.5969e+03	5.2881e+02	5.6819e+02	99	3.0394e+02
F7	PSO	4.9550	4.3444	4.3553	3.6990	2.4480e-01
	GSA	3.9781e-01	5.3710e-02	3.8489e-02	1.3178e-03	6.0581e-02
	C-PSO	2.1215e-01	3.1499e-02	4.9296e-03	0	5.1660e-02
	PSO	9.4849	6.8017	6.8324	4.9220	1.0094
F8	GSA	9.9931e-09	9.3163e-09	9.3224e-09	8.0422e-09	4.6729e-10
	C-PSO	9.9761e-09	5.1623e-09	5.5475e-09	1.5223e-11	3.4967e-09
	PSO	1.3726e+01	1.2354e+01	1.2341e+01	1.1360e+01	5.0474e-01
F9	GSA	5.7491	4.6803	4.7005	4.0999	3.9002e-01
	C-PSO	1.9999	1.5931	1.5999	9.9987e-01	1.8692e-01
	PSO	1.0215e+07	4.2791e+06	4.1347e+06	2.1372e+06	1.4783e+06
F10	GSA	4.1637e+01	2.2993e+01	2.2407e+01	1.0649e+01	6.6393
	C-PSO	5.3282	1.1469e-01	6.5795e-04	3.7046e-07	7.5250e-01

5. Conclusiones

PSO consiste en un enjambre de partículas, donde la partícula representa una solución potencial (mejor condición). La partícula se moverá a través de un espacio de búsqueda multidimensional para encontrar la mejor posición en ese espacio (la mejor posición posible para los valores máximo o mínimo).

En este artículo se presenta una comparación del algoritmo PSO y una nueva versión llamado C-PSO, en esta nueva versión hacemos uso de la ley de coulomb de ahí su nombre C- PSO. C-PSO ha sido inspirado en el hecho de que las aves pueden ver el campo magnético de la tierra y este guía su destino, así que decidimos agregar otra característica que guiará la dirección de la partícula mejorando el rendimiento del algoritmo original PSO y dando origen a C-PSO.

Los resultados de las pruebas muestran un rendimiento favorable de C-PSO comprado con PSO y GSA. En promedio el algoritmo C-PSO encuentra la solución en la iteración 100, mientras que los otros dos requieren de muchas más iteraciones. Nuestro algoritmo ha mostrado buenos resultados.

Referencias

- Palupi-Rini, D., Mariyam-Shamsuddin, S., Sophiyati-Yuhaniz, S.: Particle Swarm Optimization: Technique, System and Challenges. International Journal of Computer Applications 14(1) (2011)
- Chavan, S. D., Adgokar, N. P.: An Overview on Particle Swarm Optimization: Basic Concepts and Modified Variants. International Journal of Science and Research (IJSR) 4, pp. 255–260 (2015)
- Bansal, J. C., Singh Mukesh Saraswat, P. K., Verma A., Singh Jadon, S., Abraham, A.: Inertia Weight Strategies in Particle Swarm Optimization. In: Third World Congress on Nature and Biologically Inspired Computing, pp. 640–647 (2011)
- National Geographic News Homepage, http://news.nationalgeographic.com/news/2007/09/ 070927-magnetic-birds.html (2007)

- 5. Science X Homepage, https://phys.org/news/2009-11-birds-earth-magnetic-field.html (2009)
- 6. Science X Homepage, https://phys.org/news/2015-01-sea-turtles-magnetic-home.html (2015)
- Science X Homepage, https://phys.org/news/2014-12-antarctic-earth-magnetic-field.html (2014)
- 8. Kumar, A, Kumar Singh, B., Patro, B. D. K.: Particle Swarm Optimization: A Study of Variants and Their Applications. International Journal of Computer Applications (0975 8887) 135(5) (2016)
- Suarez, A. J., Rodríguez, J. E.: Algoritmos Meta heurísticos basados en la ley de gravitación universal para agrupamiento de datos. Sistema, Cibernética e Informática 13(1) (2016)
- Barco Ríos, H., Rojas Calderón, E., Restrepo Parra, E.: Física: principios de electricidad y magnetismo. Primera Edición. Universidad Nacional de Colombia, Colombia (2012)
- 11. Viveros-Jiménez, F., Mezura-Montes, E., Gelbukh, A.: Empirical analysis of a micro-evolutionary algorithm for numerical optimization. International Journal of Physical Sciences 7(8), pp. 1235–1258 (2012)
- 12. Awad, N. H., Ali, M. Z., Suganthan, P. N., Liang, J. J., Qu, B. Y.: Problem Definitions and Evaluation Criterial for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. Technical Report (2016)

Impreso en los Talleres Gráficos de la Dirección de Publicaciones del Instituto Politécnico Nacional Tresguerras 27, Centro Histórico, México, D.F. agosto de 2018 Printing 500 / Edición 500 ejemplares