Advances in Artificial Intelligence

Research in Computing Science

Series Editorial Board

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 82,** noviembre de 2014. 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 82**, November 2014. 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 82

Advances in Artificial Intelligence

Félix Castro Espinoza (ed.)

SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2014

ISSN: 1870-4069

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.

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Editorial

This volume of the journal "Research in Computing Science" contains selected papers on the very popular topic of Artificial Intelligence, when the computers should demonstrate the intelligent behavior in specific situation by modelling possible human actions.

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 special issue were rejected.

This volume contains papers on various topics of artificial intelligence, like bio-inspired algorithms, machine learning and classification (support vector machines, neural networks), automatic reasoning, and several applications (to traffic modelling, auctions, menu selection).

I would like to thank Mexican Society for Artificial Intelligence (Sociedad Mexicana de Inteligencia Artificial), MICAI 2014 conference, Instituto Tecnológico de Tuxtla Gutierrez (Chiapas, Mexico), and Universidad Autónoma de Chiapas for their support during preparation of this volume.

The papers were collected and the reviewing process was organized using the system EasyChair.

Félix Castro Espinoza
November 2014

Table of Contents

An Experimental Study of Parameter Selection in Particle Swarm Optimization Using an Automated Methodology

María Cosío-León[1], Anabel Martínez-Vargas[2], and Everardo Gutierrez[3]

[1] Universidad Autónoma de Baja California, FIAD, Ensenada, BC, Mexico
[2] Centro de Investigación y Desarrollo de Tecnología Digital del Instituto Politécnico Nacional (CITEDI-IPN), Tijuana, BC, Mexico
[3] Universidad Autónoma de Baja California, FC, Ensenada, BC, Mexico
cosio.maria@uabc.edu.mx, amartinez@citedi.mx,
everardo.gutierrez@uabc.edu.mx

Abstract. In this work, an experimental study to evaluate the parameter vector utility brought by an automated tuning tool, so called *Hybrid Automatized Tuning procedure* (HATp) is given. The experimental work uses the inertia weight and number of iterations from the algorithm PSO; it compares those parameters from tuning by analogy and empirical studies. The task of PSO is to select users to exploit concurrently a channel as long as they achieve the Signal-to-Interference-Ratio (SINR) constraints to maximize throughput; however, as the number of users increases the interference also arises; making more challenging for PSO to converge or to find a solution. Results show that, HATp is not only able to provide a parameter vector that improve the search ability of PSO to find a solution but also to enhance its performance on resolving the spectrum sharing application problem than those parameters values suggested by empirical and analogical methodologies in the literature on some problem instances.

Keywords: Parameter tuning, metaheuristic, particle swarm optimization.

1 Introduction

Meta-heuristic algorithms are black box procedures that, provided a set of candidate solutions, solve a problem or a set of problems instances. However, they require to select a set of parameters to tuning them, which greatly affect the meta-heuristic's efficiency to solve a given decision problem. Those parameters are classified as qualitative and quantitative; the former are related to procedures (e.g Binary or Continuous PSO), while the latter are associated with specific values (e.g. number of iterations, and population size). This work is focused on quantitative parameters to configure the PSO algorithm; which is a non-trivial problem as authors in [1] explain. This problem in the literature is called *algorithm configuration* by authors in [2]; and *parameter tuning* in [1], [3].

In [1] authors define the parameter tuning procedure as the task in which parameter values are set before executing a given meta-heuristic; and those values remain fixed while the meta-heuristic is running. Due to the aforementioned, parameter's tuning is an important task in the context of developing, evaluating and applying meta-heuristic algorithms.

1.1 Particle Swarm Optimization Algorithm

To evaluate parameter vector utility bring by the tuning procedure, this paper uses the Particle Swarm Optimization (PSO) algorithm [4], which is categorized by its authors as an evolutionary computation technique since it utilizes a population of candidate solutions to evolve an optimal or near-optimal solution for a problem.

The individuals in the PSO technique are called particles and they represent a possible solution of the optimization problem. When elements of a problem are represented as binary variables, the binary version of PSO (BPSO) is used [7]. Since its inception, many adjustments have been made to improve its performance. One of these new improvements to BPSO algorithms is Socio-Cognitive Particle Swarm Optimization (SCPSO) [8]. SCPSO introduces the distance between gbest and pbest values as a new velocity update equation which maintain diversity in the swarm, a socio-cognitive scaling parameter c_3 and a new position update equation. The latter used on spectrum sharing application to maximize throughput in the network.

This feature article is about analyzing two procedures for optimization parameters on SCPSO algorithm: a) model-base CALIBRA algorithm [9], and b) polynomial interpolation technique called Newton's Divided Difference Polynomial Method of Interpolation [10]. Along with aforementioned procedures, we use as a control group, parameter vector values taken from the state of art, tuning by analogy (TA) and empirical methodology to test the parameter vector utility.

2 Automatic Parameter Tuning

The automated tuning procedures address the parameter tuning problem; they are designed to search for the best parameter vector. Therefore, given a meta-heuristic with n parameters, tuning procedures search for the best parameter vector $P^* = \{p_0, p_1, \ldots, p_n\}$. The parameter vector P^* usually is selected by researchers using manual tuning procedures [11] or tuned by analogy's procedures [12]. The No Free Lunch theorem of optimization states that; one P^* allowing to solve all optimization problems is verifiable non-existent; therefore, tuning by analogy procedure, which uses a single parameter vector for different problems or different problem instances, is not the best strategy. On the other hand, manual tuning procedures are very time consuming, and failure prone; therefore, it is necessary to conduct other procedures to avoid those drawbacks.

In [13] the author gives a brief review about the automated parameter tuning procedures; using a two fold model classification: a) model-free, and b) model-based approaches. The former models are procedures guided by randomness, or simple experimental design (e.g. Latin Hypercube Sampling), tuners with very limited extrapolation potential. On the other hand, the latter models have the capabilities of 1) interpolating for the choice of new parameter settings; and even 2) extrapolating parameter vectors for new problems or problem instances. In this context, interesting contributions to find P^* through automated procedures are presented in [1], [3].

In the next section, we will describe CALIBRA, and Newton's Divided Difference Polynomial Method of Interpolation which is the the interpolation technique selected to find new P^* for problem instances.

2.1 The Hybrid Automatized Tuning Procedure (HATp)

Traditional tuning methods comprises three layers: a) design layer; b) algorithm layer; and c) application layer [1]. In this experimental study, we propose to use in the design layer an Hybrid Automatized Tuning procedure (HATp). Firstly, it exploits a procedure that couples fractional factorial experimental design and a local search procedure, called CALIBRA [9]. Then, an interpolation method such as Newton's divided difference polynomial works with CALIBRA to bring a particular P^* for problem instances; while reducing computer time.

HATp's first stage uses CALIBRA (HATpI); it sets up a series of experiments to find the best value for quantitative parameters in the tuning target algorithm. The notion of best depends on how the performance of the target algorithm is measured. To achieve this, CALIBRA combines two methods: experimental designs and local search. The experimental designs focus on the on promising regions [9]. Promising regions are selected using a full factorial design 2^k, and Taguchi's $L_9(3^4)$; once a region is selected, CALIBRA makes a local search. The above procedure is executed until certain stopping condition is met. CALIBRA uses P to configure the interest algorithm; same process is executed several times with P obtained from promising regions by CALIBRA up to find P^*. It is important to denote that the CALIBRA software can provided up to five parameter calibration; so for metaheuristics with more than five parameters, it is necessary to develop a new CALIBRA software version.

Considering the No Free Lunch theorem of optimization; and a continuous local function $f(x)$; once CALIBRA brought a set of parameter vectors P^*, HATp uses a polynomial interpolation method to find new problem instances P^* (HATpII). The interpolation process takes advantage of CALIBRA model-base characteristic; building a polynomial of order n that passes through the $1 + n$ points calculated by CALIBRA. To find the new points, the interpolation process uses Newton's divided differences recursive equations (1), (2), (3):

$$f[x_i] = y_i = f(x_i) \tag{1}$$

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i} \tag{2}$$

$$f[x_{i+1}, x_{i+2}, \ldots, x_{i+n}] = \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+n}] - f[x_i, x_{i+1}, x_{i+2}, \ldots, x_{i+n-1}]}{x_{i+n} - x_i} \tag{3}$$

Formalizing the automated parameter tuning procedure HATp: suppose that the performance of algorithm Ac is to be studied for a given set of problem instances I; P^* is found using a model based algorithm Ca; using a set of problems instances I' different to I. Once Ca brings the P^*, the algorithm Ac is configured with it, and a problem instance from I is resolved. Performance measures are selected according to problem instances open questions.

The second strategy in HATp is an interpolation procedure, Dd to find P^* for new problem instances as follows: given a continuous function f and a sequence of known points $x_0; x_1; \ldots; x_n$. the divided difference of f over $x_0; x_1; \ldots; x_n$ points is the value of $a_n = f[x_0; x_1; \ldots; x_n]$; which is recursively computed by equations (1), (2), (3), in intention to find P^*, and reduce tuning computer time.

3 Target Problem and Experimental PSO Setup

In cognitive wireless networks with spectrum underlay when a secondary transmitter requests for a primary channel, they must be able to check if mutual interference among secondary users (unlicensed users) and primary users (licensed users) doesn't rise to the level of harmful interference. In this case the primary users have priority over a specific channel, and secondary users are allowed to transmit in the same channel as long as they do not cause harmful interference to the primary user.

Consider Figure 1, there is a number of secondary links Sl and primary links Pl are deployed in a coverage area A. A link either secondary or primary is represented by the union of a transmitter and a receiver and it is identified by a number beside the link. The number of primary links Pl is the primary network, which is assigned with a portion of regulated spectrum. Whereas, the secondary network is composed by the number of secondary links Sl, which have to find a primary channel to exploit it. The cognitive network has a central entity; it knows the number of primary channels that can be assigned to secondary links. The primary channel allocation for secondary links doesn't depend on whether primary channels are idle or busy but once they are assigned the interference does not cause disruption in both primary and secondary networks. A primary link has a primary channel to share (the numbers in braces in Figure 1) and one primary channel can be assigned to several secondary links (the number in brackets in Figure 1), as long as they, together, do not generate enough interference to disrupt the primary communication link. The secondary link selection depends on how much interference it can generate to those primary and secondary links that use the same primary channel. To determine the level of interference that any of the links experiences in the cognitive network, the equations (4) and (5) calculate the signal-to-interference-noise-ratio (SINR) value that the receiver either secondary or primary can suffer. The SINR at the secondary receiver u is

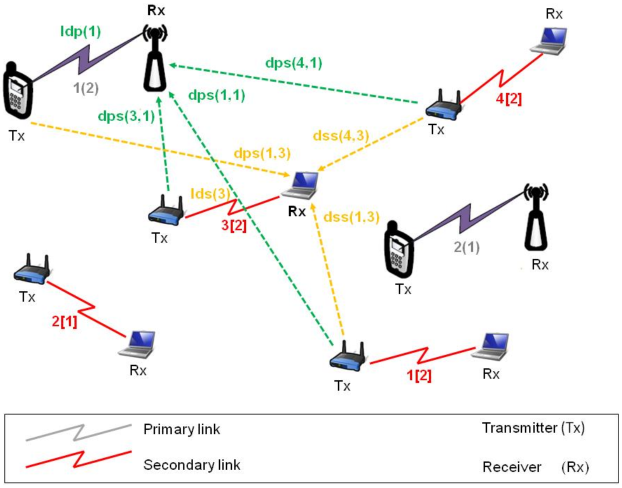


Fig. 1. System scenario.

given by:

$$SINR_u = \frac{P_u/lds(u)^n}{\sum_{k\in\varPhi} P_k/dss(k,u)^n + P_v/dps(v,u)^n}, 1 \leq u \leq Sl \tag{4}$$

where P_u is the transmit power of secondary transmitter u, P_k is the transmit power of secondary transmitter k, P_v is the transmit power of primary transmitter v, $lds(u)$ is the link distance of secondary link u, $dss(k,u)$ is the distance from secondary transmitter k to secondary receiver u, $dps(v,u)$ is the distance from primary transmitter v to secondary receiver u, k is the index of active secondary transmitters, $\varPhi$ is the set of active secondary transmitters, n is the path loss exponent (a value between 2 and 4). Similarly, the SINR at the primary receiver v is given by:

$$SINR_v = \frac{P_v/lpd(v)^n}{\sum_{k\in\varPhi} P_k/dps(k,v)^n}, 1 \leq v \leq Pl \tag{5}$$

where P_v is the transmit power of primary transmitter v, P_k is the transmit power of secondary transmitter k, $ldp(v)$ is the link distance of primary link v, $dps(k,v)$ is the distance from secondary transmitter k to primary receiver v.

Data rate contributions of the secondary links and primary links are calculated according to equations (6) and (7) respectively. The data rate depends on primary channel bandwidth B that secondary links and primary links can share and the conditions of the propagation environment (attenuation and interference).

$$c'_u = Blog_2(1 + SINR_u) \tag{6}$$

$$c''_v = Blog_2(1 + SINR_v) \tag{7}$$

Based on the above discussion, the admission and interference control problem is formulated as the following optimization problem:

$$Max \sum_{u=1}^{Sl} c'_u x_u + \sum_{v=1}^{Pl} c''_v \tag{8}$$

s.t.

$$SINR_u \geq \alpha \tag{9}$$

$$SINR_v \geq \beta \tag{10}$$

$$c'_u > 0, \qquad u = 1, 2, \ldots, Sl \tag{11}$$

$$c''_v > 0, \qquad u = 1, 2, \ldots, Pl \tag{12}$$

$$c'_u, c''_v \in R^+ \tag{13}$$

$$x_u = \begin{cases} 1, & if \;\; SINR_u \geq \alpha \;\; and \;\; SINR_v \geq \beta \\ 0, & otherwise \end{cases} \tag{14}$$

By observing the above optimization problem, the objective function is to maximize the sum throughput in the cognitive network (8), subject to the SINR requirements of the secondary links (9) and primary links (10). The maximum interference level is limited by α in the secondary network and β in the primary network in the right-hand side of each of the constraints (9) and (10). Constraints from (11) to (13) are integrity restrictions. $x_u = 1$ if secondary link u is included in the solution and $x_u = 0$ if it remains out as indicated in (14).

3.1 Solution Procedure Based on SCPSO Algorithm

The goal by using SCPSO is to decide which secondary links can achieve this, finding a binary vector P_g of size Sl representing the solution, where the bits $1/0$ symbolize if the $u-th$ secondary link is selected as part of the solution (bit 1) or not (bit 0). The maximum data rate achieved in the system is $f(P_g)$.

Assume S as the number of particles and D as the dimension of particles. A candidate solution is expressed as $X_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]$ where $x_{id} \in \{0, 1\}$. Velocity is $V_i = [v_{i1}, v_{i2}, \ldots, v_{iD}]$ where $v_{id} \in [-V_{max}, V_{max}]$. The personal best evaluation (pbest) of the i-th particle is denoted as $P_i = [p_{i1}, p_{i2}, ..., p_{iD}]$ where $p_{id} \in \{0, 1\}$. g is the index of the best particle in the swarm, therefore P_g is the best evaluation in the swarm (gbest). The swarm is manipulated according to the following velocity v_{id} and position x_{id} equations:

$$v_{id} = wv_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \tag{15}$$

$$v_{id} = w^1 v_{id} + c_3 (gbest - pbest) \tag{16}$$

$$x_{id} = x_{id} + v_{id} \tag{17}$$

$$x_{id} = x_{id} mod(2) \tag{18}$$

where w and w^1 are considered the inertia weights, c_1 and c_2 are the learning factors, c_3 is called as socio-cognitive scaling parameter, and finally r_1 and r_2

are uniformly distributed random numbers in [0,1]. Algorithm 1 is a simplified version from work presented in [14] to address the spectrum underlay problem in cognitive networks.

Algorithm 1: SCPSO solution to solve the spectrum underlay problem.

```
  Data: $Sl$, $Pl$, $\alpha$, $\beta$, $S$, and $V_{max}$
  Result: $P_g$, $f(P_g)$
1 initialization;
2 repeat
3     for $i$= 1 to number of particles do
4         Update pbest
5         Update gbest
6         Update $x_{id}$ and $v_{id}$ using equations (15) to (18)
7         if $x_{id} = 1$ then
8             allocate randomly a new channel to $x'_{id}$ from the set $PC$
9 until stopping criterion met;
```

Initialization stage includes: 1) locate randomly Sl and Pl in the scenario, 2) initialize randomly X_i, 3) initialize randomly V_i, 4) Set $P_i = X_i$, 5) Set $P'_i = X'_i$, and 6) initialize randomly vector Spectrum Status with values from Pl. Note that in initialization stage, P_i and X_i are considered to coincide. Three new vectors X'_i, P'_i, and *Spectrum Status* are included additionally. X'_i provides the possible channel allocation for secondary links. P'_i stores the best channels allocations found so far for a particle and *Spectrum Status* vector stores the channel allocations for primary links.

In update pbest (step 4 in Algorithm 1), the particle compares $f(X_i) > f(P_i)$ and overwrites pbest if $f(X_i)$ is higher than $f(P_i)$. In contrast, in update gbest, all pbest values will be compared with gbest value, so if there is a pbest which is higher than the gbest, then gbest will be overwritten. Update pbest and gbest phases require fitness calculation according to (8); to avoid infeasible solutions in the swarm, they are penalized by setting total particle's fitness to zero therefore they are not chosen in the selection process. Further details and the complete implementation of this solution procedure based on the SCPSO algorithm are provided in [14].

3.2 Quantitative Parameters: Number of Iterations and Inertia Weight

The SCPSO parameters of interest in this paper are **the number of iterations** and **inertia weight**. The inertia weight w influences the trade-off between exploration and exploitation [15]; therefore, a large w facilitates exploration, while a smaller w tends to facilitate exploitation in promising regions. Finding a suitable w helps to require fewer number of iterations on average to find the optimum value [15]. We took the reference values suggested by analogy from [8], except for the number of iterations and swarm size which are derived from an empirical tuning methodology (see Table 1); those values and *HATp* P^* were tested in the SCPSO algorithm to know their utility; it is important to denote that both parameter vectors had same values for parameters indicated (*) in Table 1.

Table 1. Parameter values.

Parameter	**Value**
Number of Secondary Users(*)	15,20,25 and 30
Number of Primary Users(*)	1
Number of Particles(*)	40
Number of iterations	150
Maximum velocity(*)	6
Minimal Velocity(*)	-6
Inertia Weight	0.721000

Taking as pivotal values, the *Number of iterations* and *Inertia weight* showed in Table 1; we define a *200 hundred percent rule* to state thresholds around them. It is a precondition in CALIBRA to define a searching area for promise regions.

Using aforementioned thresholds, CALIBRA defines a set of P vectors which are used to configure the set I' of problem instances; finally after testing P vectors on each problem instance in I'; CALIBRA brought a P^*. The combination of α, β and the Number of secondary users is used by CALIBRA to find P^*. Note that α and β are considered to coincide. Tables 2 and 3 show the entire design points used to configure SCPSO algorithm to resolve the set of problem instances I.

Table 2. Number of iteration values brought by CALIBRA.

Number of Iterations		Number of Secondary Users			
		15	20	25	30
α, β (dB)	4	48	198	168	162
	6	228	102	128	142
	8	96	93	145	227
	10	122	222	221	246
	12	31	201	199	14
	14	145	197	258	82

The range of w values brough by CALIBRA contained values gave in [8] $w = 0.721000$, and [15] $w = 0.8$ as show in Table 3. On the other hand, *the number of iterations* have differences, in [15] authors proposed up to 2500 iterations, the empirical tuning result was 150, and the values brough by CALIBRA between 40 and 250 for the number of iterations (see Table 2).

Table 3. Inertia weight values brought by CALIBRA.

Inertia weight		Number of Secondary Users			
		15	20	25	30
α, β (dB)	4	0.72309	0.81257	0.79076	0.78843
	6	0.62500	0.82340	0.88750	0.70429
	8	0.86409	0.76332	0.95552	0.93976
	10	0.85324	0.88784	0.89460	0.80000
	12	0.71726	0.80693	0.80002	**0.11093**
	14	0.88921	0.77942	0.94360	**0.45000**

4 Results

The aim of this experimental study is to know how much the SCPSO algorithm performance is affected by P^* brought by *TA-empirical tuning* and *HATp* processes. Tables 4 and 5 show SCPSO algorithm results using 30 different design points defined by parameter values in Tables 2 and 3; those design points were tested 1000 times; The characteristics of the computer equipment and software used were: a)Fine Tuner Tool, Calibra; Language, Borland C++, version 5.02; Operating system, Windows 7 enterprise 32 bits; Processor, Intel(R) Core (TM) i5-2320 CPU@3.00 GHz, and the RAM memory, 4.00 GB.

Analysing the SCPSO mean throughput in Table 5; it was higher when the SCPSO algorithm used the *TA-empirical tuning* vector than *HATpI*; however, as the number of secondary users, α, β values increase, also increases the average throughput of the SCPSO using the HATpI P^*, up to 100%. Concluding, the TA-empirical P^* utility is better with low problem complexity, while HATpI P^* is better in scenarios with high problem complexity. In line 48 of Table 5 HATpI P^* had its worse performance, when the number of secondary users is eqauls to 30 and $\alpha, \beta =$ 14 dB, the highest problem instance complexity; due to fact that CALIBRA did not provide a P^*. About the maximum value for data rate, as problem complexity increase the utility of HATpI P^* as well. However the median parameter shows zero in both process.

The SCPSO algorithm performance in Table 4 is similar to the one shown in Table 5. Although, considering the average throughput, only in three cases the *TA* vector allowed SCPSO algorithm to bring better results.

A global view of results in Tables 4 and 5, show that as the problem complexity increases, the SCPSO algorithm performance degrades. This behaviour allow us to conclude that, taking higher thresholds for w and *Number of iterations* could be possible to find better P^* vectors. This conclusion is supported by [8] and [15] as well as CALIBRA exploration in similar areas, having SCPSO low performance on average fitness for entire problem instances.

5 Conclusions

In this paper, we analyse two parameter tuning procedures, specifically focusing on two quantitative parameters of SCPSO which resolves the spectrum sharing

Table 4. Tuning by analogy versus Interpolation P_i^* SCPSO results.

	Design Point	Mean	Standard Deviation	Q1	Median	Q3	Maximum
1	4-17	696.8936	216.1854	561.3409	685.2271	825.2401	**1579.2489**
2	4-17-HATpII	705.1499	**190.4909**	580.8800	697.3810	820.5509	1336.9885
3	6-17	617.4922	**216.0871**	507.0424	624.5074	755.0196	**1360.8872**
4	6-17-HATpII	622.5656	223.6225	494.9839	626.1765	761.4327	1330.2744
5	8-17	536.3193	259.6028	409.4786	551.6591	700.4633	1298.0047
6	8-17-HATpII	559.0188	**229.8708**	429.6519	568.1481	700.5008	**1452.1818**
7	10-17	395.5316	266.3749	200.4630	441.9605	580.0636	1288.5817
8	10-17-HATpII	530.5890	**188.5329**	412.0709	525.8743	651.5861	**1333.6288**
9	12-17	243.9879	255.2588	0	244.8208	450.1429	**950.3957**
10	12-17-HATpII	168.5188	**209.0008**	0	0	330.8397	774.7756
11	14-17	135.0718	**199.5533**	0	0	288.34135	**863.5317**
12	14-17-HATpII	168.5188	209.0008	0	0	330.83975	774.7756
13	4-22	577.0828	314.40666	438.41045	625.3864	784.41305	**1502.3393**
14	4-22-HATpII	665.16352	**238.7602**	538.10245	666.6439	807.373	1390.4193
15	6-22	424.30708	340.36649	0	508.0161	683.1996	**1557.0939**
16	6-22-HATpII	579.12143	**267.8317**	452.0497	594.8837	743.12315	1527.0990
17	8-22	253.58628	**319.7857**	0	0	542.4337	**1368.1840**
18	8-22-HATpII	428.06549	308.7124	0	497.0066	651.2654	1354.8228
19	10-22	121.20538	**235.4958**	0	0	0	1107.1568
20	10-22-HATpII	404.11412	275.4165	191.6255	445.3831	598.9975	**1116.7342**
21	12-22	43.86616	**140.1225**	0	0	0	717.4726
22	12-22-HATpII	197.78717	258.8344	0	0	413.1131	**1072.9170**
23	14-22	17.44797	**84.5748**	0	0	0	806.7283
24	14-22-HATpII	98.76469	189.0751	0	0	0	**792.4581**
25	4-27	257.6821	**354.0673**	0	0	604.3964	1226.8559
26	4-27-HATpII	454.0794	375.7800	0	543.2218	751.57415	**1407.41**
27	6-27	124.4342	**276.2780**	0	0	0	1327.9231
28	6-27-HATpII	359.5787	358.58486	0	406.1548	656.4678	**1521.9564**
29	8-27	62.9879	**201.1050**	0	0	0	**1236.3037**
30	8-27-HATpII	233.1508	280.9179	0	0	484.8050	1035.4888
31	10-27	17.1658	**97.0037**	0	0	0	849.8469
32	10-27-HATpII	111.6993	232.5512	0	0	0	**953.7544**
33	12-27	6.0098	50.0705	0	0	0	641.0816
34	12-27-HATpII	2.70154	**41.9489**	0	0	0	**835.5134**
35	14-27	1.49241	**27.7714**	0	0	0	605.9482
36	14-27-HATpII	12.7723	74.91958	0	0	0	**688.865**

problem. A number of experiments are performed with different design points. Simulation results show that when Inertia weight is lower than 0.5 and the number of iterations=14 the SCPSO performance is low, therefore we conclude that an inertia weight = 0.8 is a good low threshold for this parameter. Consequently the high threshold should be modified up to find a suitable value to cope with more complex problem instances. Works [8] and [15] support the above observation, since authors show their exploration process to derive parameter values; however, they are not good for the present problem as its complexity increases.

On the other hand, *HATp* can provide better parameter values that improves the search ability of SCPSO to find a solution, enhancing its performance on

Table 5. Tuning by analogy versus CALIBRA P_i^* SCPSO results.

	Design Point	Mean	Standard Deviation	Q1	Median	Q3	Maximum
1	4-15	682.9017	**181.5390**	556.0250	673.705	794.495	**1318.43**
2	4-15-HATpI	659.3607	187.8980	527.1200	648.67	781.575	1303.3
3	6-15	635.4864	**191.7692**	504.0000	628.34	759.725	**1364.72**
4	6-15-HATpI	606.9634	221.6317	482.3050	621.485	745.45	1217.35
5	8-15	587.4071	**214.8417**	458.8550	590.33	707.28	**1492.48**
6	8-15-HATpI	145.8720	273.8555	0	0	0	1211.07
7	10-15	492.0968	223.2007	365.1100	501.86	627.47	**1154.4**
8	10-15-HATpI	532.4548	**175.3387**	413.64	522.845	629.95	1157.38
9	12-15	351.1337	**227.8941**	222.205	388.205	511.765	**1124.26**
10	12-15-HATpI	320.7400	231.2092	0	356.935	477.86	1016.81
11	14-15	220.2125	211.9554	0	251.25	376.045	**1006.27**
12	14-15-HATpI	367.0684	**159.9957**	271.54	348.345	455.86	938.5
13	4-20	654.1262	258.6153	518.245	667.765	823.59	1318.28
14	4-20-HATpI	704.7995	**214.5675**	571.155	698.325	847.28	**1489.07**
15	6-20	514.2564	315.9078	351.965	571.44	731.145	1388.26
16	6-20-HATpI	620.3161	**229.7880**	496.035	626.455	753.34	**1471.96**
17	8-20	364.8977	319.4826	0	427	621.625	**1349.14**
18	8-20-HATpI	467.7069	**294.4225**	331.835	517.33	673.465	1319.95
19	10-20	216.8342	285.9504	0	0	471.45	**1248.8**
20	10-20-HATpI	462.6305	**225.4804**	356.25	485.155	604.185	1056.78
21	12-20	107.2892	**210.5406**	0	0	0	1058.84
22	12-20-HATpI	259.1259	264.9015	0	276.805	477.35	**1088.38**
23	14-20	40.8101	**128.3257**	0	0	0	868.99
24	14-20-HATpI	89.0967	173.724	0	0	0	**986.5**
25	4-25	373.3844	367.6206	0	431.3250	674.8550	**1520.4800**
26	4-25-HATpI	556.9658	**317.0070**	446.455	600.855	764.27	1378.4100
27	6-25	217.1393	322.6422	0	0	514.75	1307.2000
28	6-25-HATpI	459.6812	**326.6541**	0	532.61	698.265	**1611.7400**
29	8-25	97.4385	234.07921	0	0	0	1163.8100
30	8-25-HATpI	386.4427	**300.07269**	0	452.31	617.46	**1214.9300**
31	10-25	44.0648	157.0061	0	0	0	967.7300
32	10-25-HATpI	204.6662	**285.2063**	0	0	460.0800	**1232.2100**
33	12-25	12.6148	79.9636	0	0	0	814.1200
34	12-25-HATpI	43.7149	**145.9721**	0	0	0	**990.7600**
35	14-25	40.8101	128.3257	0	0	0	868.9900
36	14-25-HATpI	60.1733	**158.4422**	0	0	0	**1075.4600**
37	4-30	126.9177	**282.2731**	0	0	0	1264.15
38	4-30-HATpI	265.2456	360.2032	0	0	608.92	**1453.76**
39	6-30	55.7050	194.6109	0	0	0	**1326**
40	6-30-HATpI	41.4461	**170.6329**	0	0	0	1110.18
41	8-30	20.3758	**115.4815**	0	0	0	920.09
42	8-30-HATpI	145.8720	273.8555	0	0	0	**1211.07**
43	10-30	6.0608	**59.3307**	0	0	0	793.88
44	10-30-HATpI	14.0721	96.6330	0	0	0	**1053.18**
45	12-30	1.2604	23.8519	0	0	0	568.03
46	12-30-HATpI	2.8047	**32.7819**	0	0	0	**618.89**
47	14-30	0.4429	9.9828	0	0	0	**250.14**
48	14-30-HATpI	0	0	**0**	0	0	0

resolving the spectrum sharing problem, than those parameters values suggested by TA and empirical methodology on some problem instances. This encourage us to analyse other regions using $HATp$; in intention to find better P^*. Our interest is also to analyse another automated tuning procedures as ParamILS to gather information about how parameter values affect the SCPSO algorithm performance.

References

1. Eiben, A., Smit, S.: Parameter tuning for configuring and analyzing evolutionary algorithms. Swarm and Evolutionary Computation **1** (2011) 19 – 31
2. Hutter, F., Hoos, H.H., Leyton-Brown, K., Murphy, K.P.: An experimental investigation of model-based parameter optimisation: Spo and beyond. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation. GECCO '09, New York, NY, USA, ACM (2009) 271–278
3. Montero, E., Riff, M.C., Neveu, B.: A beginner's guide to tuning methods. Applied Soft Computing **17** (2014) 39 – 51
4. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Neural Networks, 1995. Proceedings., IEEE International Conference on. Volume 4. (1995) 1942–1948 vol.4
5. Parsopoulos, K., Vrahatis, M.: Particle Swarm Optimization and Intelligence: Advances and Applications. Premier reference source. Information Science Reference (2010)
6. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001)
7. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. In: Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on. Volume 5., IEEE (1997) 4104–4108
8. Deep, K., Bansal, J.C.: A socio-cognitive particle swarm optimization for multidimensional knapsack problem. In: Proceedings of the 2008 First International Conference on Emerging Trends in Engineering and Technology. ICETET '08, Washington, DC, USA, IEEE Computer Society (2008) 355–360
9. Adenso-Diaz, B., Laguna, M.: Fine-tuning of algorithms using fractional experimental designs and local search. Oper. Res. **54** (2006) 99–114
10. Autar Kaw, E.E.K.: NUMERICAL METHODS WITH APPLICATIONS: Abridged. autarkaw.com (Licencia estndar de derechos de autor) (2011)
11. Coy, S.P., Golden, B.L., Runger, G.C., Wasil, E.A.: Using experimental design to find effective parameter settings for heuristics. Journal of Heuristics **7** (2001) 77–97
12. Bartz-Beielstein, T.: How experimental algorithmics can benefit from mayo's extensions to neyman-pearson theory of testing. Synthese **163** (2008) 385–396
13. Dobslaw, F.: Recent development in automatic parameter tuning for metaheuristics. In: Proceedings of the 19th Annual Conference of Doctoral Students - WDS 2010. (2010)
14. MartíNez-Vargas, A., Andrade, A.G.: Comparing particle swarm optimization variants for a cognitive radio network. Appl. Soft Comput. **13** (2013) 1222–1234
15. Shi, Y., Eberhart, R.C.: Parameter selection in particle swarm optimization. In: Proceedings of the 7th International Conference on Evolutionary Programming VII. EP '98, London, UK, UK, Springer-Verlag (1998) 591–600

An Ant Colony System Metaheuristic Algorithm for Solving a Bi-Objective Purchasing Scheduling Problem

José Francisco Delgado Orta[1], José Antonio Coronel Hernández[1], Laura Cruz Reyes[2],
Alejandro Palacios Espinosa[3], Christian Ayala Esquivel[1],
Isidro Moctezuma Cantorán[1], and Jorge Ochoa Somuano[1]

[1] Universidad del Mar, San Pedro Mixtepec, Juquila, Oaxaca,
Mexico

[2] Instituto Tecnológico de Ciudad Madero, Ciudad Madero, Tamaulipas,
Mexico

[3] Universidad Autónoma de Baja California Sur, La Paz, Baja California Sur,
Mexico

fdelgado@zicatela.umar.mx, lexmar171@hotmail.com, cruzreyeslaura@gmail.com,
palacios@uabcs.mx, ceaaa@hotmail.com, moctezumai@hotmail.com,
ochoa@zicatela.umar.mx

Abstract. In this work the Purchasing Scheduling Problem (PSP) is presented, based in study cases of public and private sectors. PSP models the purchasing process in both scenarios through an optimization approach. It is based on the multi-objective formulation of the knapsack problem. Therefore, PSP is defined as a based-graph problem with two objectives: maximization of satisfied demands and minimization of purchasing costs in a supplying task for inventory systems. In order to achieve these goals, the problem is defined as an integer problem, in which, feasibility of solutions is tested using a profit/cost relationship. This permits to solve PSP as a maximization of a single objective through an Ant Colony System Algorithm (ACS), an efficient solver for graph-based problems. Experimental results reveal that ACS reaches 74% of efficiency on solving instances randomly generated; obtaining purchasing plans as a result. This demonstrates the advantages of using heuristics in decision making systems such as Enterprise Resource Planning (ERP).

Keywords: Purchasing scheduling problem, ant colony system algorithm, multi-objective optimization.

1 Introduction

The purchasing process of materials and goods is very important for companies and organizations around the world. It is a main task for maintaining inventory control in warehouses which arise from customer demands and whose main objective is to provide goods to warehouses in a timely manner to satisfy all demands in periodic inventory cycles. When a company decides to purchase materials or equipment, it usually considers a set of variables related to inventory theory: stock levels, delivery times, reorder points and in most cases the availability of money to make purchases. However, according to [1], inventory systems in purchasing departments have poor planning tools and strategies for decision making in inventory tasks, as a consequence of supply chain problem related to inventory is defined as a NP Hard problem, where optimization of tasks is difficult to perform in large-scale instances.

Purchasing is a complex process that involves searches in big catalogs (or huge in the worst-case scenario) or even physical inventories, commonly found in websites or company data warehouses, which are dependent to needs in several companies and the available physical accommodation where they are stored. Industrial applications of PSP, has also an economical constrain of available funds and scenarios where a purchasing department implies the work of many people in its different roles since an assistant until the managers. Where they must decide which of inventory variables are the most relevant for the company according to priorities established by customers and prices of products. In this manner, this work approaches a planning stage for purchasing based on an economical approach, which is mathematically modeled and solved using heuristic methods.

2 Purchasing Scheduling Problem: Public and Private Problematics

Purchasing Scheduling Problem (PSP) is related to many companies in industrial, educational, and many other applications, where there are two main areas: private and public sectors. In the private sector, orders are programmed periodically in daily, weekly, monthly or yearly cycles. For each period, the supply manager authorizes company funds so that purchasing assistants can supply the goods according to reorder levels defined for each case. In the public educational sector, it is done in the same manner with the purchasing department authorizing funds to fill orders that must be prepared by personnel for each random period. Purchasing is quite different in the public sector, because it must be done through an open bidding process (defined in [2]) which must meet the rules and laws about purchases in public entities.

Additionally, in public education environment, this process requires order preparation with varying product quantities, which are commonly supplied at different times of the year because funds are not delivered periodically to managers. In addition, needs of customers in both sectors are sometimes unmet due to shipping delays, lack of suppliers' stock; whereas in the public educational sector these issues are present but mainly due to quality factors as schools' purchasing department personnel is not an expert in the technical requirements of the goods to be acquired. In addition, goods require a high description factor because suppliers are not to be taken into account when orders go out for bidding as required in [2].

The pressure for purchasing often leads to bad quality products, as expeditiousness is preferred over quality. In addition, order preparation usually must be done in haste. As a consequence, orders are usually not ready and often lack complete descriptions or specs but are still turned over to the supply department for purchasing. In addition, complexity elements of described approaches are often a barrier to include mathematical formulations in real-time inventory systems.

3 Related Work

Several applications of PSP present the problem according to the satisfaction of a function of time as a single objective, dependent of other variables; for example, holding times and penalties in project management environment [3], where the problem depends on the availability of products by the suppliers to supply independent items. In [4], is presented a purchasing scheduling in manufacturing systems, based on the computation of delivery times and quantities of product to supply items using special priorities, according to resource capacities, orders priority and leading times. In the related problem of economic lot size scheduling problem [5], time is also a decision factor according to inventory variables. However, a common feature of these approaches is influenced by no linear and unrestricted models to solve them, supposing that funds to supply the items are unlimited and planning is performed in stochastic way. Even though, multi-objective approaches to solve PSP using linear models are not reported in literature. Related works have solved similar multi-objective problems such as networking Flow Shop Scheduling [6] and the Knapsack Problem [7]. For these problems, algorithms have been developed using approaches like Ant Colony based on Pareto's optimization [8], strategies of multi-objective Particle Swarm Optimization [9], Fuzzy Ant Colony Optimization [10] or Genetic Algorithms [11]. In where, problems are formulated as a graph based-representation to generate feasible solutions through linear approaches.

According to [11], solution of multi-objective problems presents a behavior called Pareto optimality (named Pareto's front), in which a solution of a problem is represented by a set of objectives that are commonly in conflict, due to the non-dominance of objective values. In other words, if an objective is approximated to optimal value, remaining objectives can be away of their respective optimal values, or even they may be infeasible.

4 The Graph-based Representation of PSP

The Purchasing Scheduling Problem is defined from a general catalog of products in terms of a graph $G = (V,E)$, where $V = \{P \cup S\}$ consists of a set of n products (P) that must be acquired from m suppliers (S). The set E consists of pairs (p, s), where $p \in P$ and $s \in S$; each pair has an associated economic cost c_{ps} of a product to be purchased from any supplier.

Purchasing must be performed through orders, which are defined as subsets G_k, where k corresponds to the number of purchasers, who are the decision makers to get the products. Each order G_k must be formed according to an available fund a_k for each purchaser. The problem is formulated through two objectives: maximization the amount of products that must be satisfied and minimization of purchasing costs in an inventory cycle. So, these elements are the basis to perform a mathematical approach of PSP.

4.1 The Purchasing Scheduling Formulation

Solution of PSP is performed to represent purchasing plans in an inventory cycle. In this way, PSP is formulated through the next data sets:

- *P:* is the set of products to be supplied, in other words an inventory catalog with *n* products.
- *S*: is the set of suppliers of the product catalogue, requested by a purchaser with *m* suppliers.
- *A*: is a set of available funds for s purchasers, with a number a_k for each order *k*, where each non negative fund a_k is assigned to a purchaser.

PSP formulation uses the next variables:

- c_{ij} is the economic cost for each product *i* of the supplier *j*.
- x_{ijk} is a decision integer variable with values {0,1}, one if the product *i* is assigned to the supplier *j* in the order *k*, zero in otherwise.

In the context of the problem, *f* and *g* values are introduced to normalize objectives values of PSP in the domain [0,1], as the multi-objective knapsack problem [7]. Where *f* represents a profit in terms of assigned products, while g is a vector that indicates a reference of an assigned product with regard to the c_{ij} values of products to be supplied in an inventory cycle. These values are based on the utility principle proposed in [7], [14], [15], which is defined in terms of PSP through expressions (1) and (2).

$$\max f = 1 - \frac{1}{k}\sum_{k=1}^{s}\left[\frac{\sum_{j=1}^{m}\sum_{i=1}^{n} x_{ijk}}{|G_k|}\right] \quad (1)$$

$$\min g = \frac{1}{k}\sum_{k=1}^{s}\left[\frac{\sum_{j=1}^{m}\sum_{i=1}^{n} c_{ij}x_{ijk}}{\sum_{j=1}^{m}\sum_{i=1}^{n} c_{ij}}\right] \quad (2)$$

These values define a profit/weight relationship that heuristics are able to exploit for objectives of PSP. Therefore, the normalized g value is transformed to solve the multi-objective PSP as a maximization problem in the next formulation:

$$\max z = f - g \quad (3)$$

Subject to:

$$\sum_{j=1}^{m}\sum_{i=1}^{n} c_{ij}\, x_{ijk} \le a_k \quad k = 1,2,\cdots,s \quad (4)$$

$$x_{ijk} \in \{0,1\} \quad i = 1,2,\cdots,n, j = 1,2,\cdots,m, k = 1,2,\cdots,s \quad (5)$$

Objectives of the problem are related with equation (3). Equations (4) and (5) verify constrains of assigned funds and with regard to the available resources according to integer values x_{ijk}. This model is solved with an ant colony system algorithm, which obtains feasible solutions according to the related ant approach.

5 The Ant Colony System Algorithm

Ants in their evolutionary process, search feasible solutions of minimal cost for a problem. During the execution of an ant algorithm, an ant located in a state (a selected product *i*, has to choose the next supplier *j* to be assigned. Suppliers are chosen from among the set of unassigned products in the rest of suppliers. Selection of the next supplier *j* uses a probability value q_0, which is used in deterministic or random approaches. In a deterministic form, selecting the supplier with the greater amount of pheromone τ, following an exploitation of the obtained knowledge of the problem η using the neighborhood $N_k(i)$ for a selected product *i*; while in random selection, a rule of exploration is used. Both focuses are represented in equation (6).

$$j = \begin{cases} argmax_{u\in N_k(i)}\left\{[\tau(i,j)] \times \left[n(i,j)^{\beta}\right]\right\} & q < q_0 \\ f(p_{ij}^{k}) & otherwise \end{cases} \tag{6}$$

Where q is a uniformly distributed random number between [0,1], q0 is a constructive parameter 0 <q0< 1, which represents a probability value and f is a function based on the well-known roulette technique, which selects the supplier for a product with a probability proportional to $f(P_{ij}^{k})$, obtained as a result of the computation of expression (7). The basic idea of this algorithm is to choose different suppliers at each iteration. Heuristic information n_{ij} is the previous knowledge, given through the inverse of accumulated cost over the pairs (i, j). Heuristic information n_{ij} of equation and pheromone trails τ_{ij} are used by ACS to guide all ant movements.

$$p_{ij}^{k} = \begin{cases} \frac{[\tau_{ij}][n_{ij}]^{\beta}}{\sum_{l\in N_i^k}[\tau_{il}][n_{il}]^{\beta}} & j \in N_k^i \\ 0 & otherwise \end{cases} \tag{7}$$

Where p_{ij}^{k} is the probability value, associated for an ant *k* that selects a product *i* from a supplier *j*. N_k^i is the possible neighborhood for an ant *k* when it choose a product *i*; in other words, all reachable suppliers. Pheromone trails, deposited by ants evaporate in local and global forms. Local evaporation is given through equation (8), where τ_0 is the initial value of the pheromone trail and $\rho(0 < \rho < 1)$ is the rate of local dissipation.

$$\tau_0 \leftarrow (1-\rho)\tau_{ij} + \rho\tau_0 \tag{8}$$

Evaporation of global pheromone follows the pattern of behavior given by equation (9).

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \rho\Delta\tau_0, \quad \forall(i,j) \in T_k \tag{9}$$

Where $\Delta\tau_0 = {}^{1}/_{C_k}$ represents the amount of deposited pheromone and ρ is a rate of global evaporation, and C_k is the accumulated resources of the packed products performed by an ant for all arcs in an order T_k. Artificial ants always try to build feasible solutions using pheromone trails; however, infeasible solutions also can be obtained [12]. General algorithm ACS is shown in Figure 1.

AntColonySystem procedure consists of the construction of ants and activities for these and global evaporation of pheromone trails. Construction of a colony is performed through the *Generate_ants_and_activities* procedure, shown in Figure 2.

```
Procedure AntColonySystem ()
    Initialize_parameters
     While(stop_criteria_is_reached) do
         Generate_ants_and_activities()
         Evaporation_pheromone_trails()
      End_of_while
End Procedure
```

Fig. 1. The AntColonySystem Procedure.

```
1   Procedure Generate_ants_and_activities ()
2     For each ant $k$ until $m$
3       $i$=current_product
4       $S_k = i$
5       $L_k = i$
6       While (current_state is not equal to objective_state) do
7           For each $j \epsilon N_k(i)$
8               $b_{ij} = \tau_{ij} \times n_{ij}^{\beta}$
9           $q$ = generate_random_number[0,1]
10          If ($q$<$q_0$) then
11              $j = \max(b_{ij}, N_i^k)$
12          Else
13              For each $j \in N_i^k$
14                                    $p_{ij}^k = \frac{b_{ij}}{\sum_{l \in N_i^k} b_{il}}$
15              End For
16              $j = f(P, N_i^k)$
17            End If
18        $i$ = next_product
19        $S_k =< S_k, j >$
20        $\tau_0 \leftarrow (1-\rho)\tau_{ij} + \rho\tau_0$
21        $L_k = L_k \cup \{j\}$
22      End_while
23    Next ant $k$
24  End Procedure
```

Fig. 2. The Generate_ants_and_activities procedure.

Algorithm in Figure 3, is used with the described components of ACS (lines 2-23), for each ant k. Lines 4-6 choose the current product and initializes the solution. Selection of the correspondent supplier j for a product i is performed in lines 7-18 according to the defined transition rules. Line 16 obtains the supplier j for the current product i using a policy of decision, in which the values of pheromone trails for the current product and restrictions of the problem are considered according to the P values of equation (6). Lines 18-20 perform the state transition, the upgrading of a solution S_k and the evaporation of pheromone trails τ_{ij}. Line 21 includes to the algorithm a memory L_k, which storages the current assignation. It is used to build valid solutions, to evaluate a generated solution and to make a reallocation for an ant if a supplier j chosen for a product, makes infeasible the current solution.

6 Architecture of the Solution

The described ACS is proposed according to the architecture of Figure 3, which describes the construction of a purchasing schedule.

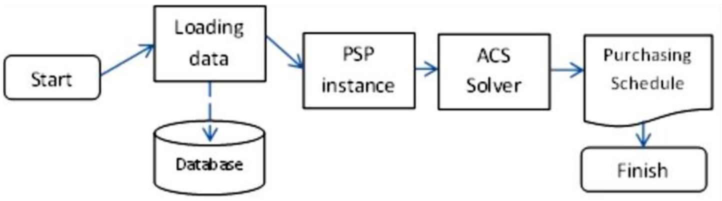


Fig. 3. Proposed solution methodology.

This architecture is designed validate the performance of heuristic approach solving PSP. For this reason, it is supposed that there is an inventory database, which contains information about the sets defined in the PSP formulation, this information is read to a model database described in [13]. Once that sets are read to the database, input sets are placed in a PSP instance. These inputs consist of a plain-text file that makes the solver independent to the database. This permits the use of the methodology in several purchasing scenes. PSP inputs are used by the solver to build purchasing plans according to the objectives through the described ACS algorithm, defined in terms of an inventory cycle.

7 Experiments and Results

Due to real instances for the problem were not available, a dataset of ten orders was built using a pseudo-random number generator. It uses the queries of web catalogs, stored in a database. The generator creates the orders with different prices and suppliers for products, maximum and minimum prices for products and available funds, parameters of generator are shown in Table 1. The ten generated orders (it is supposed an inventory cycle with ten purchasers) are the input sets for PSP, whole products have sometimes costs more expensive than available funds.

Instances were solved with ACS algorithm in a server Dell T710, Xeon four-core processor (2.2 Ghz per core), 48 GB of RAM memory, 500 GB hard disk under Debian Linux 6.0. Algorithm was developed in Java Standard Edition 7 with Eclipse Helios. Pheromone trails of ACS have initial values of 0.005, a parameter that according with [12], which permits enough exploration. Evaporation rate ρ is 0.5, and q value of ACS is 0.5, which permits to have 50% of exploration and 50% of exploitation. Algorithm design consists of 20 generations and ten ants per colony with the *β* value of 1. In each execution, an accumulated sum (∑) is stored with the average of the values of best solutions.

Table 1. Input parameters for instances generated for PSP.

Order	Quantity of Products	Min price	Max price	Available Funds
1	126	14.00	10999.00	25000.00
2	123	73.00	60000.00	50000.00
3	63	29.00	120000.00	40000.00
4	146	99.90	15980.00	65000.00
5	70	3.00	60000.00	30000.00
6	194	56.90	20000.00	80000.00
7	128	75.00	18799.00	75000.00
8	119	95.00	18000.00	55000.00
9	108	3.00	88996.00	48000.00
10	126	14.00	11499.00	40500.00

Table 2. Results of ACS algorithm solving PSP.

Iterations	∑	*f*	*g*	*z*	*t*
1000	22.26	0.8436	0.1027	0.7408	585.15
5000	24.17	0.8412	0.1030	0.7382	4753.96
10000	23.32	0.8395	0.1030	0.7406	5830.87
15000	26.47	0.8425	0.1029	0.7407	6840.30
20000	25.22	0.8431	0.1030	0.7408	8963.43
30000	25.53	0.8441	0.1030	0.7409	10732.78

It indicates a measurement of heuristic information with regard of the solution space (the exploration degree of the algorithm). Results of ACS of table 2 show also the average time (*t*) in which ACS obtains the best solutions and the *z* value for each instance. Table 2 shows the performance for ACS for six tests with 30 executions with: 1000, 5000, 10000, 15000, 20000 and 30000 iterations. ACS has a convergence of 6.2 seconds in average, with a profit coefficient of 0.8423 (84% in average of normalized *f* values) and 0.1030 in cost coefficient (10% in average of *g* values). Values PSP objectives reaches 0.7403 in average (74%) of efficiency in the multi-objective problem.

It occurs because of ACS has finished its executions before assignations of products have been completed; as a result of many remaining products have expensive costs that make infeasible equation (2) and funds can not completely be assigned. In the same way, summation column has a coefficient of variation of (0.6%). It reveals that adjusting of ACS parameters can be recommended to explore new solutions. Best values can be obtained because of results of PSP objectives (*f* and *g* values) present a Pearson coefficient of correlation of -0.3826, which demonstrates the existence of Pareto's front in solutions of PSP.

8 Conclusions and Future Works

The purchasing scheduling problem was presented in this work with an economic approach in a combinatory scene. It was feasibly solved through an ant colony system algorithm and a heuristic based on the multi-objective knapsack problem. In this way, results of the developed ACS can be tested using other neighborhood techniques, such as 2-Opt and its variants or Cross Exchange. These strategies can be used to generate a bigger subset of feasible solutions to compute Pareto's optimal values for PSP. In same way, problem can be solved using other heuristics as taboo search, simulated annealing or genetic algorithms or they can be improved through hybridization of the developed ant colony approach. Optimality degree of PSP and speed of computation shows the advantages of heuristic approaches to integrate them in making decision systems can be used as planning tools to develop ERP systems (Enterprise Resource Planning), which can be applied in industrial applications to optimize productive processes in many companies.

References

1. Herroelen, W.: Project scheduling—Theory and practice. Production and Operations Management, 14(4), pp. 413–432 (2005)
2. Cámara de Diputados del H. Congreso de la Unión.: Ley de Adquisiciones, Arrendamientos y Servicios del Sector Público. Reforma (2011)
3. Ronen, B., & Trietsch, D.: Optimal scheduling of purchasing orders for large projects. European journal of operational research, 68(2), pp. 185–195 (1993)
4. Huang, P., Li, H., & Han, L.: Order scheduling problems in make-to-order manufacturing systems. In Mechatronics and Automation, 2005 IEEE International Conference. Vol. 4, pp. 2179–2184, IEEE (2005)
5. Sarker, R., & Newton, C.: A genetic algorithm for solving economic lot size scheduling problem. Computers & Industrial Engineering, 42(2), pp. 189–198 (2002)
6. Graham, R. L.: Bounds for certain multiprocessing anomalies. Bell System Technical Journal, 45(9), pp. 1563–1581 (1966)
7. Kumar, R., Joshi, A. H., Banka, K. K., & Rockett, P. I.: Evolution of hyperheuristics for the biobjective 0/1 knapsack problem by multiobjective genetic programming. In Proceedings of the 10th annual conference on Genetic and evolutionary computation. pp. 1227–1234. ACM (2008)
8. Doerner, K., Gutjahr, W. J., Hartl, R. F., Strauss, C., & Stummer, C.: Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. Annals of Operations Research, 131(1–4), pp. 79–99 (2004)

9. Reyes-Sierra, M., & Coello, C. C.: Multi-objective particle swarm optimizers: A survey of the state-of-the-art. International journal of computational intelligence research, 2(3), pp. 287–308 (2006)
10. George, A., & Rajakumar, B. R.: Fuzzy aided ant colony optimization algorithm to solve optimization problem. In Intelligent Informatics, pp. 207–215, Springer Berlin Heidelberg (2013)
11. Coello, C. C., Lamont, G. B., & Van Veldhuizen, D. A.: Evolutionary algorithms for solving multi-objective problems. Springer (2007)
12. Dorigo, M., & Gambardella, L. M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. Evolutionary Computation, IEEE Transactions on, 1(1), pp. 53–66 (1997)
13. Coronel, J.A.: Solution of Assignation Problem in Purchasing of Goods Using Metaheuristic Algorithms. Bachelor Thesis. Universidad del Mar campus Puerto Escondido (2014)
14. Falkenauer, E.: A hybrid grouping genetic algorithm for bin packing. Journal of heuristics, 2(1), pp. 5–30 (1996)
15. Martello, S., & Toth, P.: Lower bounds and reduction procedures for the bin packing problem. Discrete Applied Mathematics, 28(1), pp. 59–70 (1990)

Multi-Spatial Classifier for Blue Whale Images using Photo-Identification Method

Rosa I. Ramos-Arredondo[1], Blanca E. Carvajal-Gámez[2], Francisco J. Gallegos-Funes[3], and Diane Gendron-Laniel[4]

[1] Escuela Superior de Cómputo, Instituto Politécnico Nacional (IPN), Mexico

[2] Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, IPN, Mexico

[3] Escuela Superior de Ingeniería Mecánica y Eléctrica, IPN, Mexico

[4] Centro Interdisciplinario de Ciencias Marinas, IPN, Mexico

alesija@gmail.com, becarvajal@ipn.mx, fgallegosf@ipn.mx, dgendron@ipn.mx

Abstract. The process of photo-identification images of the blue whale (Balaenoptera musculus), is made manually; this process classifies images blue whale through its dorsal fin characteristics. The features are extracted visually, which can generate errors at the moment of classified. In this work an image classifier blue whale is presented, which have features such as color pigmentation, background image, type of dorsal fin, among others; these common characteristics generate high statistical dependence. This statistical dependence causes the data extracted through a segmented image of the blue whale, are to be observed through a hyperplane. Using statistical techniques and the K-Nearest Neighbors classifier, the classification of three types of dorsal fin is obtained with an accuracy of 71.66%, assessing the value of K = 7, taking reference catalog of species CICIMAR- IPN.

Keywords: Classifier, K-nearest-neighbors, feature extraction, multi-spatial classifier.

1 Introduction

The technique of photo-identification of marine mammals started in the 70's by the marine biologist, Michael Bigg. The technical manual is considered a noninvasive method, in addition to many advantages in identifying natural markings, temporary marks and scars such as: the shape of the fins, the location of it and coloring marks [1]. These features are unique between each marine mammals [2]. One of the features used to differentiate other is the shape of its dorsal fin (fin found on the back of the body of the whale) and caudal (fin located on the lower body of the whale). Specialists in marine mammals, when conducting field survey samples, need to identify the element of study in the first instance is through photographs where the sides of your body and how their fins are displayed. It is noteworthy that the blue whale shows its tail fin infrequently and is due to this factor that is convenient to take as the identifier element to its dorsal fin.

Photographs of the Blue Whale were mostly taken in the Gulf of California (CICIMAR-IPN) have the particularity to be taken at sea, having among the main challenges for this analysis: the seabed, natural camouflage techniques marine animals, luminosity and different angles of the image when the picture is taken. In order to classify the element study by the type of dorsal fin need to perform a pre-processing the image without altering the morphology of the same. Later, we require segmentation stage we would help to get the shape identifier.

The classifier will give us an estimate of fin type to which it belongs. The type classification is given by dorsal contour form [3]. For the study we will focus on only three types of fin (falcate, triangular and hooked). The catalog of CICIMAR-IPN has previously classified photographs by the shape of the flap and flanks.

Extracting features of the shape of the dorsal fin factors considered scaling, translation and rotation, for example; the photographs are captured at different distances (100-300 meters). In addition a large number of images from the same individual study for the selection and classification of features to identify needs.

The article is organized as follows: Section 2 shows the methodology for classification, Section 3 describes the Hu invariant moments for generating the feature vector we need classification identifier, Section 4 shows the details of the classification, and Section 5 discusses the experiments and the results obtained. Finally, Section 6 concludes this paper.

2 Methodology

In this paper previous stages are required to obtain samples that will engage in training the classifier and to classify samples. Divide into two phases: the first is called sample collection and the second classification of samples. The following figures illustrate each of the phases and stages that carry a flow of execution.

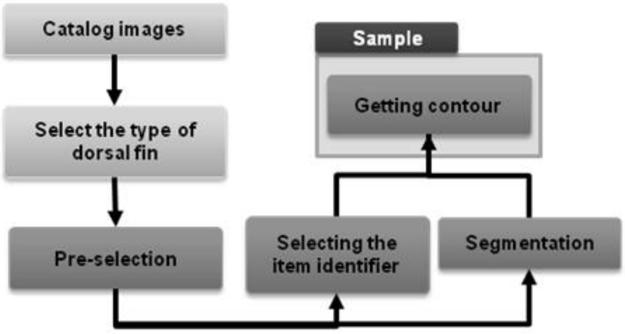


Fig. 1. First stage: sample preparation.

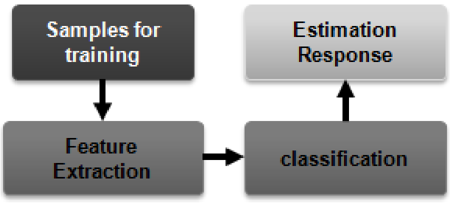


Fig. 2. Second stage: classification of samples.

2.1 Selecting the Type of Dorsal Fin

In the catalog of species-IPN CICIMAR we focus on the blue whale as identifying the dorsal fin is used. Photographs where the dorsal fin of the blue whale is displayed, regardless of the side will look. Another parameter in the selection list is the shape of the flap; for the study will consider three types of fins: falcate, triangular hooked. See in Figure 3, three examples of the ways of studying the dorsal fin.

Photos used within the context of article images from the call that will perform such a calculation from the pre-processing may consider the name of images. The images in the course of the study will be occupied but for the purpose of reducing the storage space on the calculation and standardize the treatment, we focus on a sample of the image where the identifier elements to classify (the dorsal fin whale blue).

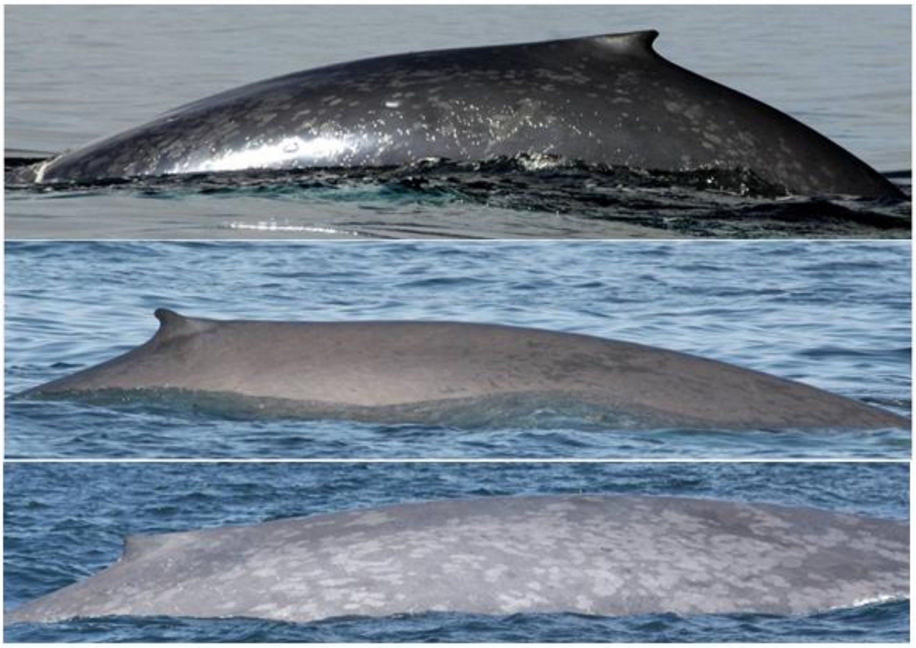


Fig. 3. Photographs of the dorsal fin.

2.2 Pre-Selection

In the stage of pre-selection we find photographs that were collected either with digital or SLR cameras. The pictures that were not taken with a digital camera was to grayscale and were digitized using a scanner to have them in digital format. To standardize the photographs to use images that were formed from three layers of color (RGB - Red, Green, and Blue) were selected.

2.3 Representation of a Photograph in a Digital Image

The picture is represented by an image f (x, y), where f represents the value of the light intensity at each point (x, y). .The x, y are the spatial coordinates. M, N, are the rows and columns of the digital image [4]. This step was necessary to delimit the number of factors to consider in the photographs and the following steps are done processing.

$$f(x,y) = \sum_{x=1}^{M} \sum_{y=1}^{N} (x,y) \tag{1}$$

2.4 Selecting the Item Identifier

The images of the blue whale's dorsal fin where shown are not the same size when they are harvested. Taking as an advantage point to classify blue whale by the shape of its dorsal fin, select a portion of the image where the louver and we can give all the relevant information in addition to normalizing the portion size.

2.5 Segmentation

In [4], [5] indicates that the segmentation is one of the most important for the extraction of features and / or steps in the image recognition step. Wanting to classify blue whale through the dorsal fin, leads us to want to find the shape of the fin to differentiate between forms of fins that were raised in the study (falcate, hooked, triangular). The portion of the image has the same problems that had full but with the advantage of being more amenable to their size image. When dealing with images of blue whales has some problem like the reflection of the sun on the skin of the whale, the blue whale color with the color of the sea. The segmentation of the dorsal fin is for localizing region-based active contours. They adapt the integration of a rectangular region centered on the image so that the number of iterations is low [6].

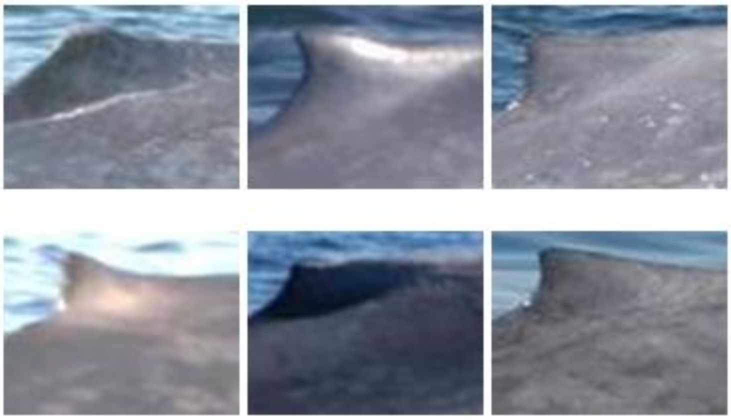


Fig. 4. Select of the dorsal fin. The images of the dorsal fin are the right flank of the blue whale. Some of the problems mentioned in the article shows.

2.6 Getting Contour

From the image with the segmented region have the dorsal fin separated from the image background. Obtaining the contour is to be used in stage two as a sample, wherein the extraction characteristics are performed. For this we need to use a technique that provides us the outline and the study sample can be represented in a binary image as shown in Figure 5.

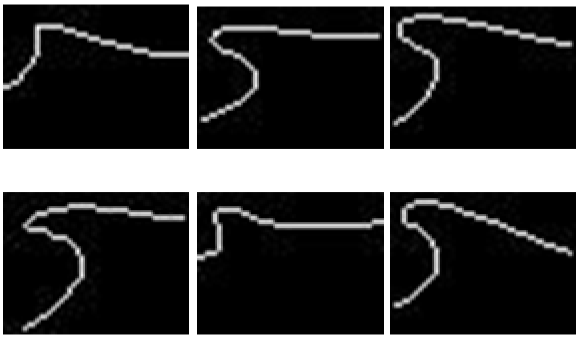


Fig. 5. Contour of dorsal fin. The contours of the dorsal fin are also called dealing samples for phase two were extracted features and the classification of the fin is made.

3 Feature Extraction

At this stage we focus on obtaining contour features that factor to differentiate between the types of dorsal fin shape. Three kinds of shape and take their flanks.

There are other features that could be obtained as the wet skin texture but the blue whale with sun reflection alters the appearance, taking other factors that should be addressed from the pre-processed before being used as a feature. Another feature to consider is into account is that the photographs are not taken from the same distance and location; for this reason the seven Hu moment invariants are a good choice in feature extraction providing be invariant to changes in scale, translation and rotation [7, 8].

They extracted features samples of the three types of shape of the dorsal fin and seven Hu invariant moments, it becomes part of the feature vector to deal in the standings. In Figure 6 the Hu invariant moments ten samples of each class is plotted and where their behavior is observed.

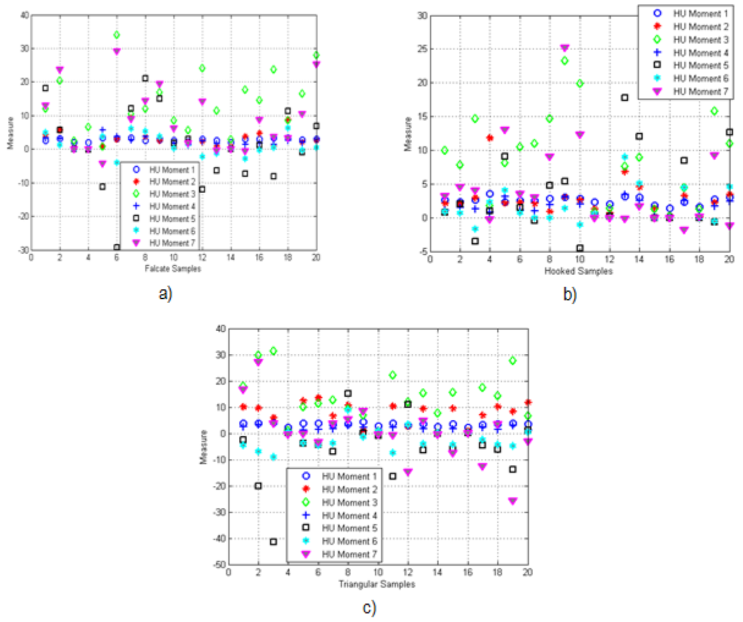


Fig. 6. Is the graph of twenty samples each of the shape of the dorsal flank without differentiating which were extracted features - HU seven moments invariant time are graphed. Where, (a) Falcate class; (b) Hooked class; and (c) Triangular class.

4 Classification

The study supervised by taking reference a previous classification by specialists and contemplated ordering your catalog using manual techniques classification is made. The minimum distance classifier and k-Nearest Neighbors (K-NN) will look to be quick in qualifying. Although training may take a little longer by marine mammal specialists (especially if the blue whale) to select samples where the shape of the dorsal fin is representative of each class, not a negative point consider.

4.1 Training

It will take at random a number of samples of each class. The classes defined are six right falcate, falcate left, right hook, left hook, right and left triangular.

4.2 Classifiers

For the minimum distance classifier to classify the samples are taken knowing how they belong and depending on the estimate is to determine whether the valuation is correct or incorrect [9].

In the KNN classifier takes K nearest neighbors of the sample to be classified. K is an odd integer number K = 3,5,7,9 Starting on number of K = 5, there may be a tie. Therefore, a tiebreaker rule is proposed. Take another neighbor and if they present another tie proceed with my order.

5 Experiments and Results

This section describes the experiments and their results will be discussed, taking the methodology set. The methods were tested in Matlab®.

For feature extraction are represented by the seven Hu moment invariants. Classes in the experiments are given as: a) Class 1-Falcate right b) Class 2 left Falcate c) Class 3 - right hook d) Class 4 - left hook e) Class 5 - Triangular right f) Class 6 - Triangular left. It requires an analysis of the income generated in the experiments; calculating deal sensitivity and accuracy. [10-12]

5.1 Minimum Distance Classifier

Conditions for training; 10 samples of each of the classes for a total of 60 training samples were taken and to classify 20 samples of each class were taken, the results of the classification are shown in Table 1. In Table 2, one can see the experiment performance analysis.

Table 1. Experiment with the minimum distance classifier.

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Estimation	0.2	0.35	0.2	0.6	0.45	0.45

Table 2. Evaluation of minimum distance classifier.

Test	Evaluation
Sensitivity (%)	37.5
Precision (%)	37.5

5.2 K-NN Classifier

For the experiment, 10 samples of each of the classes give 60 sample are taken for the training. In the group of 20 samples of each class to class, tests are performed with the value of K = 3,5,7,9.

Table 3. Experiments with the KNN classifier with six classes (distinguishing between right and left).

Class	Estimation k=3	Estimation k=5	Estimation k=7	Estimation k=9
Class 1	0.65	0.5	0.5	0.5
Class 2	0.35	0.35	0.3	0.35
Class 3	0.4	0.3	0.35	0.35
Class 4	0.5	0.55	0.55	0.55
Class 5	0.25	0.3	0.25	0.25
Class 6	0.4	0.45	0.3	0.45

Table 4. Evaluation of K-NN classifier with six classes.

	Evaluation k=3	Evaluation k=5	Evaluation k=7	Evaluation k=9
Sensitivity (%)	45.83	40.83	37.5	40.83
Precision (%)	45.83	40.83	37.5	40.83

The best result was given in which the value of K = 3 and Class 5 features in classifying low estimate.

Classes for this experiment are given by: a) Class 1 - Falcate, b) Class 2 - Hooked, c) Class 3 - Triangular. Classes include both flanks.

Table 5. Experiment with the KNN classifier regardless flanks of the dorsal fin.

Class	Estimation k=3	Estimation k=5	Estimation k=7	Estimation k=9
Class 1	0.8	0.8	0.85	0.75
Class 2	0.6	0.45	0.6	0.65
Class 3	0.7	0.65	0.7	0.55

Table 6. Evaluation of K-NN classifier with three classes.

	Evaluation k=3	Evaluation k=5	Evaluation k=7	Evaluation k=9
Sensitivity (%)	70	63.33	71.66	65
Precision (%)	70	63.33	71.66	65

By grouping classes, K = 7, presented better results when classifying. Furthermore Class 2 (hooked) presented lower rate when compared classified the other two forms.

6 Conclusions

The K-NN classifier performs best compared to the minimum distance classifier to classify the blue whale through the dorsal fin shape, besides being of low complexity. Pre-qualifying stage must ensure are as correct because the problems they had to alter the selection of samples for training and the classifier used.

The shape of the dorsal fin shown hooked to classify a lower estimate among other way and is by a factor that alters the shape of the dorsal fin of the samples and is barnacles that stick to the flap and the body, deforming its shape. Although the classifier gives an estimation of the shape of which this may give an incorrect estimate. This factor is suggested to perform a pre-processing for the detection and / or removal of external agents that alter the shape. Classification performs better when the number of classes is reduced.

Another factor that could observe that alters the shape of the dorsal fin of the blue whale is trademarks and / or mutilation that occurs throughout life, causing a misclassification. Detection and / or correction at the stage of pre-processing are suggested.

Acknowledgments. We express our gratitude to the National Polytechnic Institute of Mexico (Instituto Politécnico Nacional) and National Council on Science and Technology of Mexico (CONACYT) for the help and support for project number 221284. Thanks for the support from the team of Diane Gendron Laniel.

References

1. Auger-Méthé, M., Whitehead, H.: The use of natural markings in studies of long-finned pilot whales (globicephala melas). Marine Mammal Science, 23(1), pp. 77–93 (2007)
2. Perrin, W. F., Wursig, B., Thewissen, J. G. M. (Eds.): Encyclopedia of marine mammals. Academic Press (2009)
3. Gendron, D., & Ugalde de la Cruz, A.: A new classification method to simplify blue whale photo-identification technique. J. CETACEAN RES. MANAGE, 12(1), pp. 79–84 (2012)
4. Gonzalez, R. C., Woods, R. E.: Digital image processing (2002)
5. Shih, F. Y.: Image processing and pattern recognition: fundamentals and techniques. John Wiley & Sons (2010)
6. Lankton, S., Tannenbaum, A.: Localizing region-based active contours. Image Processing, IEEE Transactions on, 17(11), pp. 2029–2039 (2008)
7. Hu, M. K.: Visual pattern recognition by moment invariants. Information Theory, IRE Transactions on, 8(2), pp. 179–187 (1962)

8. Theodoridis, S., Koutroumbas, K.: Pattern recognition. IEEE TRANSACTIONS ON NEURAL NETWORKS, 19(2), 376 (2008)
9. Lin, H., Venetsanopoulos, A.N.: A weighted minimum distance classifier for pattern recognition. In proceedings of the Canadian Conference on Electrical and Computer Engineering, vol. 2, pp.904–907, (1993)
10. Trejo-Salazar, D., Carvajal-Gámez, B.E., Gallegos-Funes, F.J.: Algoritmo de segmentación para imágenes reales para la clasificación de cetáceos. In proceedings of COMIA (2013)
11. Fathi, A., Naghsh-Nilchi, A. R.: Automatic wavelet-based retinal blood vessels segmentation and vessel diameter estimation. Biomedical Signal Processing and Control, 8(1), pp. 71–80 (2013)
12. Fraz, M. M., Barman, S. A., Remagnino, P., Hoppe, A., Basit, A., Uyyanonvara, B., Owen, C. G.: An approach to localize the retinal blood vessels using bit planes and centerline detection. Computer methods and programs in biomedicine, 108(2), pp. 600–616 (2012)

ICA-Multiclass SVM as a Monitoring System of Complex Processes

Jesús Alejandro Navarro Acosta[1], Juan Pablo Nieto González[1], and Víctor Manuel Cortés Figueroa[2]

[1] Corporación Mexicana de Investigación en Materiales, S.A. de C.V. COMIMSA, Mexico

[2] Vertex, S.A. de C.V., Mexico

jesus.navarro@comimsa.com, juan.nieto@comimsa.com, vmcortesf@gmail.com

Abstract. Nowadays due to advancement of technology, the processes have become increasingly complex. So perform effectively monitor and correct diagnosis of these, has become a challenging task. this because the current processes have in common presence of combinations and correlation between analog and digital variables, presence of noise, etc. This issued is overcome by multivariate statistical methods which can efficiently represent the different correlations between variables. One of the most popular methods for process monitoring is the Hotelling's T^2 distance but a large amount of data makes the process monitoring a difficult task. Principal Component Analysis (PCA) is used for the data reduction which means the extraction of few numbers representing the most variance of the analyzed data. However the use of these two methods is limited to normal multivariate data set and in industry rarely normal data set is presented, hence different methods are used for this purpose. Independent component analysis (ICA) has been used in no-normal multivariate processes. In this paper a monitoring and fault diagnosis system for complex multivariate processes is presented. The new proposal is based on Multiclass Support Vector Machines (MCSVM) and the case study was addressed to the automotive industry. A comparison against a similar technique is carried out. Simulation of the new proposal shows promisory results.

Keywords: Support vector machines, independent component analysis, process monitoring.

1 Introduction

Since todays customer's satisfaction has become a priority, the manufacturing of high quality products is what every industry is looking for. In order to achieve this, different statistical control tools have been applied such as $\overline{X} - R$, $\overline{X} - S$. However, these tools are limited because their one-dimensional nature besides the normality requirement over the data. In several practical cases these methodologies are not suitable because of the correlation among the variables is not taken into consideration. The multivariate statistical process control provides certain advantages over the univariate models, since it considers the correlation between each pair of variables. Hotelling's T^2 distance is the most used tool in the multivariate statistical analysis, however large amounts of data makes the task difficult and time consuming. It is said that a big problem to tackle is the "information overload" [8] therefore a method that reduces the dimension of the data without losing a great deal of information is required. The Hotelling distance and PCA are combined for industrial process monitoring where the corresponding control limit is determined assuming that latent variables obey a Gaussian distribution. Unlike PCA, ICA has no restrictions of orthogonality which not only allows decorrelation of variables but also considers a higher order statistics to make independent latent variables. Therefore, ICA can be applied to a non-Gaussian process. On the other hand, support vector machines (SVM) based on statistical learning theory and principle of structural risk minimization, have been applied in the field of fault recognition for its excellent ability of generalization [10]. In this paper a system for monitoring and fault diagnosis for a multivariate process based on multiclass support vector machines (MCSVM) is presented, this methodology is applied in the automotive industry, the data were obtained by using the simulator VEHDYNA. The results of this system are compared with those obtained by implementing ICA as proposed in [6]. The organization of this paper is as follows: In Section 2 the techiques are explained. Section 3 presents the description of the systems to compare. In Section 4 the analysis and results are shown. Section 5 presents the conclusions.

2 Techniques

2.1 Independent Component Analysis

Independent Component Analysis (ICA) is a projection method wich finds underlying components from multivariate statistical data. The difference between ICA and other projection methods is that ICA can handle with non-Gaussian or independent statistic data. ICA was originally proposed to solve the blind source separation problem, which involves recovering independent source signals after they have been linearly mixed by unknown matrix A.

In the ICA algorithm , it is assumed that d measured variables $x_1, x_2, \ldots, x_d$ can be expressed as linear combinations of $m(\leq d)$ unknown independent components $s_1, s_2, \ldots, s_m$. The independent components and the measured variables have means of zero. The relationship between them is given by Equation 1.

$$X = AS + E \tag{1}$$

where $X = [x_1, x_2, \ldots, x_n] \in R^{d \times n}$ is the data matrix (ICA employs the tranposed data matrix), $A = [a_1, \ldots, a_m] \in R^{d \times m}$ is the unknown mixing matrix , $S = [s_1, s_2, \ldots, s_n] \in R^{m \times n}$ is the independent component matrix, $E \in R^{d \times n}$ is the residual matrix, and n is the number of samples. Here, we assume $d \geq m$

(when $d = m$, the residual matrix E becomes the zero matrix). The main problem of ICA is to estimate both the mixing matrix A and the independents S from only the observed data X. Now we can define the objetive of ICA as follows: to find a demixing matrix W whose form is such that the rows of the reconstructed matrix $\hat{S}$ becomes as independent of each other as possible (see Equation 2).

$$\hat{S} = WX \tag{2}$$

This formulation is not really different from the previous one, since after estimating A, its inverse gives W when $d = m$. It will assume $d = m$, unless otherwise specified. For mathematical convenience, we define that the independent components have unit variance. This makes the independent components unique, up to their signs. The initial step of ICA is whitening, also known as sphering, which eliminates all the cross-correlation between random variables. Consider a d-dimensional random vector x_k at sample k with covariance $R_x = (x_k x_k^T)$ where E respresent expectations. The eigen decomposition of R_x is given by Equation 3.

$$R_x = U \Lambda U^T \tag{3}$$

The whitening tranformation is expressed as:

$$z_k = Qx_k \tag{4}$$

where $Q = \Lambda^{-1/2} U^T$ [13]. One can easily verify that $R_z = E(z_k z_k^T)$ is the identity matrix under this transformation. After the transformation we have:

$$z_k = Qx_k = QAs_k = Bs_k \tag{5}$$

where B is an orthogonal matrix as verified by the following relation:

$$E\{z_k z_k^T\} = BE\{s_k s_k^T\}B^T = BB^T = I \tag{6}$$

Therefore, the problem is reduced to finding an arbitrary full-rank matrix A to the simpler problem of finding an orthogonal matrix B, since B has fewer parameters to estimate as a result of orthogonality contraint. Then, s_k from Equation 5 can be estimated as follows:

$$\hat{s_k} = B^T z_k = B^T Q x_k \tag{7}$$

The relation between W and B from Equations 2) and 7 can expressed as:

$$W = B^T Q \tag{8}$$

Hence Equation (2) can be rewritten as:

$$\hat{s} = WX = B^T Z = B^T QX = B^T \Lambda^{-1/2} U^T \tag{9}$$

To calculate B, a fast fixed-point algorithm for ICA (FastICA) is used. This algorithm calculates the column vector $b_i (i = 1, 2, \ldots, m)$ of B through iterative steps [7]. After obteining B, can calulate $\hat{s}$ by using equation (2) and W from equation (8). To divide W into two parts, dominant part(W_d) and excluded part (W_e). Lee in [9] propose three statistics for process monitoring as follows:

$$I^2 = \hat{S}_d^T \hat{S}_d \tag{10}$$

$$I_e^2 = \hat{S}_e^T \hat{S}_e \tag{11}$$

$$SPE = e^T e = (x - \hat{x})^T (x - \hat{x})$$

where $\hat{S}_d = W_d X, \hat{S}_e = W_e X$ and $\hat{x} = Q^{-1} B_d \hat{S} = Q^{-1} B_d W_d X$. In PCA monitoring, the latent variables are assumed to be Gaussian distributed, hence te upper control limit for T^2 can be determined by using Equation 4. In ICA, I^2, I_e^2 and SPE depart from the normality assumption. Lee et al. therefore proposed to use KDE to determine the control limits for I^2, I_e^2 and SPE statistics for ICA monitoring. Additionally, the variable contributions of $x(t)$ for $I^2(t)$ and $I_e^2(t)$ can be defined as:

$$X_{cd}(t) = \frac{Q^{-1} B_d \hat{S}_d(t)}{||Q^{-1} B_d \hat{S}_d(t)||} ||\hat{S}_d(t)|| \tag{12}$$

$$X_{ce}(t) = \frac{Q^{-1} B_e \hat{S}_e(t)}{||Q^{-1} B_e \hat{S}_e(t)||} ||\hat{S}_e(t)|| \tag{13}$$

After extracting essential features, it requires introducing a method for the classification problem. SVM is used in order to achieved this objetive.

2.2 Support Vector Machines

Support Vector Machines (SVM) is a machine learning algorithm used for classification and regression of high dimensionality data sets. This provides great results due to their ability to deal with problems such as local minimum, few data samples and nonlinear problems. Standard SVM are used for binary classification problems [3]. The central idea of this technique is to determine a linear separation (separating hyperplane) which is oriented in a way that its distance to the nearest data points in each class is maximized. The nearest data points are known as support vectors [1].

The Linearly Separable Case The input vectors $\mathbf{x}_i \in \mathbb{R}^d$ $(i = 1, 2, ..., n)$ correspond with labels $\mathbf{y}_i \in \{-1, +1\}$. There exists a separating hyperplane and its function is as:

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{14}$$

where $\mathbf{w} \in \mathbb{R}^n$ is a normal vector, the bias b is a scale and $\frac{|b|}{||\mathbf{w}||}$ represents the perpendicular distance from the separating hyperplane to the origin. While the

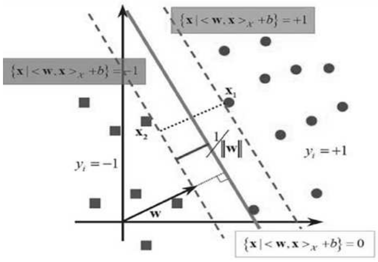

$\{\mathbf{x} \mid <\mathbf{w},\mathbf{x}>_{\mathcal{X}} +b\} = +1$
$\{\mathbf{x} \mid <\mathbf{w},\mathbf{x}>_{\mathcal{X}} +b\} = -1$
$\mathbf{x}_1$
$\mathbf{x}_2$
$1/\|\mathbf{w}\|$
$y_i = -1$
$y_i = +1$
$\mathbf{w}$
$\{\mathbf{x} \mid <\mathbf{w},\mathbf{x}>_{\mathcal{X}} +b\} = 0$

Fig. 1. Definition of geometric hyperplane and margin. Adapted from [2].

distance from hyperplane to the margin is given by $d = \frac{1}{\|\mathbf{w}\|}$. Figure 1 shows the separating hyperplane and the maximum margin between classes (dotted lines).

Two parallel hyperplanes can be represented as shown in Equation 15.

$$\mathbf{y}_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \geq 1 \tag{15}$$

As defined above SVM seeks to maximize the distance between two classes, where the amplitude of the margin between two parallel hyperplanes is $d = \frac{2}{\|\mathbf{w}\|}$, Hence for the linearly separable case can find the optimal hyperplane by solving the following quadratic optimization problem:

$$\begin{aligned} &min\ \frac{1}{2}\left\|\mathbf{w}\right\|^2 \\ &\text{subject to: } \mathbf{y}_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \geq 1 \end{aligned} \tag{16}$$

By introducing Lagrange multipliers $\alpha_i (i = 1, 2, ..., n)$ for the constraint, the primal problem (17) becomes a task of finding the saddle point of Lagrange. Thus, the dual problem becomes:

$$\begin{aligned} &max\ L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{n}^{ij} \alpha_i \alpha_j \mathbf{y}_i \mathbf{y}_j \\ &\text{subject to: } \sum_{i=1}^{n} \alpha_i \mathbf{y}_i = 0,\ \alpha_i \geq 0 \end{aligned} \tag{17}$$

By applying Karush–Kukn–Tucker (KKT) conditions, the following relationship holds:

$$\alpha_i \left[\mathbf{y}_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1\right] = 0 \tag{18}$$

If $\alpha_i > 0$, the corresponding data points are called support vectors (SVs). Hence, the optimal solution for the normal vector is given by:

$$\mathbf{w}^* = \sum_{i=1}^{N} \alpha_i \mathbf{y}_i \mathbf{x}_i \tag{19}$$

where N is the number of SVs. From Eq. (18), by choosing any SVs $(\mathbf{x}_k, \mathbf{y}_k)$, $b^* = \mathbf{y}_k - \mathbf{w}^* \cdot \mathbf{x}_k$. After $(\mathbf{w}^*, b^*)$ is determined. From the equation (19) the optimal separating hyperplane can be written as in [4,11]:

$$f(\mathbf{x}) = \mathbf{sign}\left(\sum_{i=1}^{N} \alpha_i \mathbf{y}_i (\mathbf{x} \cdot \mathbf{x}_i) + b^*\right)$$

$$\mathbf{x} \in \begin{cases} +1 \text{ si } f(\mathbf{x}) > 0 \\ -1 \text{ si } f(\mathbf{x}) < 0 \end{cases} \tag{20}$$

The Linearly Non-Separable Case SVM can also handle cases where the data are not linearly separable. These attempt to map the input vector $\mathbf{x}_i \in \mathbb{R}^d$ on a higher dimensional space. This process is based on the selection of a kernel function. Some of the most used kernel ($\mathbf{K}$) functions are [6]:

- Linear Kernel
- Polynomial Kernel
- Radial basis function
- Sigmoid kernel

Hence, the optimal hyperplane (eq. (21)) takes the form as.

$$f(\mathbf{x}) = \mathbf{sgn}\left(\sum_{i=1}^{N} \alpha_i \mathbf{y}_i \cdot \mathbf{K}(\mathbf{x} \cdot \mathbf{x}_i) + b^*\right) \tag{21}$$

Multiclass SVM As mentioned previously, SVM is able to classify into two classes only. Among the most applied methods for achieving multi classification with SVM are known as one against all (OAA) which builds M SVM where M is equal to the number of classes to be classified, after each of these SVM separates one class from the rest. The i th MSV is trained with all the training samples of the i th class with a different label to the other. Another method is known as one against one (OAO), which builds $M*(M-1)/2$ MSV. These combine their classification function to determine the class to which the test sample belongs, that predictions by accumulating round (votes), prediction with the most votes shall be the final classification [5,12].

2.3 Confusion Matrix

In the area of artificial intelligence, the confusion matrix is a visualization tool used in supervised learning. Each column of the matrix represents the number of predictions for each class, and each row represents each real class. The confusion matrix facilitates observe if the system is confusing two classes.

3 System Description

3.1 Independent Component Analysis for Monitoring Process

The feature extraction based on ICA is used to project the high dimension dataset into a lower one $m \leq d$. The extracted independent components are then used to calculate the systematic part statistic (in this paper will be used only (I^2) statistic), thus performs a process monitoring. Detailed explanation and steps of the algorithm can be reviewed at [9].

3.2 MCSVM for Fault Diagnosis

The monitoring system based on Support Vector Machines requires databases both normal operation and failure mode for learning. As shown in Figure 2, the system does not have any pre-processing. Once generated these databases serve as input for training and system validation. If system performance is suitable, this is ready for test data.

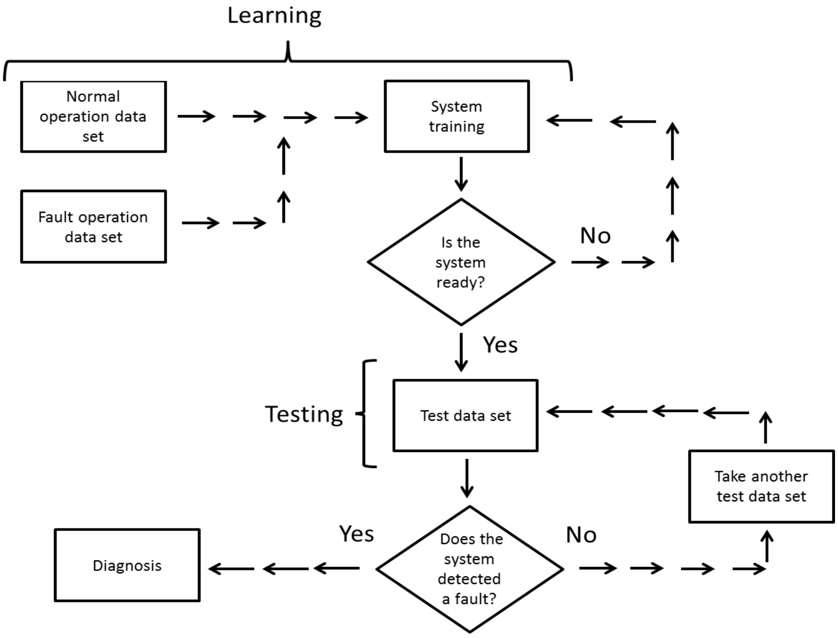


Fig. 2. Methodology based on MCSVM.

4 Analysis and Results

Five variables are monitored (A,B,C,D,E) these correspond to five sensors which monitor different parts of a vehicle. The simulations were carried out on the vehicle maneuver known as chicane. Chicane refers to a change in the straight path of the vehicle, for example, when the vehicle performs a overtake maneuver. The aim is detect and diagnose failures in the sensors. Databases both in normal operation and fault modes are generated: the first one consists of 4400 observations for the 5 variables and the second database is constructed from data for each of the five possible faults. 1600 observations for each variable into failure mode were taken, forming a 12400 × 5 size matrix. It is considered that a fault exists when in the sensor occur changes its face value outside a range of ±5%, the Fig 3 shows an example of a failure in the variable A, starting at observation 21 and ending at observation 32.

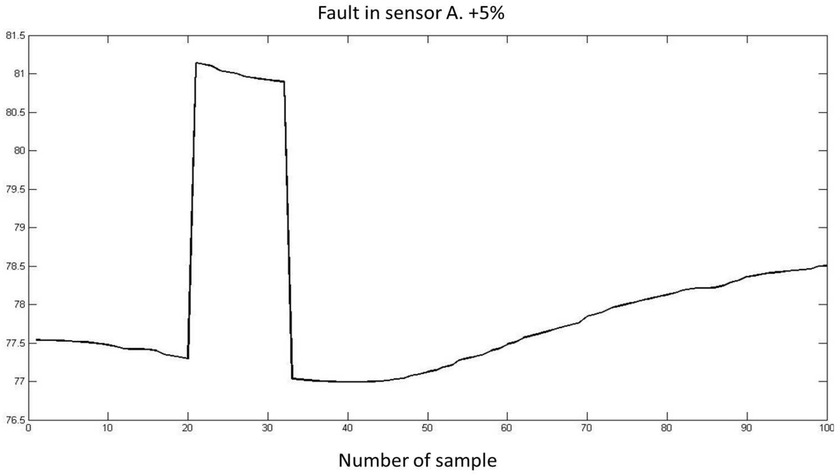


Fig. 3. Fault in sensor A.

4.1 ICA Performance

ICA is implemented to obtain the independent components and thereby can compute the statistics described in [9]. Then, the univariate kernel density estimator is used to estimate the density function of the normal statistics values and therefore, it was able to compute the control limit of normal operating data. At this stage the normal operation database is used, the control limit value is 4.6685. In the next stage the system is ready to monitor new samples. First data under control are monitored. Figure 4 shows state of the process after analyze

100 samples. It can be seen that the control chart show the process under the control limit computed by ICA algorithm.

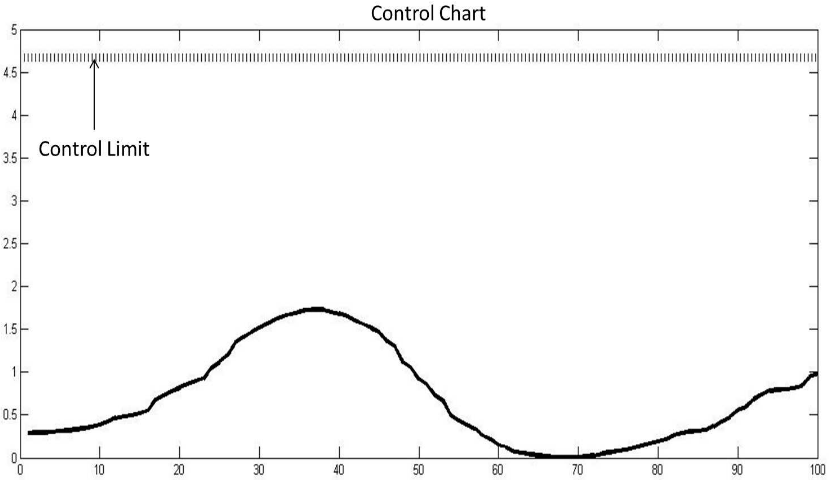


Fig. 4. Control chart for normal operation sample.

Now for the following step, the fault mode database is used to test the ICA algorithm's performance. Figure 5 shows state of the process after analyze the described failure above on the sensor A. It can see that the control chart show the process over control limit calculated by the ICA algorithm.

4.2 MCSVM Performance

A matrix size of 12400×5 and a vector of $size = 12400 \times 1$ are used for training and validation. Such vector represents six different labels for the six possible states of the system (0,1,...,5 for normal, fault in sensor A,...,E). The approximate computation time for training was 6.3 seconds, while the performance of the classifier was 99.95%. The model for the classification used a radial kernel and found a total of 1612 support vectors. Table 1 shows the confusion matrix.

Figure 6 shows the analysis of failure in the variable A of the MCSVM based system. We can observe that system has detected, located and identified the fault at 100%.

The results obtained by performing 200 simulations of single failures (one variable at a time) in different locations and at different magnitudes of deviation from the nominal values of the monitored variables are presented. Table 2 shows the performance of MCSVM system for fault diagnosis. Table 3 shows the performance of ICA system for fault diagnosis (only use I^2 statistic). Table 4 presents the comparison of qualitative features of both methods.

Table 1. Confusion matrix.

	0	1	2	3	4	5
0	899	0	0	0	0	0
1	0	313	0	0	0	0
2	0		324	0	0	0
3	0	0	0	331	0	0
4	1	0	0	0	322	0
5	0	0	0	0	0	290

Table 2. Performance of MCSVM system for fault diagnosis.

Deviation	Faulty Samples	Detection	Identification	Location
5%	15	100%	100%	100%
5%	10	100%	99.6%%	100%
5%	5	100%	100%	100%
3%	15	100%	99.4%	100%
3%	10	100%	100%	100%
3%	5	100%	100%	100%

Table 3. Performance of ICA system for fault diagnosis.

Deviation	Faulty Samples	Detection	Identification	Location
5%	15	100%	100%	100%
5%	10	100%	100%%	100%
5%	5	100%	100%	100%
3%	15	0%	0%	0%
3%	10	0%	0%	0%
3%	5	0%	0%	0%

Table 4. Features of both methods.

MSVMC system	ICA process monitoring
Need databases in normal operation and failure mode	Only needs database in normal operation
Requires training time	Only requires computation time
Able to operate online	Able to operate online
By nature of the algorithm is able to detect failure variable	Requires to build the contribution charts
Capable of handling noise	Requires the calculation of other statistics to sensitize

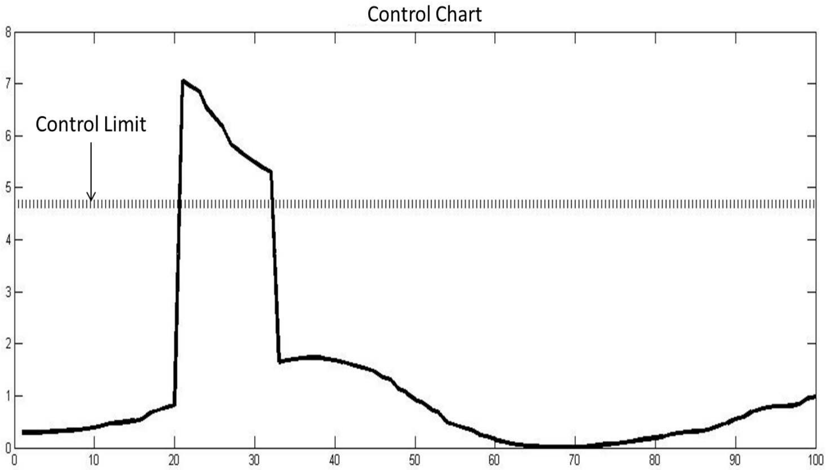
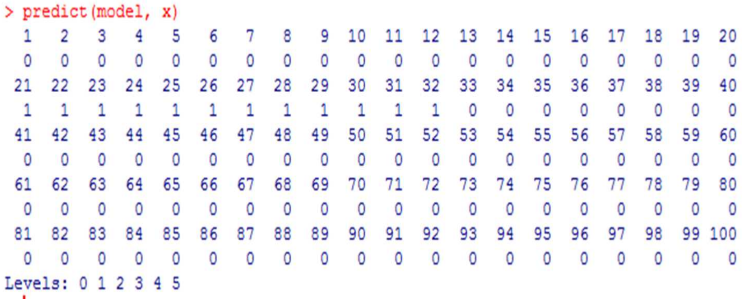


Fig. 5. Control chart for fault operation sample.



```
> predict(model, x)
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
  1   1   1   1   1   1   1   1   1   1   1   1   0   0   0   0   0   0   0   0
 41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
Levels: 0 1 2 3 4 5
```

Fig. 6. MCSVM analysis of failure in the variable A.

5 Conclusions

In this paper a system for detection and fault diagnosis based on support vector machines has been presented. The basic idea of this system is to use the multiclass approach of support vector machines algorithm and to implement it on complex systems such as an automotive process. The results showed that the proposed system has promising results in the field of fault detection, identification and location. The performance of the proposal is improved due to the high efficiency of classification of support vector machines and their ability to work with large data. In addition, the ICA approach as process monitoring proposed in [6] was implemented, obtaining high performance on process monitoring. Because ICA can handle with non-Gaussian data, it is ideal for monitoring automotive processes. Both methods provide excellent results, but the simplicity in the design of the system based on MCSVM and the results presented in this paper, concludes that the Support Vector Machines are a robust technique for classification tasks and has excellent results in the field of fault detection.

References

1. Andre, A.B., Beltrame, E., Wainer, J.: A combination of support vector machine and k-nearest neighbors for machine fault detection. Applied Artificial Intelligence: An International Journal Vol. 27, 36–49 (2013)
2. Daz, E.E.G.: Boosting support vector machines. Master's thesis, University of the Andes School of Engineering, Bogot Colombia (2010)
3. Elhariri, E., El-Bendary, N., Fouad, M.M.M., Platos, J., Hassanien, A.E.: Multiclass svm based classification approach for tomato ripeness. Innovations in Bio-inspired Computing and Applications pp. 175–186 (2014)
4. Gong Yanjie, Gao Xuejin, W.P.y.Q.Y.: Online modeling method based on dynamic time warping and least squares support vector machine for fermentation process. 8th World Congress on Intelligent Control and Automation. July 6-9, Jinan, China pp. 481–485 (2010)
5. Hassen Keskes, A.B., Lachiri, Z.: Broken rotor bar diagnosis in induction machines through stationary wavelet packet transform and multiclass wavelet svm. Electric Power Systems Research, Elsevier pp. 151–157 (2013)
6. Hsu Chun-Chin, C.M.C., Long-Sheng, C.: Intelligent icasvm fault detector for non-gaussian multivariate process monitoring. Elsevier 37, 3264–3273 (2010)
7. Hyvärinen, A., Köster, U.: Fastisa: A fast fixed-point algorithm for independent subspace analysis. European Symposium on Artificial Neural Networks, Bruges, Belgium, (2006)
8. Hyvrinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Wiley-Interscience publication (2001)
9. Jong-Min Lee, Chang, K.Y., Lee, I.B.: Statistical process monitoring with independent component analysis. Journal of Process Control Vol. 14, 467–485 (2004)
10. Liu, B., Hou, D., Huang, P., Liu, B., Tang, H., Zhang, W., Chen, P., Zhang, G.: An improved pso-svm model for online recognition defects in eddy current testing. Nondestructive Testing and Evaluation Vol. 28, No. 4, 367–385 (2013)
11. Nicholas I. Sapankevich, R.S.: Time series prediction using support vector machines. IEEE computational intelligence magazine pp. 24–37 (2009)
12. Oscar Castillo, L.X.y.S.L.A.: Trends in intelligent systems and computer engineering. Springer Science + Business Media. New York, USA (2008)
13. Roshan Joy Martis, U.R.A.: Ecg beat classification using pca, lda, ica and discrete wavelet transform. Biomedical Signal Processing and Control, Elsevier Vol. 8, 437–448 (2013)

Peirce: an Algorithm for Abductive Reasoning Operating with a Quaternary Reasoning Framework

Felipe Rodrigues, Carlos Eduardo A. Oliveira, and Osvaldo Luiz de Oliveira

FACCAMP, Campo Limpo Paulista (SP),
Brazil

rodrigues_felipe7@hotmail.com, carlos.br@gmail.com, osvaldo@faccamp.br

Abstract. Abductive reasoning algorithms formulate possible hypotheses to explain observed facts using a theory as the basis. These algorithms have been applied to various domains such as diagnosis, planning and interpretation. In general, algorithms for abductive reasoning based on logic present the following disadvantages: (1) they do not allow the explicit declaration of conditions that may affect the reasoning, such as intention, context and belief; (2) they allow little or no consideration for criteria required to select good hypotheses. Using Propositional Logic as its foundation, this study proposes the algorithm Peirce, which operates with a framework that allows one to explicitly include conditions to conduct abductive reasoning and uses a criterion to select good hypotheses that employs metrics to define the explanatory power and complexity of the hypotheses. Experimental results suggest that abductive reasoning performed by humans has the tendency to coincide with the solutions computed by the algorithm Peirce.

Keywords: Abductive reasoning, automated Reasoning, logic, human factors.

1 Introduction

Abductive reasoning formulates hypotheses to explain observed facts using a theory as the basis. Numerous intellectual tasks make use of abductive reasoning, including medical diagnostics, fault diagnostics, scientific discovery, legal argumentation and interpretation.

Abductive reasoning algorithms based on logic frequently operate with the Theory, Hypotheses and Facts (THF) ternary reasoning framework (as shown in [2], [4], [5] and [11]). When these algorithms are formalized using Propositional Logic (PL) [9], the THF framework is frequently instantiated according to Definition 1.

Definition 1 (THF reasoning framework). The THF reasoning framework for abductive reasoning is a system ⟨T, H, F⟩ that consists of the following:

- A finite and non-empty theory set, T = { t_1, t_2, t_3, ..., t_m }, of PL sentences denoting $t_1 \wedge t_2 \wedge t_3 \wedge ... \wedge t_\mathrm{m}$. This set represents the hypotheses that must be assumed as truth during the reasoning process.

- A finite hypotheses set, H = { h_1, h_2, h_3, ..., h_n }, of PL sentences denoting $h_1 \lor h_2 \lor h_3 \lor ... \lor h_n$. This set represents the hypotheses that along with the set T explain the facts represented by the set F.
- A set with a single fact, F = { *f* }, where *f* is a PL literal (atom or negated atom). This set represents an occurrence of an evidence, a manifestation, a symptom, an observation, a mark or a sign to be explained through abductive reasoning.

Having the T and F sets as input, an abductive reasoning algorithm should find a set of hypotheses H that satisfies the following conditions:

$$\mathrm{T} \nvDash \mathrm{F}, \tag{1}$$

$$\mathrm{T} \cup \{ h \} \vDash \mathrm{F}, \forall h \in \mathrm{H}, \tag{2}$$

$$\mathrm{T} \cup \{ h \} \nvDash \bot, \forall h \in \mathrm{H}, \tag{3}$$

$$\{ h \} \nvDash \mathrm{F}, \forall h \in \mathrm{H}. \tag{4}$$

The statements above refer to the concept of logical consequence described in Definition 2.

Definition 2 (A ⊨ B, i.e., B is a logical consequence of A). Let A = { a_1, a_2, a_3, ..., a_n } and B = { b_1, b_2, b_3, ..., b_m } be two finite and non-empty sets of PL sentences. Then, A ⊨ B if and only if the interpretations in which $a_1 \land a_2 \land a_3 \land ... \land a_n$ is true, $b_1 \land b_2 \land b_3 \land ... \land b_m$ is also true.

The condition (1) prevents that the theory set T alone has as logical consequence the facts set F. Hypotheses satisfying condition (2) are called **candidate hypotheses**, and they can explain the single fact denoted by F. Candidate hypotheses satisfying condition (3) are called **consistent hypotheses**. Conversely, candidate hypotheses that do not satisfy condition (3) are called inconsistent hypotheses and should be discarded. Candidate hypotheses that satisfy condition (4) are called **explanatory hypotheses**. Conversely, candidate hypotheses that do not satisfy condition (4) are called non-explanatory hypotheses and should be discarded.

Example 1. Joseph has a large lawn in front of his house. One day, Joseph arrives at home and observes that the lawn is wet. Considering only that (1) rain can make the lawn wet and that (2) sprinklers installed across the lawn can make it wet, which hypotheses can explain the fact that the lawn is wet?

One possible formalization using the THF framework consists in defining:

- Propositions '*r*: **Rain** occurred', '*s*: **Sprinklers** were activated' and '*w*: Lawn is **wet**'.
- A theory set T = { $r \rightarrow w$, $s \rightarrow w$ }.
- A fact set F = { *w* }.

The theory set T and the fact set F satisfy condition (1), whereas a theory T_1 = { $r \rightarrow w$, $s \rightarrow w$, *w* } has F as its logical consequence; therefore T_1 and F do not satisfy condition (1). Let H = { *r*, *s*, $r \land s$, $r \land \neg w$, *w* } be a set of candidate hypotheses. Each hypothesis $h \in$ H satisfies condition (2), and each hypothesis $h \in$ { *r*, *s*, $r \land s$, *w* } satisfies condition (3), i.e., they are consistent. However, the hypothesis $r \land \neg w$ is inconsistent because T ∪ { $r \land \neg w$ } ⊨ ⊥; therefore, it must be discarded. Each hypothesis $h \in \{ r, s, r \wedge s \}$ satisfies condition (4); however, the hypothesis w is not explanatory because $\{ w \} \vDash$ F; therefore, it must also be discarded. Thus, removing the inconsistent and non-explanatory hypotheses from H, we obtain H = $\{ r, s, r \wedge s \}$.

In general, abductive algorithms work as follows: having the theory set T and the fact set F as the input, the algorithm verifies whether or not the condition (1) has been satisfied; if the condition (1) is not satisfied, then there are no hypotheses to be formulated because F is a logical consequence of T; however, if the condition (1) is satisfied, then the algorithm formulates a finite set of possible hypotheses H that satisfies the condition (2). Next, the algorithm removes from H the hypotheses that do not satisfy the conditions (3) and (4), and thus returning the resulting set H as an answer by the algorithm.

Some algorithms, however, include an additional step with the goal of letting in H only hypotheses considered good, according to extra-logical criteria. A criterion commonly used is “simplicity”, which considers, for example, an atomic hypothesis better than a composite hypothesis, e.g., r is better than $r \wedge s$.

The need to represent conditions such as context, circumstance and intention is common and important when conducting abductive reasoning. For example, reasoning to make a medical diagnosis considering the context of diseases of a region. Operating with a THF reasoning framework, the existing algorithms to perform abductive reasoning have the disadvantage of forcing the representation of these conditions in the theory set T. This solution is not appropriate because representing conditions in the theory set T mischaracterizes the theory, making it less general and more *ad hoc* (specific to explain what one wants to explain).

Abductive reasoning formulates hypotheses, and some of these hypotheses may be better at explaining the facts than others. Today we do not know, exactly, which criteria determine what makes a hypothesis better than another, authors from several fields [3] [8] [10] [13] [16] [17] have suggested that abductive reasoning involves the selection of good hypotheses. However, the existing abductive reasoning algorithms have the disadvantage of dedicating little or no consideration for criteria required to select good hypotheses.

Many practical applications of reasoning require the definition of a set of $n \geq 2$ facts. However, the many existing algorithms have the disadvantage of operating with only a single fact.

This work proposes an algorithm, called Peirce, that performs abductive reasoning, and this algorithm differs from the existing solutions mainly because (1) it works with a reasoning framework called TCHF (Theory, accepted Conditions, Hypotheses and Facts), thus allowing conditions to be explicitly represented; (2) it allows $n \geq 2$ facts to be represented; and (3) it introduces a criterion to select good hypotheses that employ metrics to define the explanatory power and the complexity of the hypotheses.

Section 2 describes the algorithm Peirce, dedicating particular attention to the design and operation of the TCHF reasoning framework (Subsection 2.1) and the definition of a criterion to select good abductive hypotheses (Subsection 2.2). The pseudocode for the algorithm Peirce is presented and discussed in Subsection 2.3. Section 3 details an experimental study conducted to verify whether the solutions computed by the algorithm Peirce tend to coincide with the abductive reasoning performed by humans. Section 4 describes related works, highlighting the differences with this work. Section 5 presents the conclusions.

2 The Algorithm Peirce

The abductive reasoning algorithm proposed in this study has been named Peirce in honor of the American philosopher Charles Sanders Peirce, who created the concept of abductive reasoning [14]. The following subsections detail the reasoning framework used by the algorithm Peirce, a criteria to select good hypotheses and the pseudocode of the algorithm.

2.1 TCHF Reasoning Framework

The TCHF reasoning framework proposed in this study differs from the classic THF reasoning framework (Definition 1) by including the accepted conditions set C and by redefining the facts set F to allow the declaration of not just one single fact, but rather a finite number of one or more facts. The TCHF framework is formalized in Definition 3 and uses the PL sentences in HF form as specified in Definition 4.

Definition 3 (TCHF reasoning framework). The TCHF framework for abductive reasoning is a system $\langle$T, C, H, F$\rangle$ consisting of the following:

- A finite and non-empty theory set, T = { t_1, t_2, t_3, ..., t_m }, of PL sentences in HF form denoting $t_1 \land t_2 \land t_3 \land ... \land t_m$. This set represents the hypotheses that must be assumed as truth during the reasoning process.
- A finite hypotheses set, H = { h_1, h_2, h_3, ..., h_n }, of PL sentences in HF form denoting $h_1 \lor h_2 \lor h_3 \lor ... \lor h_n$. This set represents the hypotheses that along with the sets T and C explain the facts represented by the set F.
- A finite accepted conditions set, C = { c_1, c_2, c_3, ..., c_p }, of PL sentences in HF form denoting $c_1 \land c_2 \land c_3 \land ... \land c_p$. This set represents the conditions that must be assumed as truth during the reasoning process.
- A finite and non-empty facts set F = { f_1, f_2, f_3, ..., f_q } of PL positive literals, denoting $f_1 \land f_2 \land f_3 \land ... \land f_q$. The role of this set is to represent evidences, manifestations, symptoms, observations, marks or signs to be explained by the abductive reasoning.

Definition 4 (HF form). A sentence of PL in the HF form is an acyclic sentence written in one of the following formats:

- $a_1 \land a_2 \land a_3 \land ... \land a_n$, where a_i ($1 \leq i \leq n$) are literals.
- $a_1 \lor a_2 \lor a_3 \lor ... \lor a_n$, where a_i ($1 \leq i \leq n$) are negative literals.
- $a_1 \land a_2 \land a_3 \land ... \land a_n \rightarrow b_1 \land b_2 \land b_3 \land ... \land b_m$, where a_i ($1 \leq i \leq n$) are positive literals and b_j ($1 \leq j \leq m$) are literals.
- $a_1 \lor a_2 \lor a_3 \lor ... \lor a_n \rightarrow b_1 \lor b_2 \lor b_3 \lor ... \lor b_m$, where a_i ($1 \leq i \leq n$) are literals and b_j ($1 \leq j \leq m$) are negative literals.

The restriction of TCHF framework to sentences in the HF form aims to make the algorithm Peirce run the conversion of sentences in polynomial time, because sentences in HF form can be easily converted into Horn Clauses [9] (disjunction of literals with at most one positive literal). As will be described in Subsection 2.3, algorithm Peirce uses Resolution as the inference mechanism and this mechanism can be efficiently implemented on Conjunctive Normal Form (CNF) sentences with Horn Clauses.

The set C gives the TCHF reasoning framework the advantage of allowing the explicit definition of conditions that in the classical THF framework, would normally be declared within the theory set T. Thus, the set C avoids "contaminating" the set T with sentences that fundamentally do not belong to the theory. Moreover, this makes it easier to represent two or more instances of abductive reasoning that share the declarations of T and F but differ in the set of accepted conditions. Example 2, which is described next, illustrates the use of the TCHF reasoning framework.

Example 2. Consider once more the scenario described in Example 1 in which Joseph arrives at home and observes his lawn wet. However, let us say that Joseph knows that the water tank supplying the sprinklers has been empty for a month; therefore, under this condition, the sprinklers could not have been activated.

One possible formalization using the TCHF framework is to define the following:

- Propositions '*r*: **Rain** occurred', '*s*: **Sprinklers** were activated', '*w*: Lawn is **wet**' and '*t*: Water **tank** that supplies sprinklers is empty'.
- A theory T = { $r \rightarrow w, s \rightarrow w$ }, which is the same as in Example 1.
- A set of accepted conditions C = { t, $t \rightarrow \neg s$ }.
- A set of facts F = { w }, which is the same as in Example 1.

The conditions in abductive reasoning are motivated by several factors which are linked to context (information associated to space), circumstances (information associated with time), intention (manifestation of the will to reach some wanted conclusions), belief or faith (information that is accepted on principle) etc. Examples of specific conditions used in abductive reasoning are as follows: (1) In abductive reasoning used for medical diagnoses, regional context may allow one to specify a set of diseases that are common or uncommon for a given region; (2) In abductive reasoning used for anthropological studies, the specification of possible agents that might have been responsible for the death of a hominid based on the knowledge that the hominid lived 4 million years ago (circumstance); (3) In abductive reasoning for judicial decisions, possible conditions may be specified with the intent of acquitting (or condemning) a defendant; and (4) In abductive reasoning for religious or metaphysical beliefs, the faith or belief that there is life after death can be declared as a condition upon which reasoning are made.

Taking the sets T and F as inputs, an abductive reasoning algorithm operating with the TCHF framework should find a set of hypotheses H that satisfies the following conditions:

$$\mathrm{T} \cup \mathrm{C} \nvDash \mathrm{F}, \tag{5}$$

$$\mathrm{T} \cup \mathrm{C} \cup \{\, h \,\} \vDash_{\mathrm{p}} \mathrm{F},\ \forall\, h \in \mathrm{H}, \tag{6}$$

$$\mathrm{T} \cup \mathrm{C} \cup \{ h \} \nvDash \bot, \forall h \in \mathrm{H}, \tag{7}$$

The condition (6) uses the partial logical consequence as defined in Definition 5.

Definition 5 (A $\vDash_p$ B, i.e., B is a partial logical consequence of A). Let A = { a_1, a_2, a_3, ..., a_n }, B = { b_1, b_2, b_3, ..., b_m } and C = { c_1, c_2, c_3, ..., c_q }, C ⊆ B, be three finite and non-empty sets of PL sentences. Then, A $\vDash_p$ B if only if the interpretations in which $a_1 \wedge a_2 \wedge a_3 \wedge ... \wedge a_n$ is true, $c_1 \wedge c_2 \wedge c_3 \wedge ... \wedge c_q$ is also true.

2.2 Selection of Good Abductive Hypotheses

In general, several hypotheses may be able to explain observed facts. However, certain hypotheses may explain facts better than others. Therefore, abductive reasoning can be observed as a process that formulates $m \geq 1$ general hypotheses followed by the selection of $n \leq m$ good hypotheses. Naturally, selection criteria must be established, but it is still difficult to define the conditions that make a hypothesis good.

Contemporary philosophers have analyzed the issue of selecting good hypotheses. Harman [8] considers abduction to be an inference of the best explanation and argues that the best hypothesis is the simplest, most plausible and is the least *ad hoc*. By comparing theories (e.g., Darwin’s Theory of Evolution vs. Creationist Theory or Lavoisier’s Theory of Combustion vs. Phlogiston Theory), Thagard [16, 17] establish criteria that explain the preference for one hypothesis over another and considers the best hypothesis to be the most consilient (explains more facts), the most simple, and it would provide the best analogy with hypotheses that explain facts in other domains.

Criteria to select good hypotheses have been extensively studied in the fields of philosophy (e.g., [2] [8] [16]), psychology (e.g., [13]) and artificial intelligence (e.g., [3] [10] [15]). However, the precise formulation of these criteria remains controversial. In general, factors such as the “explanatory power” and the “complexity” of a hypothesis are recurrent and have similar connotations across several studies. Therefore, this study has proposed using these two factors to develop a selection criterion. Aiming at the development of algorithms to perform abduction that need dealing with quantitative measures for the explanatory power and the complexity of a hypothesis, this study proposes an understanding of these factors as follows:

- Explanatory power (or comprehensiveness): the explanatory power of a hypothesis quantifies the degree to which it is capable of explaining the facts involved in the reasoning. A metric for a hypothesis’ explanatory power is given by the ratio between the number of facts it can explain and the total number of facts to be explained by the abductive reasoning process. For example, a hypothesis that explains 4 out of 5 facts has an explanatory power of 4/5, and a hypothesis that explains all of the facts has an explanatory power of 1.
- Complexity: the complexity factor refers to how many different elements and relationships are present in a hypothesis. A metric for hypothesis complexity is the number of atomic propositions that it contains. For example, hypothesis r has a complexity of 1, and hypothesis $r \wedge s \wedge w$ has a complexity of 3.

Based on the metrics for explanatory power and complexity, this study proposes a criterion to select good hypotheses, which is declared in Definition 6.

Definition 6 (A criterion to select good hypotheses). Given a set H of candidate hypotheses to explain a set F of facts, $h \in$ H is considered a good hypothesis if it satisfies all the following conditions:

- The explanatory power of h is equal to or greater than a constant λ_1. The constant $\lambda_1 = 0.5$ has been used in the experiments described in this article.
- The complexity of h is equal to or less than a constant λ_2. The constant[1] $\lambda_2 = 5$ has been used in the experiments described in this article.
- The hypothesis h has the minimum complexity among all of the hypotheses that have the maximum explanatory power in H.

Examples 3 and 4 illustrate the application of Definition 6.

Example 3. Diseases manifest themselves through symptoms. Consider the following:

- Propositions 'c: Disease is **cold**', 'p: Disease is **pneumonia**', 'r: Disease is **rhinitis**', 'f: Symptom is **fever**', 'h: Symptom is **headache**', and 'z: Symptom is **coryza**';
- Theory T = { $p \rightarrow f \wedge z \wedge h$, $c \rightarrow f \wedge z$, $r \rightarrow h \wedge z$ }, the empty set C = { } of accepted conditions and observed facts set F = { f, z, h } (symptoms);
- A set of candidate hypotheses H = { p, c, r, $p \wedge c$, $p \wedge r$, $c \wedge r$, $p \wedge c \wedge r$ }.

Table 1 describes the explained facts, explanatory power and complexity of each candidate hypothesis $h \in$ H.

Table 1. Explained facts, explanatory power and complexity of candidate hypothesis of the Example 3. The '√' signals an explained fact.

Hypothesis	Explained facts			Explanatory power	Complexity
	f	***z***	***h***		
p	√	√	√	1	1
c	√	√		0.66	1
r		√	√	0.66	1
$p \wedge c$	√	√	√	1	2
$p \wedge r$	√	√	√	1	2
$c \wedge r$	√	√	√	1	2
$p \wedge c \wedge r$	√	√	√	1	3

All of the hypotheses have an explanatory power equal to or greater than $\lambda_1 = 0.5$ and complexity equal to or less than $\lambda_2 = 5$. The hypotheses p, $p \wedge c$, $p \wedge r$, $c \wedge r$, $p \wedge c \wedge r$ have explanatory power equal to 1, which is the maximum among all candidate

[1] The λ_1 and λ_2 values were chosen to coincide with human factors. Considering Miller experiment [12], human memory and human processing capacity is limited to 7 ± 2 simultaneous elements, hence $\lambda_2 = 5$. Good hypotheses explain at least 50% of the facts, hence $\lambda_1 = 0.5$.

hypotheses. Among these hypotheses with maximum explanatory power, hypothesis *p* has the complexity equals to 1, which is the minimum among the hypotheses. Therefore, *p* is a good hypothesis according to Definition 6.

Example 4. Another example involving diseases and symptoms. Consider the following:

- Propositions '*d*: Disease is **dengue**', '*u*: Disease is **flu**', '*b*: Symptom is **breathlessness**', '*f:* Symptom is **fever**', '*h*: Symptom is **headache**', '*m*: Symptom is **muscle pain**', '*r*: Symptom is **red spots**' and '*s*: Symptom is **sneezing**';
- Theory T = { $u \rightarrow f \wedge\ h \wedge m \wedge \mathrm{s}$, $d \rightarrow f \wedge h \wedge m \wedge \mathrm{r}$ }, the empty set C = { } of accepted conditions and observed facts set F = { *f, h, m, b* } (symptoms);
- A set H = { *u, d,* $u \wedge d$ } of candidate hypotheses.

Table 2 describes the explained facts, explanatory power and complexity of each candidate hypothesis $h \in$ H.

Table 2. Explained facts, explanatory power and complexity of candidate hypothesis of the Example 4. The '√' signals an explained fact.

Hypothesis	**Explained facts**				**Explanatory power**	**Complexity**
	f	***h***	***m***	***b***		
U	√	√	√		0.75	1
D	√	√	√		0.75	1
$u \wedge d$	√	√	√		0.75	2

All of the hypotheses have an explanatory power equals to 0.75 (i.e., explanatory power equal to or greater than $\lambda_1 = 0.5$) and complexity equal to or less than $\lambda_2 = 5$. Hypotheses *u* and *d* have a complexity of 1, which is the minimum across all of the candidates. Therefore, *u* and *d* are good hypotheses according to Definition 6.

2.3 Pseudocode for the Algorithm Peirce

Figure 1 presents the pseudocode for the algorithm Peirce. The algorithm Peirce formulates hypotheses that comply with equations (5), (6), (7) and the criterion to select good hypotheses of the Definition 6. Synthetically, the algorithm Peirce formulates candidate hypotheses and stores them in set H (line 11). Next, the algorithm removes inconsistent hypotheses from H (line 12) and then selects and leaves only the good hypotheses in H (line 13). The details of the algorithm are described below.

The algorithm uses the Resolution rule of inference for PL sentences in CNF expressed with Horn Clauses [9]. Candidate hypotheses are hypotheses *h* that satisfy equation (6). To compute these hypotheses, the algorithm translates the set of sentences T ∪ C ∪ ¬F to CNF expressed with Horn Clauses (line 4) and applies the mechanism of resolution (line 5). The result of the resolution is stored in the data structure R (set of clauses). If R contains at least one empty clause, then T ∪ C ⊨ F and no hypotheses are formulated (lines 6 and 7). If R does not contain an empty clause, then $T \cup C \not\models F$, equation (5) is met and candidate hypotheses can be formulated.

At line 11 each clause in R presents the possibility of formulating a hypothesis. Because R is in CNF, negating each clause results in a candidate hypothesis. The algorithm `Formulate_Candidate_Hypotheses` (line 11), operates as follows: (1) the algorithm negates each of the $m \geq 1$ clauses in R to obtain m first-candidates hypotheses h_1, h_2, h_3, ..., h_m, (2) combines these m first-candidates hypotheses in pairs to obtain conjunctive hypotheses of the type $h_i \wedge h_j$ $(i \neq j)$; and (3) combines the m first-candidates hypotheses three by three to obtain conjunctive hypotheses of the type $h_i \wedge h_j \wedge h_k$ $(i \neq j \neq k)$; … and then combines the m first-candidates hypotheses q by q to obtain conjunctive hypotheses of the type $h_i \wedge h_j \wedge \ldots \wedge h_q$ $(i \neq j \neq \ldots \neq q)$, where q = min (m, λ_2) and is λ_2 the constant that defines the maximum complexity of the hypotheses (λ_2 is defined in Definition 6).

```
    Algorithm Peirce(T, C, F)
    Input
    Theory set T, accepted condition set C and facts set F
    (specification is given in Definition 3).
    Output
    Hypotheses set H (specification is given in Definition 3).
1   {
2     if Consistent(T, C) then
3     {
4       R := Conjunctive_Normal_Form_Horn_Clauses(T, C, ¬F);
5       R := Resolution(R);
6       if R contains an empty clause then
7         write ("No hypotheses to formulate: T ∪ C ⊨ F");
9       else
10      {
11        H := Formulate_Candidate_Hypotheses(R);
12        H := Remove_Inconsistent_Hypotheses(T, C, H);
13        H := Select_Good_Hypotheses(T, C, H, F);
14      }
15    }
16    else
17      write("Unable to formulate hypotheses: T ∪ C ⊨⊥.");
18  }
```

Fig. 1. Algorithm Peirce.

At line 12, the algorithm `Remove_Inconsistent_Hypotheses` receives a set H of candidate hypotheses and removes from H hypotheses that do not satisfy $T \cup C \cup \{ h \} \not\models \perp$ (conformity to equation (7)). The algorithm works as follows: For each $h \in$ H: (1) the algorithm translates the sentences in the set $T \cup C \cup \{ h \}$ to CNF expressed with Horn Clauses, (2) applies the Resolution mechanism to this system of sentences and (3) removes hypothesis h from H if the Resolution mechanism derives an empty clause.

The algorithm `Select_Good_Hypotheses` receives a set H of candidate hypotheses that are all consistent and then operates as follows: (1) it computes the explanatory power and complexity of each hypothesis $h \in$ H, (2) removes all of the hypotheses h with explanatory power below some constant λ_1 (0.5 in our experiments) or complexity above some constant λ_2 (5 in our experiments), (3) computes set E with the hypotheses that have the maximum explanatory power in H, (4) computes set X with the hypotheses that have the minimum complexity in E and (5) returns set X as answer.

It can be proved that the Peirce algorithm computes three different types of solutions: (1) H = F when the theory and the accepted conditions does not allow Peirce algorithm to formulate explanatory hypotheses; (2) H = { } when Peirce algorithm does not consider any hypotheses to be good, among the candidate hypotheses; (3) H contains at least one explanatory hypothesis; In this last type, H does not contain non-explanatory hypothesis.

Example 5 illustrates a run of the algorithm Peirce.

Example 5. This example illustrates the execution of the algorithm Peirce using the scenario and formalization from Example 2. Therefore, the algorithm Peirce receives as input the theory T = $\{ r \rightarrow w, s \rightarrow w \}$, the set of accepted conditions C = $\{ t, t \rightarrow \neg s \}$ and the set of facts F = $\{ w \}$. Because T ∪ C $\nvDash \bot$, algorithm `Consistent(T, C)` returns the value true (line 2), and the data structure R is filled in with T ∪ C ∪ ¬F in CNF expressed with Horn Clauses. The following is then established:

- At line 4: R = $\{ \{ \neg r, w \}, \{ \neg s, w \}, \{ \text{t} \}, \{ \neg t, \neg s \}, \{ \neg w \} \}$;
- At line 5 after Resolution: R = $\{ \{ \neg r \}, \{ \neg s \} \}$.

Because there are no empty clauses in R (test at line 6), the candidate hypotheses are formulated at line 11. Thus, H = $\{ r, s, r \wedge s \}$ at line 11 after executing `Formulate_Candidate_Hypotheses`. Because T ∪ C ∪ $\{ s \} \vDash \bot$ and T ∪ C ∪ $\{ r \wedge s \} \vDash \bot$, hypotheses s and $r \wedge s$ are removed from H by the algorithm `Remove_Inconsistent_Hypotheses` (line 12), leaving H = $\{ r \}$. The hypothesis r has an explanatory power of 1, a complexity of 1 and the minimum complexity of all hypotheses with maximum explanatory power in H (r is the only hypothesis in H), therefore the algorithm `Select_Good_Hypotheses` (line 13) selects r as a good hypothesis. The algorithm Peirce thus returns as answer H = $\{ r \}$.

In general, the complexity of logic-based abduction is NP-complete [6]. However, the algorithm Peirce has a running time $O(n^{2+\lambda_2})$. As λ_2 is a constant, typically equals to 5, Pierce algorithm runs in polynomial time. This occurs by the following facts. The algorithm `Conjunctive_Normal_Form_Horn_Clauses` has running time $O(n)$ because since every sentence of T, C and F is restricted to HF form (Definition 4) they can be transformed directly into Horn Clauses in $O(1)$. The execution of the Resolution mechanism of the PL sentences in CNF with Horn Clauses can be done in $O(n^2)$. Thus, `Consistent` and `Resolution` algorithms have running time $O(n^2)$. The algorithm `Formulate_Candidate_Hypotheses` has a running time of $O(n^{\lambda_2})$ because produces at most hypotheses combinations $O(n^2) + O(n^3) + \ldots + O(n^{\lambda_2})$. The algorithm `Remove_Inconsistent_Hypotheses` has a running time of $O(n^{2+\lambda_2})$ because executes at most a constant amount of $O(n^{\lambda_2})$ resolutions each of them in $O(n^2)$. The algorithm `Select_Good_Hypotheses` has a running time of $O(n \log n)$, to sort and select the set of hypotheses with minimal complexity among the hypotheses with maximum explanatory power.

3 Tendency of Solutions Computed by the Algorithm Peirce to Coincide with Abductive Reasoning Done by Humans

A study was realized to verify whether the abductive reasoning performed by humans tends to coincide with the solutions computed by the algorithm Peirce. The study was conducted using a questionnaire containing ten questions, with each question presenting an implicit description of a theory, observed facts and accepted conditions. The alternatives for each question present possible abductive hypotheses. Table 3 illustrates in the left column one question in the questionnaire.

Table 3. Example of a question used in the questionnaire. The left column describes the question itself, and the right column presents the corresponding formalization to the question and solution as computed by the algorithm Peirce.

Question	Formalization and solution computed by the algorithm Peirce
Joshua is in the desert and sees something green in the distance. What would best explain what Joshua sees? a) I am convinced that it is a lawn. b) I am convinced that it is a cactus. c) I am convinced that it is a green flag. d) It could be either a cactus or a green flag.	**Propositions**: '*c*: It is a **cactus**', **'***d*: It is a **desert**', '*f*: It is a green **flag**', '*l*: It is a **lawn**', '*s*: Joshua **sees** something green'. T = { $l \rightarrow s, c \rightarrow s, f \rightarrow s$ }, C = { $d, d \rightarrow \neg l$ }, F = { s }. **Solution** - After formulating candidate hypotheses (line 11): H = { $l, c, f, l \wedge c, l \wedge f, c \wedge f, l \wedge c \wedge f$ }. - After removing inconsistent hypotheses (line 12): H = { $c, f, c \wedge f$ }. - After selecting good hypotheses (line 13): H = { c, f }, i.e., the alternative 'd' coincides with the solution of algorithm Peirce.

The questionnaire, validated by a pilot-test with 25 individuals, was designed to be answered in 15 minutes. A total of 133 undergraduate and graduate students participated in the study. The profile of the participants showed a slight predominance of female individuals (53%) and ages ranging from 18 to 60 years, with an average and median close to 25 years.

Each participant's answers to the questionnaire were computed, and one point was attributed to each answer on the questionnaire that coincided with a solution produced by the algorithm Peirce. The results showed an average of 86 answers coinciding with the algorithm Peirce and 47 that did not coincide.

The Chi-square (χ^2) test at 1% significance was used as a statistical measure of the significance with which the participants' answers coincided with solutions produced by the algorithm Peirce. For the studied population, the χ^2 test suggested that the coincidence between the participants' answers and the solutions computed by the algorithm Peirce was significant: (χ^2 (1) = 11.44, *p*–value = 0.001 < 0.01).

4 Related Works

Different approaches have been used to develop algorithms for abductive reasoning. Among the many contributions, there are proposals that use search techniques [15] and probabilistic reasoning over Bayesian Networks [7]. Logic approaches are based on two types of contributions: (1) proposal of new algorithms and (2) extension of traditional logical programming to process abductive reasoning problems.

Examples of type 1 contributions include [2] and [5]. Both of the proposals refer to abductive reasoning algorithms that operate with a THF reasoning framework (Definition 1). The main differences between these proposals and those of the present study are as follows: (1) They allow only one fact to be declared; (2) They do not allow define explicitly a set of accepted conditions; and (3) Semantic Tableaux is used in the proposal described in [2] instead of Resolution as the mechanism of inference.

Contributions of type 2 include Abductive Logic Programming (ALP) [11] and use the languages Prolog with Constraint Handling Rules (CHR) [1] [4]. The main differences between these proposals and those of the present study are as follows: (1) They operate with Predicate Logic; (2) They require special "abducible" predicates (possible hypotheses) to be declared; and (3) They dedicate little attention to criteria to select good abductive hypotheses.

Studies related to the one presented here, that address the selection of good hypotheses, include [8], [16] and, recently, [3]. This work differs from proposals [8] and [16] mainly by the proposed metrics for complexity and explanatory power of hypotheses.

5 Conclusions

The abductive reasoning algorithm Peirce is distinct from other solutions mainly because it employs the TCHF reasoning framework and a simple criterion for selecting good hypotheses that consider quantitative metrics to define the explanatory power and complexity of the formulated hypotheses.

The TCHF reasoning framework has shown itself to be useful in organizing the elements that participate in abductive reasoning because it does not "contaminate" the theory with sentences that fundamentally do not belong to the theory. This framework provides an additional advantage because it explicitly exposes the conditions (contexts, circumstances, intentions etc.) under which the reasoning process is conducted, which is fundamental and frequent in the formulation of abductive reasoning.

The criteria for selecting good hypotheses are subjects of ongoing research. There is no consensus as to which criteria should be used and under which circumstances or in which domains they work. The criterion used by the algorithm Peirce, which is described in Definition 6, attempt to produce a simple algorithm that works in practice. Alternatives to Definition 6 exist and can be proposed.

The study that depicted the coincidence of the solutions computed by the algorithm Peirce to those derived through abductive reasoning performed by humans was not exhaustive because there is such a high number of domains, and it did not include the diversity and quantity of individuals. However, these results provided value suggesting that the abductive reasoning conducted by humans tends to coincide with the solutions computed by the algorithm Peirce.

References

1. Alberti, M., Gavanelli, M., Lamma, E.: The CHR-based Implementation of the SCIFF Abductive System. Fundamenta Informaticae 124 (4), pp. 365–381 (2013)
2. Aliseda, A.: Abductive Reasoning: Logical Investigations into Discovery and Explanation. Springer, Netherlands (2006)
3. Caroprese, L., Trubitsyna, I., Truszczynski, M., Zumpano, E.: A Measure of Arbitrariness in Abductive Explanations. To appear in Theory and Practice of Logic Programming, 25 p (2014)
4. Christiansen, H.: Executable specifications for hypothesis-based reasoning with Prolog and Constraint Handling Rules. Journal of Applied Logic 7 (3), pp. 341–362 (2008)
5. Dillig, I., Dillig, T.: Explain: A Tool for Performing Abductive Inference. In Sharygina, N., Veith, H. (eds) CAV 2013. LNCS, vol. 8044, pp. 684—689. Springer, Heidelberg (2013)
6. Eiter, T., Gottlob, G.: The Complexity of Logic-Based Abduction. Journal of the Association for Computing Machinery, 42 (1), pp. 3–42 (1995)
7. Fortier, N., Sheppard, J., Strasser, S.: Abductive inference in Bayesian networks using distributed overlapping swarm intelligence. Soft Computing Journal, May 2014, pp. 1–21 (2014)
8. Harman, G. H.: The inference to the best explanation. The Philosophical Review 74 (1), pp. 88–95 (1965)
9. Howard, P.: Introduction to Logic: Propositional Logic. Prentice-Hall, New Jersey (1999)
10. Josephson, J. R., Josephson, S. G.: Abductive Inference: Computation, Philosophy, Technology. Cambridge University Press, Cambridge (1994)
11. Kakas, A. C., Kowalski, R. A., Toni, F.: Abductive Logic Programming. Journal of Logic and Computation 2 (6), pp. 719–770 (1995)
12. Mackenzie, I. S.: Human-Computer Interaction: An Empirical Research Perspective. Morgan Kaufmann, New York (2013)
13. Magnani, L.: Abductive Cognition: The Epistemological and Eco-cognitive Dimensions of Hypothetical Reasoning. Springer, Berlin (2009)
14. Peirce, C. S.: Collected Papers of Charles Sanders Peirce. Oxford University Press, London (1958)
15. Romdhane, L. B., Ayeb, B.: An Evolutionary Algorithm for Abductive Reasoning. Journal of Experimental & Theoretical Artificial Intelligence 23, pp. 529–544 (2011)

16. Thagard, P. R.: The Best Explanation: Criteria for Theory Choice. The journal of philosophy 75 (2), pp. 76–92 (1978)
17. Thagard, P. R.: Explanatory Coherence. Behavioral and Brain Sciences 12, pp. 435–502 (1989)

Tracer Trails Urban Transport System of the City of Villahermosa: Case of Transbus

Emmanuel Palomera May, Gerardo Arceo Moheno, Guillermo de los Santos Torres, Martha Patricia Silva Payro, Pablo Payró Campos, and José Adán Hernández Nolasco

Division Académica de Informática y Sistemas, Universidad Juárez Autónoma de Tabasco, Villahermosa, Mexico

emanuelpmay@gmail.com

Abstract. The city of Villahermosa, considered the gateway to southern Mexico, has experienced growth due to the economic activities of the region. This growth has not been based on an urban development plan where a suitable system of streets and avenues is contemplated, which impacts the service of the public transportation system, which consists of the mass transit system urban (vans), the rental car service (taxis), Transbus system (with routes within the city) and Transmetropolitano system (covering the metropolitan area of the city). According to the above, it is necessary to have an application support to determine the transfer of a part of the city to another in an efficient and intuitive way. The proposed tool (in the first phase only Transbus system) makes use of the PRM (Probabilistic Road Map) and Dijkstra algorithm to find a path Shuttle routes Transbus using interfaces in his version of HTML 5, jQuery, CSS and GoogleMaps, having the ability to adapt to different screen sizes of devices, thus enhancing the ways in which the user can have the tool.

Keywords: Tracer, probabilistic roadmap, Dijkstra algorithm.

1 Introduction

Transportation systems are key components in the social, economic and physical structures of an urban area [1]. In economic terms, the urban transport enables the reproduction of the labor force through the massive displacement of labor, by increasing the economies of scale and overall productivity of the city. Town planning, urban transport has effects on the size and configuration socio - spatial city. Culturally, enables different social relations to the strictly productive and generates spaces in which citizens can present and imagine the city and others [2].

The city of Villahermosa, as the capital of Tabasco state, receives a large tributary of people from different parts of the state, or outside, being necessary to provide adequate means of transport to travel within the city, in addition to signal properly routes and stops covering these transportation services; between these services is the so-called Transbus.

At its inception, the service Transbus consisted of 210 buses equipped with air conditioning and television screen, which replaced 1,293 vans with over 12 years providing the service. Having not only the challenge of modernity urban public transport but expedite the move within the city of Villahermosa, it is important to mention that in just four years of operation Transbus system has mobilized nearly 122 million users until December 2012, mobilizing about 150 thousand users daily [3].

Moreover, as published by the INEGI [4], Tabasco in 2011, less than 35% of the population had access to computers, less than 30% had access to fixed Internet services and only 16% have access in their households; however, 56% have access to mobile phone services and 99.1% of households have an internet connection broadband.

Although no official data updated by gender and age, in a poll by Consulta Mitofsky [5] shows that 56% of men, 51% of women and two of every three Mexicans under 30 have access to a cellular phone. As for the age range 18 to 29 years, 61% said they know all the features of your cell phone [6].

In the National Development Plan (NDP) promotes sustainable urban mobility projects supported by public and mass transit, and promotes the use of non-motorized transport [7]. Thus, following the guiding principles of the NDP, in the State Development Plan (EDP) arises stimulate sustainable human mobility options that favor the use of non-polluting means of collective transport [8].

In this way having the commitment to promote the use of public transport and the growth media in the state of Tabasco, the idea of having a tool based on information technology, which allows the method to move within the city of Villahermosa through service offering Transbus units, allowing greater utilization from persons who, supported by the tool, Transbus can make use of economic resources. The development was guided by adaptive software development methodology to implement the probabilistic roadmap method for obtaining results.

2 Description of the Method

2.1 Methodological Approach

We used an action research approach. The purpose of it is solve everyday problems and immediate; it is understandable to make the social world and to improve the quality of life of people and immediate solving [9]. This is defined as "the study of a social situation with a view to improving the quality of action within it". It involves understanding the teaching office, integrating reflection and intellectual work in the analysis of the experiences carried out. Similarly, Sampieri et al. [10] define it as a "small-scale intervention in the functioning of the real world and the thorough examination of these interventions".

The sources of information were the primary and secondary type. The instruments of data collection were unstructured interviews made to the officer in charge of public transport area and documentation provided by the Ministry of Communications and Transport of the State of Tabasco (SCT).

2.2 Development Methodology

The methodology used for development is Adaptive Software Development (ASD) that is an agile methodology that has the following characteristics [11]:

- Iterative.
- Cyclic.
- Fault tolerant.
- Guided by the risks.

The review of the components allow us to learn from mistakes and not force commitments to restart the development cycle. This methodology consists of three stages (speculate, collaborate and learn), which are shown in Figure 1, same to be used repeatedly until the routes obtained are satisfactory in quality testing.

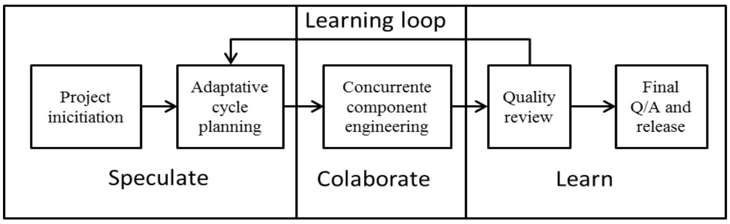


Fig. 1. Description of our methodology.

2.3 Probabilistic Roadmap

As Kavraki [12] mention, probabilistic roadmap method (PRM) has two phases: the learning and the query, each one comprised of different steps. The PRM method was implemented in conjunction with the Dijkstra algorithm to offer the user the shortest route to your destination:

- Learning phase
  - Mapping routes of Transbus,
  - Correction and debugging routes,
  - Connect the nodes generated.
- Query phase
  - Determine the start and end point,
  - Switch start and end with the roadmap generated,
  - Apply Dijkstra's algorithm,
  - Show results.

First, mapping routes of Transbus is performed to obtain a map that has paths routes. Post-mapping and integrating routes in a single environment, we corrected the routes by eliminating irrelevant points. Then, we proceed to enter the vicinity of the nodes generated within the database for later use in the consultation phase.

Already developed the roadmap is possible to make queries, i.e., ask if there is a way to move from a start point to an end point. The start point and end point are determined depending on the proximity there is a node in the roadmap.

To run a query, first the initial and final roadmap, after this the Dijkstra algorithm is implemented to obtain the shortest path between nodes connected configurations. Finally, the sequence of nodes becomes a feasible route, that is, the path consists of three sub-paths connecting the initial configuration to a node in the roadmap, the sub-path present in the roadmap between the two connection nodes and the sub-path connecting a node in the roadmap to the final configuration. This is shown through the interfaces previously elaborated.

3 The Learning Phase

3.1 Mapping Routes of Transbus

Transbus routes were obtained from different sources such as page Juarez Autonomous University of Tabasco provides students and the general public a means to know the Transbus routes through a PDF file. In figure 2 shown the M1 route comprising of Pino Suarez Market Ixtacomitan.

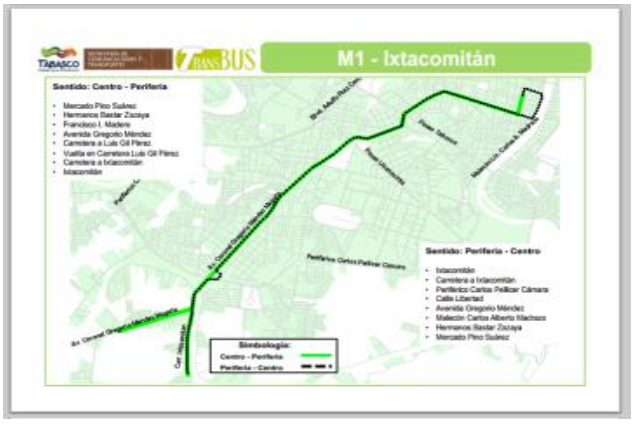


Fig. 2. Route M1 Ixtacomitan.

With the routes already obtained, we proceeded to develop the course of each route using the lite version of Google Maps Engine, as shown in Figure 3.

3.2 Correcting and Debugging Routes

With the possibility that the user can observe the path of the individual paths, the route was elaborated in detail using a high amount of nodes in each route, including the direction of each route.

The number of nodes used on each route is decreased by replacing the segments where the journey is made in both directions, with one segment that represents both directions through adjacencies in the next step of the PRM.

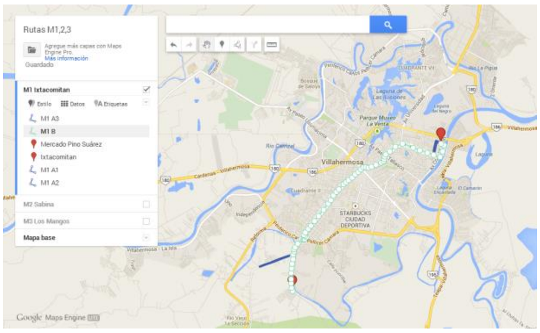


Fig. 3. Development of the route M1.

Many nodes in the path are removed, leaving only the nodes representing established by the Transbus stops, which are located at an approximate distance of 200 feet between them. Decreasing the number of nodes involved in the roadmap, decrements the running time Dijkstra algorithm, which is shown in Figure 4.

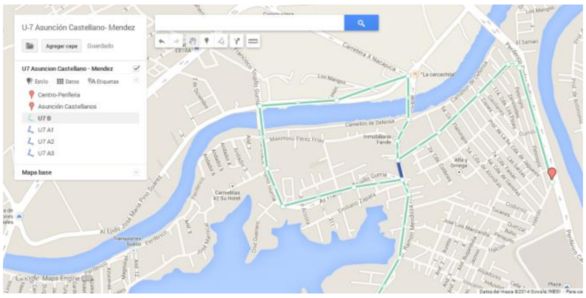


Fig. 4. Debug output node.

Having made all the routes, we proceeded to perform an integration of the 30 routes under a single environment, because Google Maps Engine imposes a limit on the number of layers available for use in the lite version. The integration was developed by modifying routes KML files generated for each route, extracting code segment with the sequence of nodes of a route identified with the <LineString> shown. The result of the integration can be seen in Figure 5.

After performing the integration of routes, were grouped according to with the sense, i.e., the group where the tour takes place from the periphery to the center and the other group where the tour takes place from the center to the periphery of the city, obtaining a map as shown in Figure 6, which was used to identify existing adjacency between the nodes.

Once the integration and aggregation of the routes are done, the route is cleared in a single layer to unify the paths where it passes over other routes, therefore, simplifying the roadmap. The result is seen in Figure 7.

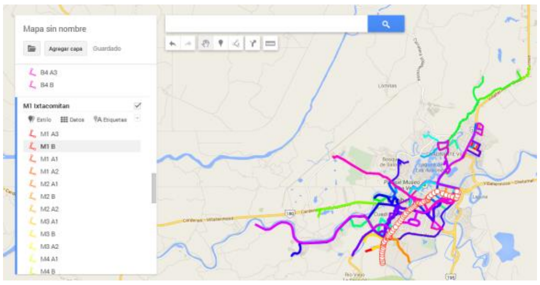


Fig. 5. Integrating routes.

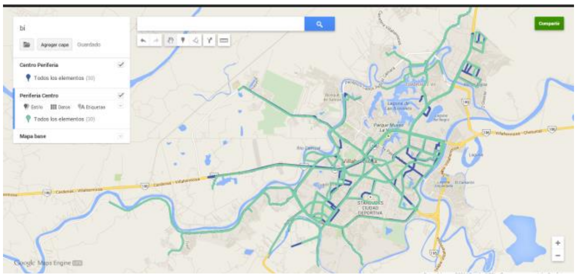


Fig. 6. Grouping routes.

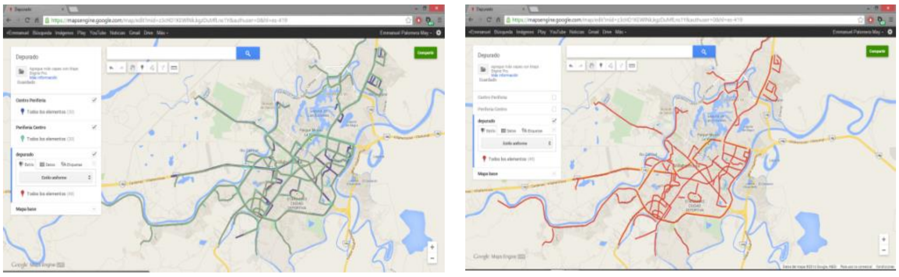


Fig. 7. Debugging nodes.

Once the debugging of route is done, the KML file containing the roadmap is processed with a tool available online (kml2x), converting the KML file to a plain text file that can be handled more easily.

3.3 Connecting Nodes

The adjacency of the nodes was determined according to the senses of Transbus routes and was introduced to the database. Using PHP to generating an array in

JavaScript that contains the location of the nodes, latitude and longitude, as well as adjacencies. It is possible to display input and adjacent nodes on a map as shown in Figure 8.

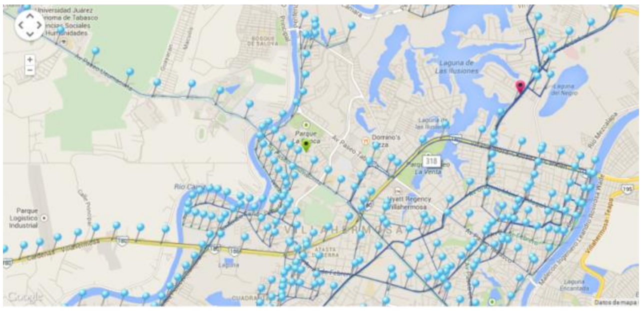
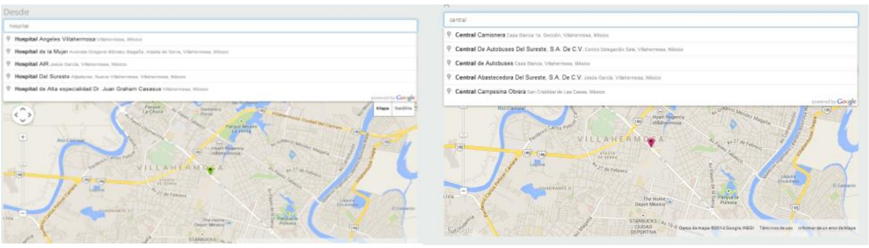


Fig. 8. Nodes on the map.

4 The Query Phase

4.1 Setting the Start and End Points

The user -through the user interface- determines the start and end points by using the integrated library of Google Maps Place. It has a feature called autocomplete that shows a list of the places that match the name you enter. It is noteworthy that the places listed are located solely within the city of Villahermosa (see Figure 9).


Desde

hospital

- **Hospital** Angeles Villahermosa Villahermosa, México
- **Hospital** de la Mujer Avenida Gregorio Méndez Magaña, Atasta de Serra, Villahermosa, México
- **Hospital** AIR Jesús García, Villahermosa, México
- **Hospital** Del Sureste Adyacente, Nueva Villahermosa, Villahermosa, México
- **Hospital** de Alta especialidad Dr. Juan Graham Casasus Villahermosa, México

A

central

- **Central** Camionera Casa Blanca 1a. Sección, Villahermosa, México
- **Central** De Autobuses Del Sureste, S.A. De C.V. Centro Delegación Seis, Villahermosa, México
- **Central** de Autobuses Casa Blanca, Villahermosa, México
- **Central** Abastecedora Del Sureste, S.A. De C.V. Jesús García, Villahermosa, México
- **Central** Campesina Obrera San Cristóbal de Las Casas, México

Fig. 9. Setting the start and end points.

The user adjusts the start and end points by moving the marker in the map in order to increase the accuracy of the results.

4.2 Connecting Start and End Points with the Roadmap

To connect the start and end points given by the user to the roadmap, is to determine the closest node to the start and end points, having to calculate the distance between all nodes with respect to the start point and go always keeping the node with the smallest distance. The pseudocode for determining the nearest node is shown in Figure 10.

```
program begin
var distance, tmp = infinity, closer, u
u = lat and lng
start for cycle i < numNodes
 distance = distance between nodes u and [i]
 if distance <tmp
  tmp = distance
  closer = i
 end if
 increase i
end for cycle
program end
```

Fig. 10. Pseudocode for determining the nearest node.

4.3 Applying the Dijkstra Algorithm

After determining the start and end points in the roadmap, Dijkstra's algorithm is run. The algorithm uses as an input the start and end values determined in the previous step, obtaining the identifier of the nodes that are present in the shortest path between them. After obtaining the identifier of the nodes, we proceed to generate a sequence with the location of the nodes, i.e., latitude and longitude of each node on the path generated to be represented to the user via a map (see Figure 12). The way to proceed is shown in Figure 11.

```
program begin
var x = determine start node
var y = determine end node
run Dijkstra (x, y)
path obtained from x, and y
show path between x, y
show path between start - x
show path between end - and y
program end
```

Fig. 11. Algorithm for generating the sequence of location nodes.

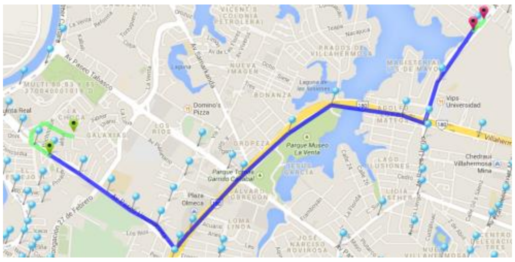


Fig. 12. Shortest path.

Distance between nodes is used as the only criterion to determine the shortest path, as the service has Transbus lanes on the avenues more vehicular affluent. Also, it is not possible to determine the transfer time since the speeds are not constant, nor are monitored.

4.4 Show Results

Finally, it should show the user the path to follow from the start point to the starting node in the roadmap and the way forward between the end node and the end point roadmap introduced by the user (see Figure 13).

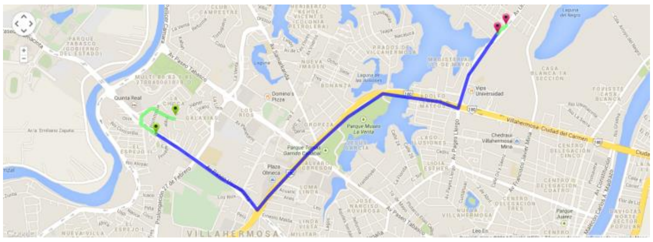
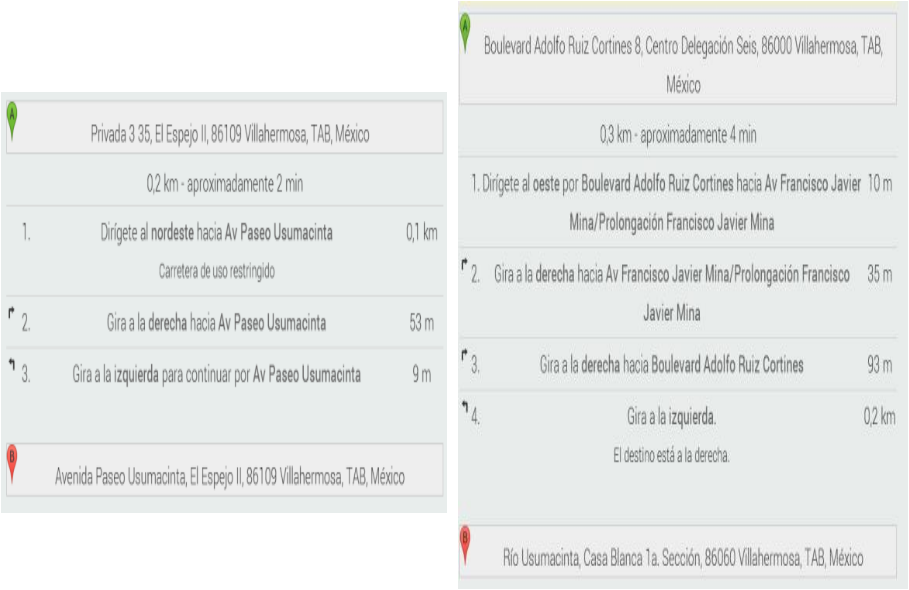


Fig. 13. Result displayed to the user.

Besides showing graphically the shortest path between the start and end nodes, also displayed in textual sequence of steps to go from source to destination (see Figure 14).



A Privada 3 35, El Espejo II, 86109 Villahermosa, TAB, México

0,2 km - aproximadamente 2 min

1. Dirígete al **nordeste** hacia **Av Paseo Usumacinta** 0,1 km
Carretera de uso restringido

2. Gira a la **derecha** hacia **Av Paseo Usumacinta** 53 m

3. Gira a la **izquierda** para continuar por **Av Paseo Usumacinta** 9 m

B Avenida Paseo Usumacinta, El Espejo II, 86109 Villahermosa, TAB, México

A Boulevard Adolfo Ruiz Cortines 8, Centro Delegación Seis, 86000 Villahermosa, TAB, México

0,3 km - aproximadamente 4 min

1. Dirígete al **oeste** por **Boulevard Adolfo Ruiz Cortines** hacia **Av Francisco Javier Mina/Prolongación Francisco Javier Mina** 10 m

2. Gira a la **derecha** hacia **Av Francisco Javier Mina/Prolongación Francisco Javier Mina** 35 m

3. Gira a la **derecha** hacia **Boulevard Adolfo Ruiz Cortines** 93 m

4. Gira a la **izquierda**. 0,2 km
El destino está a la derecha.

B Río Usumacinta, Casa Blanca 1a. Sección, 86060 Villahermosa, TAB, México

Fig. 14. Route instructions.

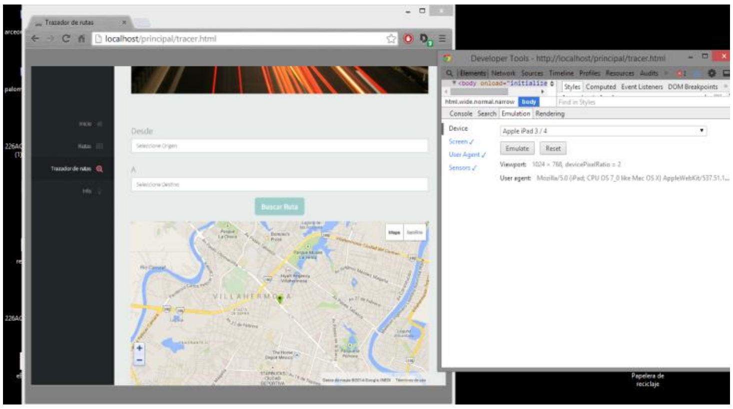
4.5 Performance Results

A roadmap consisting of 530 nodes representing the path of the 30 service routes Transbus was generated. The roadmap adjacency node was pointing in both directions, indicating that each segment of the roadmap could go either way, i.e., center to periphery or periphery to center. This roadmap primarily benefit was the processing time, which averaged 233 milliseconds. However, it was not possible to determine the paths involved in the route generated so a second roadmap which had 1041 nodes was developed. The execution time was not significantly affected: the average time was 273 milliseconds, i.e., there was only an increase of 40 milliseconds processing time, which is irrelevant in view of the increase in the number of nodes.

The major problem of the roadmap of 1041 nodes was that if the tour began in a segment where the addresses were not separated, the identification of appropriate routes for the transfer was not done correctly. That is the reason for separating the management of the routes in the entire roadmap. So, 1325 roadmap nodes were separate of the paths and more accurate results were added. The processing time for the 1325 nodes test runs was an average 279 milliseconds, i.e., there was an increase of only 6 milliseconds with respect to the processing time of the roadmap of 1041 nodes and an increase of 46 milliseconds relative to the roadmap 531 nodes. By using this roadmap, unlike the previously generated, it was possible to correctly identify Transbus routes present in the path.

Table 1. Performance tests.

Number of Nodes	Execution time
530	233 milliseconds
1041	273 milliseconds
1325	279 milliseconds



Developer Tools - http://localhost/principal/tracer.html

Device: Apple iPad 3 / 4

Viewport: 1024 × 768, devicePixelRatio = 2

User agent: Mozilla/5.0 (iPad; CPU OS 7_0 like Mac OS X) AppleWebKit/537.51.1...

Fig. 15. Interface adapted to the dimensions of an iPad.

5 Interfaces

Interfaces were developed using Html5, CSS and jQuery, getting results in a site that conforms to the resolution of the device from which it is viewed. As seen in Figure 15, the interface adapts to the screen resolution of an iPad.

6 Conclusions

The solution proposed in this paper describes how we can apply the method of probabilistic maps in conjunction with the Dijkstra algorithm, to solve the problem of travel within the city of Villahermosa using the routes offered by Transbus. Among the benefits that users can contribute to mention the ease of use of the site, the pleasant manner in which indications are provided for transfers using Transbus routes, proper way to adapt interfaces different devices and that the site is freely accessible via the Internet.

In tests, users felt satisfied with the results generated by the site, as of today there is a similar tool to make transfers within the city of Villahermosa using the public transport system. The solution is applicable to other means of transportation, and while the introduction of routes does not occur automatically but manually, the results displayed to the user are performed properly.

There is also the option to develop a mobile application, although better to control the user's experience, they are costly to develop and are complicated to migrate platforms or operating systems as appropriate.

References

1. Krishna Rao, K. V., Muralidhar, S., & Dhingra, S. L.: Public transport routing and scheduling using Genetic Algorithms. In 8th international conference on computer aided scheduling of public transport, Berlin, Germany (2000)
2. Canclini, N. G., Castellanos, A., & Mantecón, A. R.: La ciudad de los viajeros: travesías e imaginarios urbanos, México, 1940-2000. Grijalbo (1996)
3. De la Cruz, A.: Movilización Transbus a 122 millones de usuarios en 4 años. *De Facto*. http://www.defacto.com.mx/v2/sureste_nota.php?id=7186 (2012)
4. INEGI (Instituto Nacional de Estadística, Geografía e Informática). Estadísticas sobre disponibilidad y uso de tecnología de información y comunicaciones en los Hogares, 2011. México, INEGI (2012)
5. Consulta Mitofsky.: La generación milenio... Cuando el destino nos alcance (2013)
6. Consulta Mitofsky. México: usuarios de telefonía celular (2013)
7. Peña-Nieto, E.: Plan Nacional de Desarrollo. Diario Oficial de la Federación. México: Secretaría de Gobernación (2013)
8. Núñez, A.: Plan estatal de desarrollo. Gobierno del Estado de Tabasco (2013)
9. Álvarez-Gayou, J. L.: Cómo hacer investigación cualitativa. Fundamentos y metodología. Métodos básicos. Ed. Paidós. México, 127-8 (2005)

10. Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P.: Metodología de la investigación. México: Editorial Mc Graw Hill (2010)
11. Highsmith, J.: Adaptive software development: a collaborative approach to managing complex systems. Addison-Wesley (2013)
12. Kavraki, L. E., Svestka, P., Latombe, J. C., & Overmars, M. H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. Robotics and Automation, IEEE Transactions on, 12(4), pp. 566–580 (1996)

Implementation of a Two-tier Double Auction for On-line Power Purchasing in the Simulation of a Distributed Intelligent Cyber-Physical System

Denise M. Case[1], M. Nazif Faqiry[1], Bodhisattwa P. Majumder[2], Sanjoy Das[1], and Scott A. DeLoach[1]

[1] Kansas State University, College of Engineering,
USA
[2] Jadavpur University, Electronics and Telecommunication Engineering,
India

dmcase@ksu.edu, mnfaqiry@ksu.edu, sdas@ksu.edu, sdeloach@ksu.edu, mbodhisattwa@gmail.com

Abstract. The increasing penetration of distributed renewable generation brings new power producers to the market [2]. Rooftop photovoltaic (PV) panels allow *home owners* to generate more power than personally needed and this excess production could be voluntarily sold to nearby homes, alleviating additional transmission costs especially in rural areas [24]. Power is sold as a continuous quantity and power markets involve pricing that may change on a minute-to-minute basis. *Forward markets* assist with scheduling power in advance [25]. The speed and complexity of the calculations needed to support online distributed auctions is a good fit for intelligent agents [14]. This paper describes the simulation of a *two-tier double auction* for short-term forward power exchanges between participants at the outer edges of a power distribution system (PDS). The paper describes the double auction algorithms and demonstrates online auction execution in a simulated distributed system of intelligent agents assisting with voltage/var control near distributed renewable generation [15]. The agents were enhanced to autonomously create local power market organizations and execute the series of online power auctions using Advanced Message Queuing Protocol (AMQP).

Keywords: Smart grid, power market, online auction, double auction, intelligent systems, cyber-physical systems.

1 Introduction

Distributed intelligent systems will support a variety of objectives for power distribution systems (PDS) [7]. Devices installed in smart cyber-physical systems may support a variety of objectives [19]. For example, smart meters and smart inverters installed in residential homes may be enhanced to provide assistance with voltage regulation during periods of intermittent solar generation. These control functions would fall under the adaptive control responses specified by the local utility, but may also be subject to the various objectives and interests of the homeowners. A smart system running on or near the smart meter may be a likely candidate to support the brokering of online power sales agreements between homeowners and the grid. The work presented examines a multiagent system architecture capable of adaptively controlling future PDS while simultaneously supporting concurrent calculations for bidding and brokering online sales agreements among the stakeholders.

The paper introduces a two-tier double auction scheme where home prosumer agents create bids to express their intentions and send them to an agent acting as the broker in a local power market organization. The agent brokering the local auction determines the optimal resolution of the auction, and in the event of any unsatisfied amounts, participates as a bidder in a secondary, higher-level auction. The approach exploits the applicability of the double auction in the second-tier, where the auction takes place between the secondary participants representing their remaining community bids and shows the efficacy of the proposed hierarchical model as it further maximizes the overall social utility.

The project demonstrates an architecture for multigroup agents that provides a modular, extensible approach for supporting agents participating in multiple affiliated and independent groups, each with their own behavior specification, while providing a means to customize the intelligent agents based on homeowner preferences and personal market strategies.

The remainder of this paper is organized as follows. Motivation and related work is presented in Section 2. The double auction algorithm is defined in Section 3. In Section 4 we describe the implementation in a holonic multiagent system, and the simulation test case is presented in Section 5. Finally, we present the results Section 6 and our conclusions and recommendations for further work in Section 7.

2 Motivation and Related Work

The motivation for our work grows out of research into several two-tier resource allocation techniques. Most specifically, that of spectral allocation such as Zhou's [26] where a two-tier resource allocation approach has been proposed that integrates a dispatcher-based node partitioning scheme with a server-based dynamic allocation scheme. Also, the work of Abdelnasser, et. al. [1], that also proposes a semi-distributed (hierarchical) interference management scheme based on resource allocation for femtocells. In addition, several other market-based economic models have been proposed for the process of competitive buying and selling to solve for an optimal power flow in a smart grid. Local interactions [10] and decentralized resource scheduling [5] have been considered with better convergence under tight computational budget constraints. Auctions are an efficient mechanism, easily implemented in a grid structure, that allows buyers and sellers to compete for the resources to be auctioned to achieve an optimal resource flow in order to maximize the social benefits to the participants. *Double auctions* are auctions that involve both buyers and sellers. These auctions can be designed as an efficient, incentive-compatible mechanism where buyers and sellers participate without the risk of losing anything by choosing to participate. A recent study on auctions for spectrum allocation in wireless networks [21] has shown that a double auction can achieve a greater social welfare (benefit) compared to other auction mechanisms such as the well-known Vickrey–Clarke–Groves (VCG) mechanism [20]. An efficient double auction mechanism with uniform pricing has been proposed by Weng, et. al. [23], that considers the dynamic, heterogeneous and autonomous characteristics of resources in a grid computing system. Wang, et al. [22], developed and analyzed the double auction as a mechanism to characterize the trading price of the energy trading market that involves the storage units and the potential energy buyers in the grid. Furthermore, several applications [9], [13], [11] have been proposed in a different field of study and have been shown as an effective mechanism when interest of both buyers and sellers are taken into consideration for a competitive market happening in a computational grid system. To the best of our knowledge, we believe there is no existing literature where a double auction has been implemented in a hierarchical manner for electricity trading in isolated micrograds to achieve a greater social benefit in power distribution systems. We believe our implementation in a two-tiered structure, comprised of intelligent agents participating in the auction by sending messages to the auctioneer indicating an interest to buy or sell, demonstrates a novel and potentially useful approach due to the following reasons.

First, the proposed two-tier approach implements bids in two stages, in the first tier, the auction involves individual homes within a neighborhood acting as buying and selling agents. In the second tier, an auction between multiple neighborhoods takes place, with each neighborhood modeled as an agent. This arrangement follows the spatial topology of the power distribution system, where feeders deliver power via several transformers to the neighborhoods. Hence, our suggested approach can be implemented easily into existing distribution systems without the need for additional channels for information exchange, with agents at the transformers and the feeder acting as brokers.

Secondly, the double auction mechanism that our suggested approach entails are formulated as linear programming problems known to be of exponential complexity. The tiered-approach can be perceived as a divide-and-conquer scheme that divides the larger auction problem at the feeder level into several smaller, more tractable sub-problems, one corresponding to each neighborhood, that are solved in a parallel fashion.

Lastly, the constraints imposed upon the auctions taking place at the first tier and second tier are different. It can be assumed that across individual homes within a small geographical neighborhood would entail an underlying well-connected social group. Hence, the demands or supplies of electricity of individual homes at any given instance can be gleaned either from historical data or from prediction algorithms. These can serve as bids for the first tier auction, obviating the need for direct human intervention. Furthermore, the energy pricing must be uniform across the entire neighborhood. These requirements need not hold at the feeder level, where each neighborhood may be priced differently. Furthermore, due to larger geographic distances between neighborhoods, the auction may have to take into account additional factors such as I^2R loss, local cloud conditions, etc. [8]. Some of these issues, not currently taken into account, could be readily incorporated with minor modifications.

3 Two-tier Double Auction

Power distribution systems (PDS) often form a natural hierarchy, with a single central substation distributing power through a tree-based network of 3-phase feeder lines, down through single phase lateral lines, and out through neighborhood transformers to the lowest, most distributed layer of residential homes. A hierarchical cyber-physical system (CPS) such as a PDS includes both a physical component and a computational component, and may be referred to as holonic (the word comes from the greek words for both *whole* and *part*) [12]. The overall multiagent system (MAS) acts as a complex MAS, or a system of systems. Each level of the *holorchy*, may consist of one or more local organizations [18]. All organizations operating in intermediate levels, i.e., not the top level substation or lowest level homes, can be viewed as operating in two separate, but analogous, local organizations.

In our two-tier double auction, each *home prosumer agent* (*prosumer* indicates the ability to both *produce* and *consume* electrical power) participates in a single holonic organization at the lowest level of the holarchy. Each of these lowest level organizations includes a neighborhood transformer agent that may be situated on or near the pole transformer that supplies a small set of homes with power. For testing, we assumed each neighborhood transformer agent supports four homes supplied by the associated transformer, one of which has rooftop photovoltaic (PV) panels for generation.

Each neighborhood transformer agent was equipped to broker a local power market auction, accepting bids from the four participating homes to exchange power at a given future time period. Homes equipped with rooftop solar panels were assumed to have surplus power to sell that nearby homes (those served by the same transformer) could bid on. The neighborhood transformer agent and and the homes supplied by the transformer would autonomously create a small local power market organization and execute the auction. Each neighborhood transformer agent also further equipped to auction power at a higher level. In these secondary auctions, the neighborhood transformer agents served in a different role. In the higher organization, each neighborhood agent served as an auction participant, while the single lateral power line agent, supplying power to several neighborhoods, was equipped to accept their bids and serve as broker in the second-tier double auction.

The holonic nature of these local power market organizations is illustrated in Figure 1. Home prosumer agents bid in first-tier auctions brokered by agents

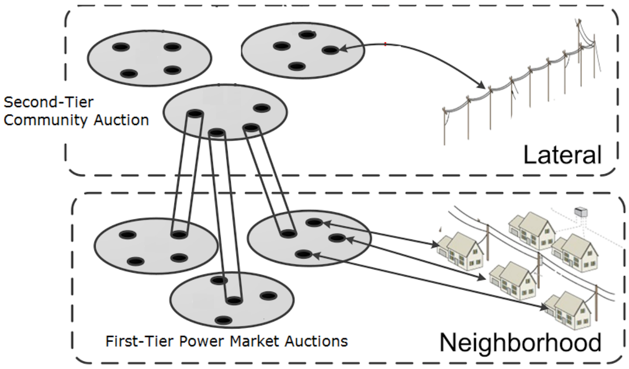


Fig. 1. Holonic power market organizational structure for the two-tier, distributed double-auction simulation.

running on neighborhood transformers. Neighborhood agents then bid in second-tier auctions brokered by an agent running on their supplying power line.

3.1 First-Tier Auction

At the first tier of the proposed scheme, each of the neighborhood transformer agents (indexed $k \in 1, 2, ...N$) conducts an independent auction from the bids provided by the home prosumer agents supplied by the associated transformer. In each local first-tier auction, we let N_B^k and N_S^k be the number of potential buyers and sellers with indexes i and j, respectively, their bid prices per unit of energy be $c_{b,i}$ and $c_{s,j}$, and their maximum demands and available supplies (in energy units) be d_i and s_j. With denoting c_0^k the clearing price per unit of power, the agents utilities can be defined as follows. For buyers:

$$u_{b,i} = \begin{cases} (c_0^k - c_{b,i})q_{b,i}, & c_{b,i} \geq c_0^k \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

and for sellers:

$$u_{s,j} = \begin{cases} (c_0^k - c_{s,j})q_{s,j}, & c_{s,j} \geq c_0^k \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Here, the volumes of energy $q_{b,i}$ and $q_{s,j}$ bought and sold are determined through the auction by maximizing the total utility of all participating agents, U^k. With p^k being the assigned energy volume imported (exported when positive) to neighborhood k, the underlying auction is formulated as the following linear programming problem. Maximize:

$$U^k = \sum_{i \in W_B^k} u_{b,i} + \sum_{j \in W_S^k} u_{s,j} \tag{3}$$

subject to:

$$0 \leq q_{b,i} \leq d_i \tag{4}$$

$$0 \leq q_{s,j} \leq s_j \tag{5}$$

$$\sum_{j \in W_S^k} q_{s,j} - \sum_{i \in W_B^k} q_{b,j} = b^k \tag{6}$$

The neighborhood transformer agent, serving as the broker, places the quantities b^k and c_0^k as the bid volume and price, respectively.

3.2 Second-Tier Auction

This secondary auction requires the power requested from each neighborhood transformer agent k, to serve as the neighborhood bid volume b^k and clearing price c_0^k. The lateral feeder line agent serves as the broker in the second-tier auction and determines the final clearing price c_0 at which subsequent power trading occurs and the power flow from each power-exporting neighborhood l to every power-importing neighborhood k. There are various ways in which the clearing price may be determined, e.g. through negotiations with the utility company, to obtain budget balance, or by other means. These issues are not addressed here, and a price c_0 is determined somewhat arbitrarily, to lie within the range of prices in the neighborhoods bids. This clearing price determines the *winner sets*, i.e. the set of neighbors that ultimately participate in the auction, either as buyers or sellers as defined below.

$$W_l = \{k | \, b^k \leq 0, c_0^k \geq c_0\} \tag{7}$$

$$W_E = \{k | \, b^k \leq 0, c_0^k \geq c_0\} \tag{8}$$

The objective of the auction is to maximize the social welfare function (SWF), i.e. the aggregated utilities of all winners, as provided in the following equation.

$$SWF = \sum_{k=1}^{N} U^k \tag{9}$$

The neighborhoods' utilities as seen by the broker in this tier are now determined as follows.

$$U^k = \begin{cases} (c_0^k - c_0)p^k, & k \in W_l \\ (c_0 - c_0^k)p^k, & k \in W_E \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

This allows the SWF to be expressed directly in terms of the bids in the following linear programming formulation to obtain the power flows $P^{k,l}$. Maximize:

$$SWF = \sum_{k \in W_l} \sum_{i \in W_E} (c_0^k - c_0^l)p^{k,i} \tag{11}$$

subject to:

$$p^k = \sum_{l=1}^{N} p^{k,l} \tag{12}$$

$$\begin{cases} 0 \leq p^k \leq b^k, & k \in W_l \\ p^k \leq b^k \leq 0, & k \in W_E \end{cases} \tag{13}$$

$$\sum_{k \in W_l} p^k + \sum_{l \in W_E} p^l = 0. \quad \text{(power balance)} \tag{14}$$

The power balance constraint above assumes an isolated microgrid that does not transfer power from external sources. Since we have assumed a single clearing price in the simulations discussed here, the approach is strongly budget balanced. However, it should be noted that the above problem can be reformulated in various ways, in which case a strong budget balance requirement may be added as another constraint.

4 Holonic Multiagent System Implementation

In addition to the computational approach for the auctions, we wanted to evaluate the ability to extend an existing intelligent power distribution system so that it might be able to support future power markets. For example, future power distribution systems may include distributed intelligent agents supporting advanced capabilities such as reactive and proactive power quality control for voltage regulation [6]. We wanted to explore mechanisms for enhancing intelligent agents by adding capabilities to autonomously create and conduct on-line power auctions. This required agents that could operate under the external guidance of multiple affiliated organizations and adapt their behavior to provide the additional functionality without compromising or impacting prior agent behaviors.

4.1 Equipping Agents to Conduct On-line Auctions

To implement the on-line double auctions, we chose an existing holonic MAS (HMAS) being used to evaluate power quality control algorithms for future intelligent power distribution systems [16]. The topology, shown in Figure 2 is based on the IEEE 37-bus feeder test case, with a sample data for a community of four neighborhoods, with four homes each with one of the four having distributed generation that could be made available for sale.

The power market organizations were arranged in a holonic manner, similar to the grid control options, but are subject to different behavior specifications and external stakeholders. We implemented a smaller, but highly parallel second holarchy to support our holonic power market simulation. The agents we chose employed the OBAA^{++} [4] architecture specifically designed for *multi-group agents* simultaneously participating in multiple independently-controlled organizations.

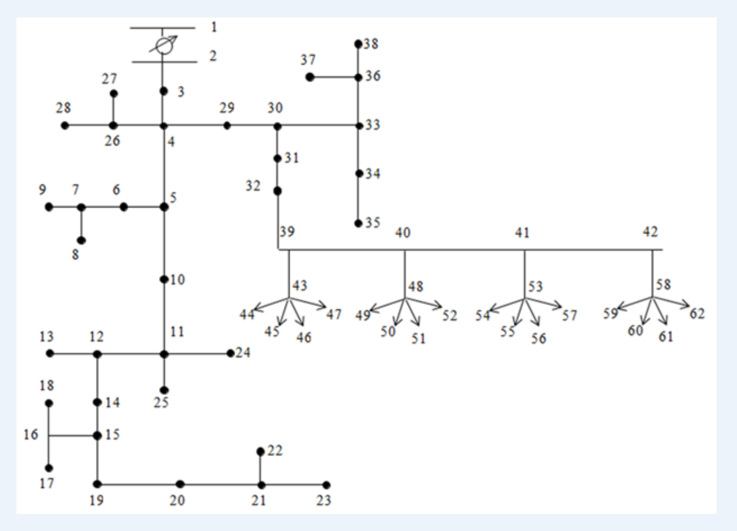


Fig. 2. Power distribution network topology (excerpt) for the double-auction test case.

OBAA^{++} agents are equipped with *capabilities* that provide specific functionality. The architecture includes an executable goal model for specifying organizational behaviors and defining the behavior goals for each of the local power market organizations. During execution of the system, suitably-equipped agents are dynamically assigned to specific roles that can achieve a particular organizational goal. Agents in our power market organizations can be assigned to only one of two roles. They either act as an *auction participant*, to achieve the goal we called *Auction Power*, or they act as the *auction broker*, accepting bid messages and executing the double auction for the participants to achieve the goal we called *Broker Power*. The necessary capabilities include typical group formation and administration abilities such as the ability to create authorized connections to affiliated agents (for example, an auction participant must be able to establish a secure line of communication with the local power market organization broker) and to register with the organization, essentially presenting the participants capabilities to the broker so it can get assigned roles to achieve the goals defined for the local power market organization.

Additional online power market-specific capabilities focus on the ability to prepare bids, send bid messages to the broker, or call the necessary analytical capabilities to execute or broker the auction and determine the degree to which each bid is satisfied. A list of the capabilities required for each role is shown in Figure 3 along with the goal that role can achieve to meet the overall objectives of the organization.

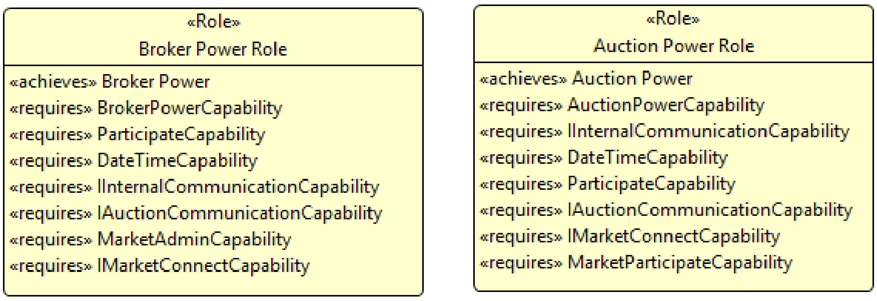

«Role»
Broker Power Role
«achieves» Broker Power
«requires» BrokerPowerCapability
«requires» ParticipateCapability
«requires» DateTimeCapability
«requires» IInternalCommunicationCapability
«requires» IAuctionCommunicationCapability
«requires» MarketAdminCapability
«requires» IMarketConnectCapability
«Role»
Auction Power Role
«achieves» Auction Power
«requires» AuctionPowerCapability
«requires» IInternalCommunicationCapability
«requires» DateTimeCapability
«requires» ParticipateCapability
«requires» IAuctionCommunicationCapability
«requires» IMarketConnectCapability
«requires» MarketParticipateCapability

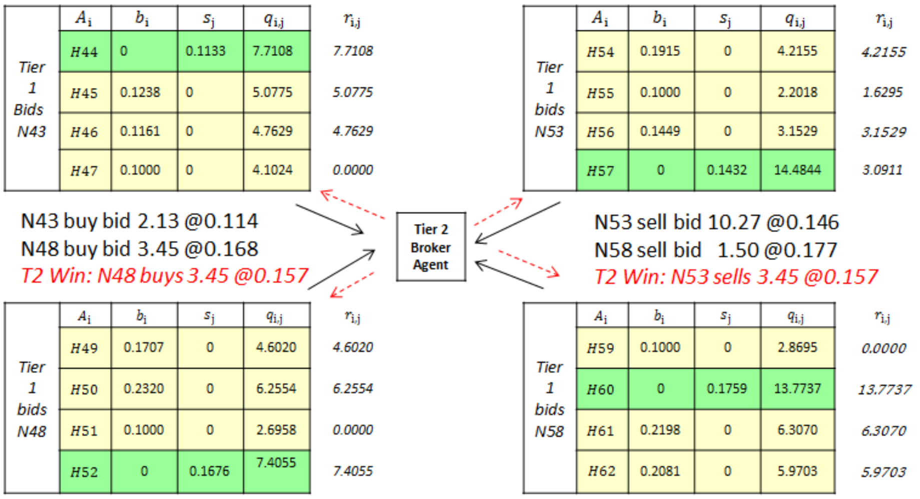
Fig. 3. Agents may be assigned to either *Auction Power* or to *Broker Power*, that is to either to bid in their local auction, or to resolve the auction by calling the double auction algorithm on a set of bids. Mid-level Neighborhood Transformer agents may *Broker Power* in a lower-level group and *Auction Power* in a higher-level group.

4.2 Exchanging Market Messages and Brokering Auctions

Each auction is conducted asynchronously in accordance with the specific guidelines provided. Guidelines include those specified for the power market organizations in which the online auctions will be conducted, as well as custom guidelines given to each multigroup agent that serve to direct the behavior of each agent in such a way that the agent could be customized to reflect the personal pricing strategies and comfort/profit motives of the owner. We expect some agents may be ultimately controlled by the homeowner, who makes the decision to sell power or not - and some agents may be wholly owned by the power company or power market agency, for example, those running along the later lines. Communication between agents was simulated using RabbitMQ, an implementation of the open-source Advanced Message Queuing Protocol (AMQP) [17].

5 Simulation

We tested the implementation in a *complex MAS*, a MAS consisting of of multiple local groups of intelligent agents working together in a multilevel holarchy. All local groups in the holarchy were fed from a single power power line, with a single lateral feeder agent, L39. The power line was assumed to supply four neighborhood transformers, each hosting one of the neighborhood transformer agents, N43, N48, N53, and N58. Each transformer supplied four homes, with one of the four homes providing mid-day power from rooftop PV panels. Each home was assumed to host a multigroup intelligent power distribution agent. The four agents associated with PV-enabled homes generated offers to sell power at a given future time to the other three homes in their neighborhood in the first-tier auction. The four neighborhood agents all received four bids from the supplying homes - one to sell power, and three offers to buy power. Upon receiving the bid messages from the four home agents, each neighborhood agent acted as a broker to execute the local auction. After executing the first-tier auctions, some bids were not completely fulfilled. The brokering agent determined the remaining quantity and forwarded the offer to the lateral agent for the second tier auction to be brokered by the lateral agent.



	A_i	b_i	s_j	$q_{i,j}$	$r_{i,j}$
Tier 1 Bids N43	H44	0	0.1133	7.7108	7.7108
	H45	0.1238	0	5.0775	5.0775
	H46	0.1161	0	4.7629	4.7629
	H47	0.1000	0	4.1024	0.0000

	A_i	b_i	s_j	$q_{i,j}$	$r_{i,j}$
Tier 1 bids N53	H54	0.1915	0	4.2155	4.2155
	H55	0.1000	0	2.2018	1.6295
	H56	0.1449	0	3.1529	3.1529
	H57	0	0.1432	14.4844	3.0911

N43 buy bid 2.13 @0.114
N48 buy bid 3.45 @0.168
T2 Win: N48 buys 3.45 @0.157

Tier 2 Broker Agent

N53 sell bid 10.27 @0.146
N58 sell bid 1.50 @0.177
T2 Win: N53 sells 3.45 @0.157

	A_i	b_i	s_j	$q_{i,j}$	$r_{i,j}$
Tier 1 bids N48	H49	0.1707	0	4.6020	4.6020
	H50	0.2320	0	6.2554	6.2554
	H51	0.1000	0	2.6958	0.0000
	H52	0	0.1676	7.4055	7.4055

	A_i	b_i	s_j	$q_{i,j}$	$r_{i,j}$
Tier 1 bids N58	H59	0.1000	0	2.8695	0.0000
	H60	0	0.1759	13.7737	13.7737
	H61	0.2198	0	6.3070	6.3070
	H62	0.2081	0	5.9703	5.9703

Fig. 4. Aggregate neighborhood results are forwarded to the lateral broker for secondary auction and the results of the Tier 2 auction are communicated back to the appropriate homes.

6 Experimental Results and Discussion

The bid information and auction clearing results for the initial four first-tier auction bids is shown in Figure 4. The associated home agents send their bid message to sell power at the given future time t_f1 to their associated neighborhood broker. The neighborhood brokers translated the message content information from each participant into a array of bids and bid information and used their double-auction computational capability to execute the auction. In addition to serving as brokers in the lower organizations, the neighborhood transformer agents also serve as auction participants in the higher-level second tier auction organization brokered by the agent running on the power line. The lateral power line broker agent then brokers a second-tier auction with the new secondary bids by again translating the message information into an input array and executing the secondary auction computational capability. The inputs and results of the second tier auction are also presented in Figure 4.

During simulation initialization, each auction agent connects their broker. The broker agent creates a local power market organization and issues goal assignments to itself (to broker) and to each participant (to participate). Given their assignments, the agents select an appropriate plan to play a role the achieves their assigned goal. If the current simulation time, $t_{now} < t_p$ where t_p is the future time at which the power will actually be exchanged, the agents will look at their personal goals and power availability to determine their quantity and unit price to buy or sell as q_k and c_k, respectively. When all bids have arrived, the broker executes the auction. The broker agent on neighborhood transformer N43, for example calculates a clearing price, $c_{0k} = 0.1139$ with an unsettled combined amount to sell, $q_{1k} = -2.1296$. A negative selling quantity means this neighborhood bid will be entered as a buy bid in the Tier 2 auction. When the broker on the lateral feeder line conducting the Tier 2 auction receives the aggregate neighborhood bids (BUY 2.13 at 0.11, BUY 3.45 at 0.17, SELL 1.5 at 0.18, and SELL 1.5 at 0.18 for N43, N48, N53, and N58 respectively), the Tier 2 broker executes the secondary double auction. In this example, N48 buys 3.45 at the 0.16 clearing price and N53 sells 3.45 at the same clearing price. N43 offered a lower buying price than N48 and bought nothing in the second round, while N58 wanted a higher selling price (compared to N53) and sold nothing as well. The successful Tier 2 neighborhood participants were N48 which bought all of the 3.45 requested, while N53 could only sell the matching 3.45/10.27 it had for sale. The two successful neighborhoods then cascade the results down to their participating homes as shown, with N43 and N58 homes not changing, the secondary buy for N48 completes the buy requests for homes H49 and H50, while the secondary sale for N53 gets distributed accordingly.

The simulation demonstrated the extensibility of the multigroup agents to support new multigroup organizations and behaviors concurrently. Agents continued to perform voltage control for a 5-tier grid control hierarchy [16] while implementing the new online power auctioning behaviors. Capabilities and messaging protocols were independently configured using the recommended process [3]. Changes to the desired market behaviors have minimal impacts on the previously existing functionality, and specifications for the behavior of the power market behaviors remain unaffected by the modifications to the prior functionality (related to managing voltage fluctuations). Therefore, in addition to testing the distributed implementation of a double auction, the results showed the ability of the multigroup agents to successfully create and participate in new organizations, implement new and independent goal-driven behavior specifications, and successfully manage the addition of new capabilities to support the new objectives.

7 Conclusions and Recommendations

The project demonstrated a mechanism for enhancing distributed intelligent agents supporting future power distribution systems to initiate, participate, and broker autonomous online power auctions employing a two-tier double auction mechanism. Additional work will focus on evaluating additional iterative, distance-adjusted auction alogorithms and the introduction of additional mechanisms for adapting behavior due to communication unreliability and delays, agents entering and leaving the local auctions, and evaluating responses to attempts to manipulate the market based on known (or learned) effects of agent-assisted pricing mechanisms.

Acknowledgments. This work was supported by the US National Science Foundation via Award No. CNS-1136040. The views expressed in this paper are those of the authors.

References

1. Abdelnasser, A., Hossain, E., Kim, D.: Clustering and resource allocation for dense femtocells in a two-tier cellular ofdma network (2014)
2. Bazilian, M., Onyeji, I., Liebreich, M., MacGill, I., Chase, J., Shah, J., Gielen, D., Arent, D., Landfear, D., Zhengrong, S.: Re-considering the economics of photovoltaic power. Renewable Energy 53, 329–338 (2013)
3. Case, D., DeLoach, S.: Applying an o-mase compliant process to develop a holonic multiagent system for the evaluation of intelligent power distribution systems. In: Engineering Multi-Agent Systems, pp. 78–96. Springer (2013)
4. Case, D.M., DeLoach, S.A.: Obaa++: an agent architecture for participating in multiple groups. In: Proceedings of the 2014 Intl conference on Autonomous agents and multi-agent systems. pp. 1367–1368. Intl Foundation for Autonomous Agents and Multiagent Systems (2014)
5. Chen, H., Lau, H.C.: Decentralized resource allocation and scheduling via walrasian auctions with negotiable agents. In: Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM Intl Conference on. vol. 2, pp. 356–360. IEEE (2010)
6. Deshmukh, S., Natarajan, B., Pahwa, A.: Voltage/var control in distribution networks via reactive power injection through distributed generators. Smart Grid, IEEE Transactions on 3(3), 1226–1234 (2012)
7. Fang, X., Misra, S., Xue, G., Yang, D.: Smart gridthe new and improved power grid: A survey. Communications Surveys & Tutorials, IEEE 14(4), 944–980 (2012)
8. Faqiry, Kunduand, Mukherjeeand, Das, Pahwa: Game theoretic model of energy trading strategies at equilibrium in microgrids. NAPS 2014 29 (2014)
9. Feng, X., Chen, Y., Zhang, J., Zhang, Q., Li, B.: Tahes: A truthful double auction mechanism for heterogeneous spectrums. Wireless Communications, IEEE Transactions on 11(11), 4038–4047 (2012)
10. HomChaudhuri, B., Kumar, M., Devabhaktuni, V.: Market based approach for solving optimal power flow problem in smart grid. In: American Control Conference (ACC), 2012. pp. 3095–3100. IEEE (2012)
11. Huang, P., Scheller-Wolf, A., Sycara, K.: Design of a multi–unit double auction e–market. Computational Intelligence 18(4), 596–617 (2002)
12. Koestler, A.: The ghost in the machine (1989)
13. Le, T., Beluri, M., Freda, M., Gauvreau, J.L., Laughlin, S., Ojanen, P.: On a new incentive and market based framework for multi-tier shared spectrum access systems. In: Dynamic Spectrum Access Networks (DYSPAN), 2014 IEEE Intl Symposium on. pp. 477–488. IEEE (2014)

14. Maes, P., Guttman, R.H., Moukas, A.G.: Agents that buy and sell. Communications of the ACM 42(3), 81–ff (1999)
15. Malekpour, A.R., Niknam, T.: A probabilistic multi-objective daily volt/var control at distribution networks including renewable energy sources. Energy 36(5) (2011)
16. Malekpour, A.R., Pahwa, A., Das, S.: Inverter-based var control in low voltage distribution systems with rooftop solar pv. In: North American Power Symposium (NAPS), 2013. pp. 1–5. IEEE (2013)
17. O'Hara, J.: Toward a commodity enterprise middleware. Queue 5(4), 48–55 (2007)
18. Pahwa, A., DeLoach, S.A., Das, S., Natarajan, B., Ou, X., Andresen, D., Schulz, N., Singh, G.: Holonic multi-agent control of power distribution systems of the future. CIGRE Grid of the FutureSymposium (2012)
19. Ramchurn, S.D., Vytelingum, P., Rogers, A., Jennings, N.R.: Putting the'smarts' into the smart grid: a grand challenge for artificial intelligence. Communications of the ACM 55(4), 86–97 (2012)
20. Shoham, Y., Leyton-Brown, K.: Multiagent systems: Algorithmic, game-theoretic, and logical foundations. Cambridge University Press (2008)
21. Wang, S., Xu, P., Xu, X., Tang, S., Li, X., Liu, X.: Toda: truthful online double auction for spectrum allocation in wireless networks. In: New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on. pp. 1–10. IEEE (2010)
22. Wang, Y., Saad, W., Han, Z., Poor, H.V., Basar, T.: A game-theoretic approach to energy trading in the smart grid. Smart Grid, IEEE Transactions on 5(3), 1439–1450 (2014)
23. Weng, C., Lu, X., Xue, G., Deng, Q., Li, M.: A double auction mechanism for resource allocation on grid computing systems. In: Grid and Cooperative Computing-GCC 2004, pp. 269–276. Springer (2004)
24. Wiginton, L., Nguyen, H., Pearce, J.M.: Quantifying rooftop solar photovoltaic potential for regional renewable energy policy. Computers, Environment and Urban Systems 34(4), 345–357 (2010)
25. Wilson, R.: Architecture of power markets. Econometrica 70(4), 1299–1340 (2002)
26. Zhou, X., Cai, Y., Edward Chow, C., Augusteijn, M.: Two-tier resource allocation for slowdown differentiation on server clusters. In: Parallel Processing, 2005. ICPP 2005. International Conference on. pp. 31–38. IEEE (2005)

Nutritional Menu Planning: A Hybrid Approach and Preliminary Tests

Oscar Chávez-Bosquez[1], Jerusa Marchi[2], and Pilar Pozos-Parra[1]

[1] Universidad Juárez Autónoma de Tabasco, División Académica de Informática y Sistemas. Tabasco,
Mexico

[2] Universidade Federal de Santa Catarina, SC,
Brazil

{oscar.chavez, pilar.pozos}@ujat.mx,jerusa.marchi@ufsc.br

Abstract. Menu planning is a process appearing to be straightforward but many complexities arise when it is tried to be solved by computer means. Actually, although there is evidence of previous work since 50 years ago, at present there is no wide know tool which can solve this task in an automated manner. Also, not all proposals deal with full recipes along with considering the user food preferences. In this paper we propose a system architecture based on hybrid optimization: a first module based on mathematical programming, a well known robust approach to this problem; and a second module based on belief merging, a lesser known framework aimed to combine the nutrition scientist advices and policies along with the user food desires. The association of numerical and symbolic approaches will allow us to generate of a more agreeable menu. In order to illustrate our proposal, we present a motivating example detailing the main aspects of the system.

Keywords: Diet planning, menu planning, knowledge representation, belief merging, laws of nutrition.

1 Introduction

A good nutrition is the one that provides to the body the correct amount of energy and the necessary nutrients to maintain the vital functions of an individual and to perform his/her daily activities. Human beings ingest energy and nutrients in the form of a diet consisting of different foods. Dietitians and nutritionist are the scientists who plan food and meal preparations, promote healthy eating habits and help patients with specific nutrition needs to get the essential nutrients for them.

As there is no instruction guide with the steps to generate menus, nor a generic accepted criteria to classify a produced menu as good or bad, we propose to build a model based on the called "Laws of Nutrition" or "Laws of Feeding":

Law of Quantity, Law of Quality, Law of Harmony, Law of Adequacy [7]. This set of rules were proposed by Pedro Escudero, which is considered the "father of nutrition" in Latin America [14,23,17].

We enforce two of the four laws in our proposal:

Law of Quantity. Individual's food intake must supply the amount of calories his/her body needs. This value is commonly obtained via anthropometric measurements of the user, mainly gender, age, weight and height; in addition to physical activity. For flexibility, a diet can supply ± 10% of the user's caloric requirement.

Law of Adequacy. The diet must be adequate to each user according not only to his nutritional requirements, but also to his both social and physiological needs. This means that the diet must consider intolerance and preference of aliments. This Law is focused on the final user. For example, it can set restrictions such as cultural manners, climate topics, or geographical issues.

Due to the above restrictions, planning an adequate nutritious menu for individuals and groups is a complex task. There have been multiple attempts to solve it using mathematical algorithms via computational ways since the decade of 60s [1,8]. Since then, there have intensive research in the field, from solutions ranging from mathematical programming ([15,26]) to expert systems ([22,9]) to evolutionary computing and metaheuristics [10,13].

Most of these solutions work mainly with numerical values of the food: amount of calories, carbohydrates, protein and lipids; some proposals consider the food cost. However, the user choices, preferences, acquired tastes or dietary habits usually are not taken into account, since the computational solutions are implemented from the point of view of the nutrition scientist, for whom the goal is mainly to prescribe healthy foods.

Therefore, we propose a hybrid approach to take into account the user desires and preferences in order to generate a menu more according to the user's point of view, and consequently more acceptable by the user. This inclusion will be done via Belief Merging, a framework used for the integration of knowledge from different sources modeled as propositional formulae.

Finally, some of the approaches work only with basic food items, letting to the nutrition scientist the responsibility of choose the items forming the recipes. In our proposal, we work with basic food items as well with full recipes in order to give a personalized diet considering the user preferences.

2 Belief Merging

Belief merging aims at combining several pieces of (possibly inconsistent) information coming from different sources [12]. The goal is to produce a single consistent set of information, trying to keep the most of the information of the sources.

Belief merging is an important issue in artificial intelligence and databases, and its applications are many and diverse [4]. For example, in multiagent systems a merging operator defines the beliefs of a group of agents according to the beliefs of each member of the group. When agents have conflicting beliefs about the "true" state of the world, belief merging can be used to determine the "true" state of the world for the group [24].

A belief merging operator is the responsible for making the belief merging. Several merging operators have been defined and characterized in a logical way. Among them, *PS-Merge* is a versatile operator which can be used to real-world problems, although only demonstrated solving "toy examples" due to its exponential complexity [6].

Preliminary concepts. In this theory we consider a language $\mathcal{L}(P)$ of propositional logic using a finite ordered set of symbols $P = \{p_1, p_2, ..., p_n\}$ where the formulae are in Disjunctive Normal Form (DNF). A formula υ is in DNF iff υ is a disjunction of terms $\upsilon =D_1 \vee ... \vee D_m$, where each term D_i is a conjunction of literals $D_i = l_1 \wedge ... \wedge l_k$, with $l_i = p_j$ or $l_i = \neg p_j$.

A *belief base* or *knowledge base* (KB) K is a finite set of propositional formulae of $\mathcal{L}$ representing the beliefs from a source. We identify K with the conjunction of its elements.

An *interpretation*, or *state*, or *world* is a function w from P to $\{1, 0\}$. These values are identified with the classical truth values *true* and *false* respectively. The set of all possible interpretations will be denoted as $\mathcal{W}$, its elements are denoted by vectors of the form $(w(p_1), ..., w(p_n))$. A model of a propositional formula υ is an interpretation such that $w(Q) = 1$ once w is extended in the usual way over the connectives, and the set of models of a formula υ will be denoted by $mod(\upsilon)$. K is consistent if there exists a model of K.

If υ is a propositional formula or a set of propositional formulae then $\mathcal{P}(\upsilon)$ denotes the set of atoms appearing in υ. $|P|$ denotes the cardinality of set P. A *literal* is an atom or its negation.

A belief *profile* $E = \{K_1, ..., K_m\}$ is a multiset (bag with possible repeated elements) of m belief bases.

***PS-Merge* Operator.** This belief merging operator is an alternative way to distance-based operators [11,16,18,25] or to syntax-based operators [2,19], as it is based on the notion of Partial Satisfiability [5]. *PS-Merge* produces similar results to other merging approaches, but while other approaches require many merging operators in order to achieve satisfactory results for different scenarios, this approach obtains similar results for all these different scenarios with a unique operator [24].

Let $E = \{K_1, ..., K_m\}$ be a belief profile and *PS-Merge* be a function which maps a belief profile to a belief base, *PS-Merge* : $\mathcal{L}(P)^n \rightarrow \mathcal{L}(P)$, then the Partial Satisfiability Merge of E is $PS\text{-}Merge(E)$ such that the set of models of the resulting base is:

$$\left\{ w \in \mathcal{W} \;\middle|\; \sum_{i=1}^{m} w_{ps}(K_i) \geq \sum_{i=1}^{m} w'_{ps}(K_i) \text{ for all } w' \in \mathcal{W} \right\} \tag{1}$$

Let K be a belief base, w any interpretation of $\mathcal{W}$ and $|P| = n$, we define the Partial Satisfiability of K for w, denoted as $w_{ps}(Q_K)$, as follows:

- if $Q_K := C_1 \wedge \cdots \wedge C_s$ where C_i are literals then:

$$w_{ps}(Q_K) = \max \left\{ \sum_{i=1}^{s} \frac{w(C_i)}{s}, \frac{n - |\mathcal{P}\left(\bigwedge_{i=1}^{s} C_i\right)|}{2n} \right\} \tag{2}$$

- if $Q_K := D_1 \vee \cdots \vee D_r$ where each D_i is a literal or a conjunction of literals then:

$$w_{ps}(Q_K) = \max \{w_{ps}(D_1), \ldots, w_{ps}(D_r)\} \tag{3}$$

The intuitive explanation of Partial Satisfiability is as follows [5]: it is natural to think that if we have the conjunction of two literals and just one is satisfied then we are satisfying 50 % of the conjunction. If we generalize this idea we can measure the satisfaction of a conjunction of one or more literals as the sum of their evaluation under the interpretation divided by the number of conjuncts. However, the agent may consider only some atoms of the language, then the agent is not affected by the decision taken over the atoms that are not considered. So, in this case we give a partial satisfaction of 50 % for each atom not appearing in the agent's beliefs. On the other hand, the agent is interested in satisfying the literals that appear in its beliefs, and we interpret this fact by assigning a satisfaction of 100 % to each literal verified by the interpretation and 0 % to those that are falsified.

3 Proposal

We present a hybrid approach to solve the menu planning problem, by means of the system architecture shown in Figure 1. Although significant data preprocessing is done, we divide the overall system functionality in two algorithms, each one performing one main task of the process. The system uses different inputs according to each step, whereas outputs are menus at different level of *adequation*: some menu could have a high level of adequacy for a nutritionist (it meets the user's caloric requirement), but a poor level of adequacy for the final user (it has no favourite foods included).

3.1 Phase 1

First, system takes the user caloric needs and the database with all the available foods to generate a first approximation to a menu in accordance with the energy needed by the user. It is worth noticing that this approximation to a menu may correspond to a possibly incorrect diet according to the Law of Adequacy, that's why we name this result as *list of foods*. Each list of foods contains no element of the user's food intolerance set.

The proposed menu is divided into meals in a day. User specifies the amount of meals he wants to take in a day, and he can also specify which meals are most important by allocating a percentage of intake per meal. System generates 3 lists of foods per meal according to the user caloric needs: the one closer to the needs -10%, another one closer to the exact value of the user needs, and the closer to the needs $+10\%$.

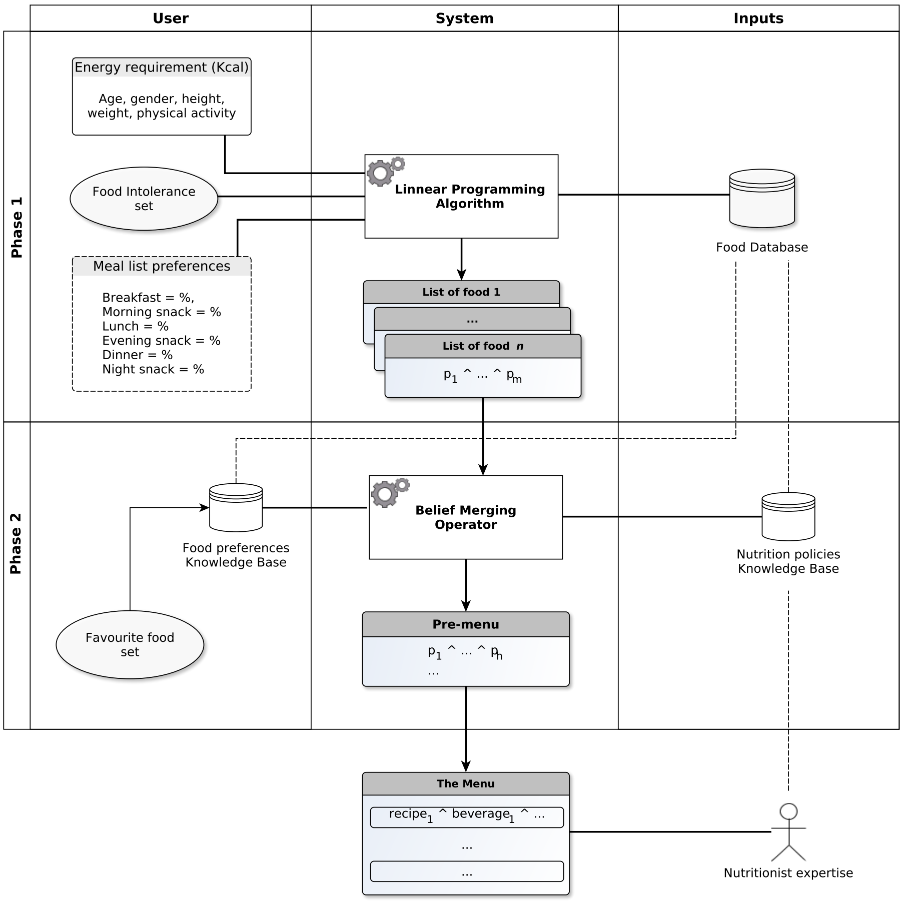


Fig. 1. System architecture. The system (center column) is composed of two algorithms, one for each phase. Input consists of data from the user (first column) and predefined data structures (third column). The final result is an adequate menu.

The main algorithm in this step generates each list of foods via Linear Programming, inspired in the model presented in [21]. Thus we had to translate the problem into a mathematical model. The result of this stage is in knowledge base format, so we have to translate each food to its corresponding variable.

3.2 Phase 2

At this step, the system tries to meet the user preferences by means of a belief merging operator, since all inputs are symbolic data coming from different sources equally respectable but most likely inconsistent.

We start by translating the user food preferences into a knowledge base format. This KB is merged with the lists of food generated in the previous phase and with a KB of nutritional polices, i.e. "common sense" information (for example the *adequate* combination of foods). The last KB will be filled up by the nutrition scientist knowledge, but it can contain other kind of information. The main algorithm performing this step is the PS-Merge belief merging operator [24]. The output is expected to be a consistent KB per list of food.

In order to show a human-readable menu, the resulting KB is translated from propositional variables to foods names, and the final configuration is left to the specialist. This resulting menu meets the user's caloric requirement, as well as it complies with the two Laws of Nutrition considered.

3.3 Food Database

Figure 2 shows the relational model of the food database proposed. Main tables are `food` and `patient`, whose main relations are about the user's food intolerance set (`patient_intolerant_food`), user favourite food (`patient_favourite_food`), and the final menu for the day (`menu`).

4 A Motivating Example

Now we will show a simple instance that illustrates the hybrid approach of our proposal. We will describe the entire process by generating a breakfast, one single meal from a complete menu. Each element of the system architecture will be described in terms of the example instance, emphasizing in the belief merging process, the main contribution of this work.

4.1 User Data

Energy Requirement. We need to plan a menu for a 33 year old male, weight 84 kg, 186 cm tall, and an intense physical activity. To obtain his caloric requirement we have to calculate the Basal Metabolic Rate (BMR), which is the minimum amount of energy that the body needs without realizing any activity using the following formula [3]:

$\mathrm{BMR}_{\mathrm{men}} = (10 \times \text{weight}) + (6.25 \times \text{height}) - (5 \times \text{age}) + 5 = 1680\ \text{kcal}$

Afterwards, we calculate the Total Energy Expenditure (TEE) representing the amount of energy needed to maintain the vital functions of the body and

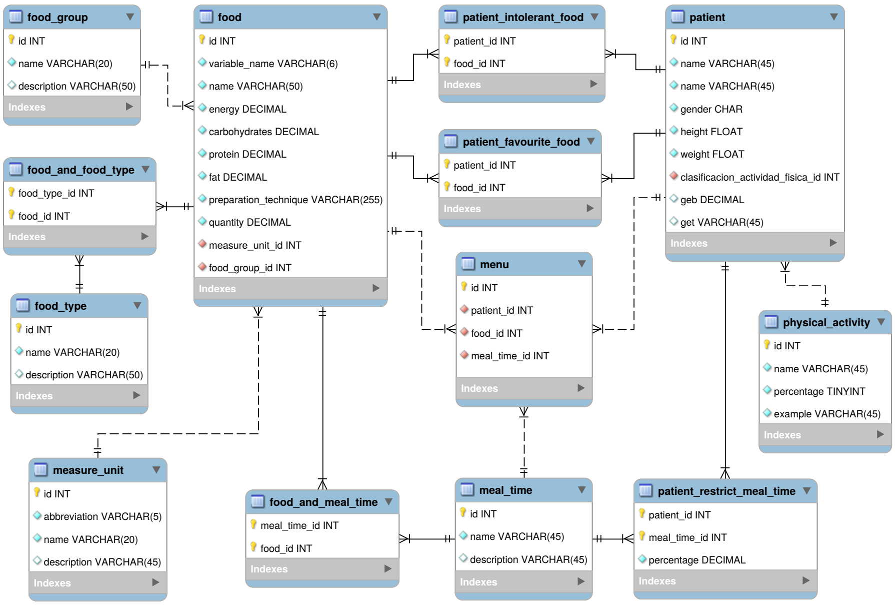


Fig. 2. Foods database relational diagram.

perform the daily activities. This is obtained from the BMR plus two values: adding a 10 % due to the digestive effect (Thermic Effect of the Food, TEF), and adding a percentage depending the user's physical activity (Physical Activity Factor, PAF) according to Repose = 20%, Sedentary = 37.5%, Moderate = 55%, Intense = 72.5% [3]. So the TEE for this patient is as follows:

$$\text{TEE} = \text{BMR} + \text{TEF} + \text{PAF} = 1680 + 168 + 1218 = 3066 \text{ kcal}$$

Food Intolerance. The user claims to be lactose intolerant.

Menu Distribution. The individual can only take three meals per day, so we must distribute all his food intake into three meals, according to the following distribution:

Meal	Percentage	Energy
Breakfast	30 %	920 kcal
Lunch	40 %	1226 kcal
Dinner	30 %	920 kcal

Favourite Foods. Apples and bread in the breakfast, and he always eats pepper sauce in each meal. He also dislikes papaya fruit. According to foods shown in Table 1, these beliefs can be modeled in the Food Preferences KB as:

$$K_1 = (\ a \wedge b \wedge m \wedge \neg p\ ) \tag{4}$$

4.2 Input Data Structures

Food Database. Table 1 considers the foods from the database related to breakfast. Data were obtained from a local nutrition lab. For the sake of space, we only shown the relevant information related to this example.

Table 1. Subset of breakfast foods in database.

Var.	Name	Energy	Group	Type	Time	Meal	Unit	Qty.
a	Apple	70	F	F	1,2,3	B,S,D	Piece	1
n	Banana	96	F	F	1,2,3	B,S,D	Piece	1
p	Papaya	43	F	F	1,2	B,S,D	gr	200
s	Eggs with ham	310	P	E	4	B	Portion	1
u	Lima beans soup	328	P	C	4	B,L	Portion	1
e	Eggs with beans	446	P	E,L	4	B,L	Portion	1
h	Hot cakes	362.5	C	C	4	B	Portion	1
i	Cappuccino coffee	157	C	C	5	B,D	Cup	1
c	Caffè Americano	21	C	C	5	B,D	Cup	1
o	Orange juice	197.5	F	F	5	B	Glass	1
k	Hot milk	116	P	K	5	B,S	Glass	1
b	Bread roll	301	C	C	2	B,L,D	Piece	1
t	Tortilla	246	C	C	2	B,L,D	Piece	1
g	Green salad	79.5	V	V	4	B,S,D	Portion	1
l	Accompanying salad	26	V	V	2	B,L,D	Portion	1
m	Tomato spicy sauce	45	V	V	2	B,L,D	ml	50

Table 1 uses the following acronyms:

- **Type:** C = Cereal, D = Dairy, E = Egg, F = Fruit, L = Legumes, M = Meat, V = Vegetables.
- **Group:** C = Cereal, F= Fruit, P = Protein, V = Vegetables.
- **Time:** 1 = Entrance, 2 = Accompanying, 3 = Dessert, 4 = Main dish, 5 = Drink.
- **Meal:** B = Breakfast, S = Snack, L = Lunch, D = Dinner.

Nutrition Policies KB. For this case, we consider the following expert advises: *(i)* not to eat bread and tortilla in the same meal, because both are high carbohydrates foods; and *(ii)* it is recommended to eat apple, orange or papaya in breakfast due to their digestive properties. With the foods shown in Table 1, these two beliefs can be modeled in the Nutrition Policies KB as:

$$K_2 = (\ \neg(b \wedge t) \wedge (a \vee o \vee p)\ ) \tag{5}$$

4.3 Phase 1

In this step we employ Linear Programming to generate the lists of foods. First we get rid of the intolerance foods using a subset from Table 1 without food from dairy type (no hot milk, cappuccino coffee nor hot cakes in this case). Within this subset we find three lists of foods whose energy complies the breakfast's energy required by the user (920 kcal):

List of foods 1	
Apple	70
Eggs with beans	446
Caffè Americano	21
Bread roll	301
Total energy:	838

List of foods 2	
Papaya	43
Apple	70
Eggs with ham	310
Orange juice	197.5
Bread roll	301
Total energy:	921.5

List of foods 3	
Green salad	79.5
Apple	70
Lima beans soup	348
Orange juice	197.5
Bread roll	301
Total energy:	996

Energy of the first list of foods is under the breakfast's exact energy requirement, but over the -10%. Energy of the second list of foods is close to the exact breakfast's energy requirement. Energy of the third list of foods is above the breakfast's energy requiriement but lower than the $+10\%$.

4.4 Phase 2

In this step we employ Belief Merging to generate menus. Translating list of foods 1 as KB:

$$K_3 = (a \wedge e \wedge i \wedge b) \tag{6}$$

Gives us the belief profile to be merged:

$$E = (K_1, K_2, K_3) \tag{7}$$

PS-Merge operator requires normalizing each KB to DNF:

$$Q_{K_1} = (a \wedge b \wedge m \wedge \neg p) \tag{8}$$
$$Q_{K_2} = (\neg b \wedge a) \vee (\neg b \wedge o) \vee (\neg b \wedge p) \vee (\neg t \wedge a) \vee (\neg t \wedge o) \vee (\neg t \wedge p) \tag{9}$$
$$Q_{K_3} = (a \wedge e \wedge i \wedge b) \tag{10}$$

PS-Merge operator is used to find KB with no inconsistencies and with most of the food preferences and nutrition policies satisfied. For example, taking Q_{K_2} and the interpretation $w' = (1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 0,\ 0)$ corresponding to the propositional values $\big(w(a), w(b), w(e), w(i), w(m), w(o), w(p), w(t)\big)$ in that order, we have:

$$\begin{aligned} w'_{ps}(Q_{K_2}) = \max \big\{ & w'_{ps}(\neg b \wedge a),\ w'_{ps}(\neg b \wedge o),\ w'_{ps}(\neg b \wedge p),\ w'_{ps}(\neg t \wedge a), \\ & w'_{ps}(\neg t \wedge o),\ w'_{ps}(\neg t \wedge p) \big\} \end{aligned}$$

Obtaining the partial satisfiability of the first element:

$$w'_{ps}(\neg b \wedge a) = \max\left\{\frac{w'(\neg b) + w(a)}{2}, \frac{8 - |\mathcal{P}(\neg b, a)|}{2(8)}\right\} = \max\left\{\frac{0+1}{2}, \frac{8-2}{16}\right\}$$
$$= \max\left\{\frac{1}{2}, \frac{3}{8}\right\} = \frac{1}{2}$$

Computing the partial satisfiability of the remaining elements we can obtain the partial satisfiability of the KB with respect to w':

$$w'_{ps}(Q_{K_2}) = \max\left\{\frac{1}{2}, \frac{1}{2}, \frac{3}{8}, 1, 1, \frac{1}{2}\right\} = 1$$

For the sake of space and clarity, below we only show the interpretations corresponding to the best partial satisfiability. Both w_1 and w_2 contain the views of the majority, representing the best possible menu according to the input KBs:

Belief merging result via PS-Merge

	a	b	e	i	m	o	p	t	$w_{ps}(Q_{K_1})$	$w_{ps}(Q_{K_2})$	$w_{ps}(Q_{K_3})$	Sum
w_1	1	1	1	1	1	1	0	0	1	1	1	3.0
w_2	1	1	1	1	1	0	0	0	1	1	1	3.0

The final process of this stage is to translate these KBs to menus, and then verify if the menu is still consistent according to the energy requirement of the user:

Menu 1

Apple	70
Eggs with beans	446
Caffè Americano	21
Bread roll	301
Tomato spicy sauce	45
Orange juice	197.5
Total energy:	1080.5

Menu 2

Apple	70
Eggs with beans	446
Caffè Americano	21
Bread roll	301
Tomato spicy sauce	45
Total energy:	883

We can notice that energy in Menu 1 is above the user's needs, so this menu must be discarded. On the other hand, Menu 2 has a right amount of energy, as well as satisfies the user's food preferences and also complies with the nutrition policies.

Lastly, the adequate menu, or maybe a set of menus, are presented to the nutrition scientist, who will choose or modify the proposals to finally prescribe the diet to the user.

5 Conclusions

Diet planning is considered as an art, but we consider the need of a set of strict rules to give a reference of the "correctness" of a menu. In order to produce adequate menus, we use the "Laws of Nutrition" as a formal reference to validate our results. So, our proposal enforces the Law of Quantity as well as the Law of Adequacy, the latter with emphasizing the user preferences.

We have presented a novel approach for dealing with the menu planning problem. The proposed architecture is based on two formal frameworks used in two phases to solve the different views of the diet planning problem.

In the first step, the Linear Programming method has been proven to solve the problem properly and on time. For this purpose, we focus on build a mathematical model containing the quantitative information of the problem. In this way we obtain menus in accordance with the user's nutritional needs.

In the second step, we implement a Belief Merging operator, a relatively new framework which few applications to real-world problems are known (i.e. [20]). For this purpose, we focus on build a logical model containing the qualitative information of the problem. In this way we obtain menus in accordance with the user's food desires. PS-Merge belief merging operator provides good results, as it tries to satisfy the majority opinion and so minimize the level of total dissatisfaction.

We have shown that this hybrid proposal can be a viable approach to solve the problem at a very specific level. Further research is required to provide evidence of the effectiveness and feasibility of this proposal to complete menus with bigger input data.

References

1. Balintfy, J.L.: Menu planning by computer. Communications of the ACM 51(6), 255–259 (1964)
2. Baral, C., Kraus, S., Minker, J., Subrahmanian, V.: Combining knowledge bases consisting of first order theories. In: Ras, Z., Zemankova, M. (eds.) Methodologies for Intelligent Systems, Lecture Notes in Computer Science, vol. 542, pp. 92–101. Springer Berlin Heidelberg (1991)
3. Berman, R.: Boosting Your Metabolism For Dummies. EBL ebooks online, Wiley (2013)
4. Bloch, I., Hunter, A.: Fusion: General concepts and characteristics. International Journal of Intelligent Systems 10(16), 1107–1134 (2001)
5. Borja Macías, V., Pozos Parra, P.: Model-based belief merging without distance measures. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems. pp. 154:1–154:3. AAMAS '07, ACM, New York, NY, USA (2007)
6. Borja Macías, V., Pozos Parra, P.: Implementing PS-Merge Operator. In: Hernández Aguirre, A., Monroy Borja, R., Reyes García, C. (eds.) MICAI 2009: Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 5845, pp. 39–50. Springer Berlin Heidelberg (2009)
7. Deon, B.C., Medeiros, L.B., de Freitas Saccol, A.L., Hecktheuer, L.H., Saccol, S., Naissinger, M.: Good food preparation practices in households: A review. Trends in Food Science & Technology (In press) (2014)
8. Eckstein, E.F.: Menu planning by computer: The random approach. Journal of the American Dietetic Association 51(6), 529–533 (1967)

9. Ejtahed, H.S., Sarsharzadeh, M.M., Mirmiran, P., Asghari, G., Yuzbashian, E., Azizi, F.: Leemoo, a dietary assessment and nutritional planning software, using fuzzy logic. International Journal of Endocrinology Metabolism 11(4), e10169 (2013)
10. Gaál, B., Vassányi, I., Kozmann, G.: A novel artificial intelligence method for weekly dietary menu planning. Methods of Information in Medicine 44(5), 655–664 (2005)
11. Konieczny, S., Pino Pérez, R.: On the logic of merging. In: Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98). pp. 488–498 (1998)
12. Konieczny, S., Pino Pérez, R.: Logic based merging. Journal of Philosophical Logic 40(2), 239–270 (2011)
13. Koroušić Seljak, B.: Computer-based dietary menu planning. Journal of Food Composition and Analysis 22(5), 414–420 (2009), 7th International Food Data Conference: Food Composition and Biodiversity
14. Kott, S., Droux, J.: Globalizing Social Rights: The International Labour Organization and Beyond. International Labour Organization (ILO) Century Series, Palgrave Macmillan (2013)
15. Leung, P., Wanitprapha, K., Quinn, L.: A recipe-based, diet-planning modelling system. British Journal of Nutrition 74, 151–162 (8 1995)
16. Liberatore, P., Schaerf, M.: Arbitration (or how to merge knowledge bases). IEEE Transactions on Knowledge and Data Engineering 10(1), 76–90 (1998)
17. Library, U.H.S.: Pedro Escudero. `http://hsl.lib.unc.edu/specialcollections/bios/escudero` (2013)
18. Lin, J., Mendelzon, A.O.: Knowledge Base Merging by Majority. In: Pareschi, R., Fronhoefer, B. (eds.) Dynamic Worlds: From the Frame Problem to Knowledge Management. Kluwer (1999)
19. Marchi, J., Bittencourt, G., Perrussel, L.: Prime forms and minimal change in propositional belief bases. Annals of Mathematics and Artificial Intelligence 59(1), 1–45 (2010)
20. McAreavey, K., Liu, W., Miller, P., Meenan, C.: Tools for finding inconsistencies in real-world logic-based systems. In: STAIRS. pp. 192–203 (2012)
21. Papadimitriu, C.: Computational complexity. Addison-Wesley, Reading,Massachusetts (1994)
22. Petot, G.J., Marling, C., Sterling, L.: An artificial intelligence system for computer-assisted menu planning. Journal of the American Dietetic Association 98(9), 1009–1014 (1998)
23. Pite, R.: Creating a Common Table in 20th-century Argentina. University of North Carolina Press (2013)
24. Pozos Parra, P., Borja Macías, V.: Partial satisfiability-based merging. In: Gelbukh, A., Kuri Morales, Ángel Fernando. (eds.) MICAI 2007: Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 4827, pp. 225–235. Springer Berlin Heidelberg (2007)
25. Revesz, P.: On the semantics of arbitration. International Journal of Algebra and Computation 7(02), 133–160 (1997)
26. Sklan, D., Dariel, I.: Diet planning for humans using mixed-integer linear programming. British Journal of Nutrition 70 (1993)

A Comparative Study of Type 1 Singleton Fuzzy Logic Systems in Machining Application

Pascual Noradino Montes Dorantes[1], Marco Aurelio Jiménez Gómez[1],
Xavier Cantú Rodriguez[2], and Gerardo Maximiliano Méndez[3]

[1] Departamento de Ingeniería Industrial, División de estudios de posgrado e investigación,
Instituto Tecnológico de Ciudad Victoria, Tamaulipas, Mexico

[2] Universidad del Valle de México (Campus Monterrey), Nuevo León, Mexico

[3] Posgrado en Ingeniería Mecatrónica Instituto Tecnológico de Nuevo León,
Nuevo León, Mexico

pascualresearch@gmail.com

Abstract. This paper presents a comparison of two different techniques of Soft Computing (SC), in union with a novel approach of a hybrid model with Central Composite Design (CCD) with capacity of performing a rule reduction, without loss of reliability, and also performs a reduction of the computational time spent in prediction. Results obtained with this method show better performance thought generating a minimum error rate when it is compared with the Adaptive Network Fuzzy Inference Systems (ANFIS), or with the least squares method in case of pure the Fuzzy Singleton system.

Keywords: Fuzzy logic, hybrid model, ANFIS, CCD, T1SFLS.

1 Introduction

Dynamics of modern manufacturing processes require systems that can be fast, reliable and safe for products and persons that operating them. The science and technology developed in XX century provide some tools in the form of intelligent systems. The Soft Computing (SC) has proven to be reliable in industrial applications, but modern systems need to deal with a lot of information provided by multiple sources. Also, they need to allocate the resources in a safe, reliable and different manner [1]. There are many real word examples in which some data is missing. In these cases the mathematical models such as Least Squares (LS) are useful in obtaining an output response for the function that represents the systems. This prediction and the historical data serve as inputs for the mathematical model that serves as a database in modeling the expert system. These classes of new systems must be able to adapt in fulfillment of the requirements, but these needs, new models capable to manage: knowledge and reasoning such as humans do. Fuzzy Inference Systems (FIS) can satisfy these requirements and can be modeled in a combination with other SC techniques to create or simplify these models. e.g., the combination of neural networks (NN) with Fuzzy Logic Systems (FLS) creates the ANFIS model.

The aim of this paper is test the SC techniques of fuzzy systems in their version singleton in the form of pure Type 1 Singleton fuzzy logic system (T1SFLS) and hybrid systems (ANFIS) and evaluate their performances to find a model capable to work on a real time and to provide a simplification of the system itself. This is made with the use of design techniques such as CCD, and serves to reduce the complexity of the design and the implementation of these intelligent systems. These techniques were used for tuning the parameters (in this case for a machining process).

The organization of this paper is as follows: Section 1 already reviewed above, Section 2 shows the state of art. Section 3 presents a general overview of the SC techniques (FLS & ANFIS). Section 4 presents the proposed method. Section 5 presents the results obtained for a machining process. Section 6 presents conclusions and finally. Section 7 presents some references.

2 State of the Art

The machining process is defined as a high precision method to model pieces for both people and industry. The most ancient machining process was made 150,000 years ago in the form of carved stone [2]. Basically, the machining processes are classified into three forms: turning, grinding and milling in the other hand drilling is added by [3]. These processes are used to give the shape of the pieces. Drilling is a critical operation because it provides the necessary holes in the assembly process [4]. The use of advanced techniques in manufacturing processes such as SC provides the chance to simulate the performance of the system without expends resources, also can be evaluated different states of the system under different conditions

In [5] is mentioned that the machining in some materials is a difficult task because there are no actual specification manuals in literature. Also, technical specifications of materials do not determine a specific kind of tool or parameters to make the process. These ideas were proposed by the International Journal of Automation Technology (IJAT) in their call for papers of a special number Machining of difficult to cut materials in 2013.

The performance of machining processes can be simulated with FIS or FLS as is mentioned in [6] also this method can be used as a control system. One of the advantages of this technique is its fast response and fast adaptation without overshoot. [7] shows the complications for drilling composites and advanced materials. They say that these processes need new approaches or new methods and technologies to adapt and tuning the parameters to drill these kinds of materials.

The ANFIS model developed by Jang in 1993 [8], now they are talking, which this is one of the forms of hybridization of the FL (Fuzzy Logic) and NN and says “ANFIS is a combination of the best part of fuzzy logic and neural networks” as mentioned in [9] and they makes a review of ANFIS models for machining process. In [10] is proposed a combination of FLS with Design of Experiments (DOE) in CCD form with good results, but their proposal uses individual base inference.

[11] shows in its proposal the use of response surface methodology, but they does not present the use of SC techniques. Their technique has a restriction of three variables when the model is multivariate, and in fact, with more than three variables

the additional variables need to be fixed, and this condition finally produces an exponential grow of the model.

[12] presents an optimized model based on the CCD / FLS and CCD/ANFIS with the approach of composed base inference. This approach reduces the complexity of the model and the rule base. In this mentioned model one of the advantages is that all states of the system are covered by the combinatorial itself. In this combinatory some parameters are fixed, and they evaluate the behavior of the remaining parameters. Another advantage or these models is that they do not need a hard mathematical model to develop FLS or ANFIS (e.g. individual base inference). The requirements for these applications are based only on historical data, and then are modeled and tuning to obtain a prediction.

Other approaches proposed in literature such as: Naïve Bayes and Ant Colony Optimization for combinatorial optimization [13], in [14] show the use of evolutionary strategies to obtain a model called evolutionary design of experiments. In [15], they used multiple techniques of SC to adjust an extrusion blow molding process; this uses a Taguchi design method, full factorial design and fractional factorial design to create a hybrid combination.

3 Preliminaries

3.1 Soft Computing Techniques Based on Fuzzy Logic (FL)

Fuzzy Logic

The FL was born in 1965 with the propositions made by Zadeh, these ones create a function in the form (1) as a matrix of rules that evaluate in quantitative form linguistic labels with a membership degree, without limitations of Boolean logic such as the human reasoning. The first proposal [16] includes crisp values. Later, an analysis presented in 1975 [17] includes linguistics in this technique. This theory becomes reality in 1975 with the Mamdani controller that is described in its paper [18].

$$\textit{Rule n: IF } X_1 \textit{ is } A_1 \textit{ and ... and } Xi \textit{ is } Ai\textit{, Then } Y \textit{ is } B \qquad (1)$$

ANFIS

In 1993 the ANFIS model appears in literature as an effort to enhance pure soft computing models with the hybrid union of FL and NN. This model is capable to adapt and learn from feedback. In the actual ANFIS applications the basic model proposed by Jang [19] does not have evolved and continues with the same proposition. The novel approaches found in literature deals in the field of learning and training with the use of: Back propagation [20], least squares [21], morphological transforms [22], and decision trees [23].

The ANFIS system is equivalent to Sugeno system (see Figure 1), and the model is a 5 layer NN (See Figure 5c). The basis for this NN is actually a fuzzy system. The main difference with FLS and ANFIS systems is the prediction algorithm, this is based on several transformations ruled in the assignation of weights to the inputs, and the main advantage is they do not need a defuzzifier at all [19].

In an ANFIS system every layer is dedicated to perform a specific function (mathematical operation), and these layers are composed of neurons that only can perform simply mathematical calculus due their simplicity. The use of this model is that it can simplify the complex problems of mathematical model, but it still needs the formulation of the fuzzy rules to assemble the ANFIS model.

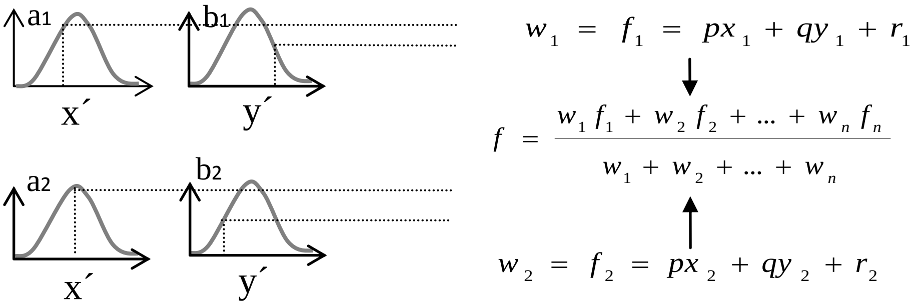

a1
b1
x´
y´
a2
b2
x´
y´
$w_1 = f_1 = px_1 + qy_1 + r_1$
$f = \frac{w_1 f_1 + w_2 f_2 + ... + w_n f_n}{w_1 + w_2 + ... + w_n}$
$w_2 = f_2 = px_2 + qy_2 + r_2$

Fig. 1. Sugeno system adapted from [8].

3.2 Design of Experiments (DOE)

The CCD is a technique of DOE, it was developed to analyze two or more factors that can have interaction between them. These factors can have some levels; levels work in an interval; edges of this interval are defined as high and low levels. However, graphic representation of the model is impossible when having more than three factors. Usually, only the three factors change in every iteration. The remaining factors are fixed in order to represent the model graphically.

Jang’s [8, 19] presents a schematic representation of a model called Type-3 ANFIS (see Figure 2a). This particular case presents nine rules in their partition space, this model would be equivalent to (2), where the dimension of space (quantity of variables) *N* represents the levels and *k* represents the factors (Fuzzy sets) that need to be evaluated. In this case a 2^k model (see Figure 2b) was selected, this model have maximum and minimum levels are known as treatments, also called axial points (see vertex of Figure 2b), this point was used as fuzzy sets that represents the inputs to the system the low and high levels represents the centers of the fuzzy sets to conform the rules. This model is very useful in the beginning of the experiments because it only needs a few tests to evaluate the behavior of the model in a confidence interval.

For the specific case of CCD, the model needs a series of samples with different combinations of the factors in order to accomplish the most possible states; this is mentioned in [24]. However, in this particular case was used to perform the rule reduction in the FLS and the ANFIS models with the use of the limits only. Commonly, all possible combinations of the levels are produced, in this case, four combinations (which are shown as sub-spaces in Figure 2a).

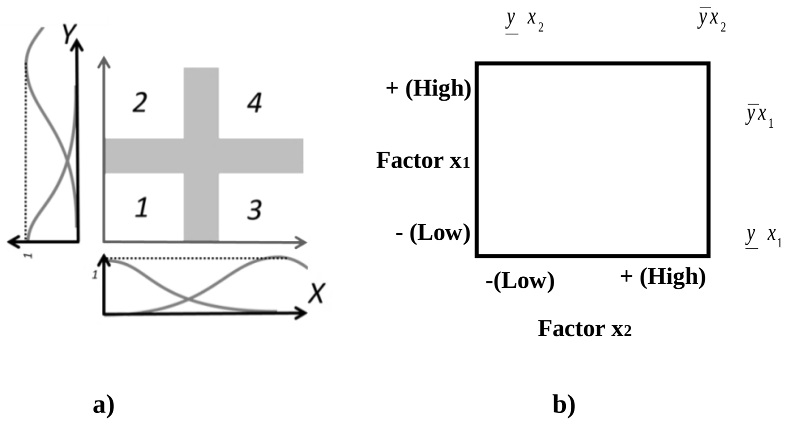


Fig. 2. a) Type-3 ANFIS with four rules. Adapted from [8, 19]. b) Factorial experiment 2^k with response (y) at the vertex. Adapted from [24].

$$N^k \tag{2}$$

Calculation of the principal effects of the axial points is needed to complete the model. Their values proceed by the combinations of the levels in every factor. In CCD the factors (for the fuzzy case variables x_1 and x_2) are called ***A*** and ***B*** in the case of high levels (for the lower levels ***a*** and ***b*** are used). The combinations are described with letters. The low levels are omitted in the symbolic representation. E.g. ***ab*** represents both variables in high level, ***b*** is used with the combination of ***a*** low and ***b*** high, ***a*** high and ***b*** low yields ***a***, finally ***1*** is used for both in low level (see Figure 3).

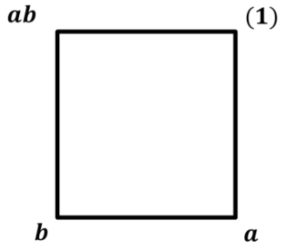


Fig. 3. CCD model graphical adaptation, adapted from [25].

The principal effect of *A* is obtained by a combination of *n* replies or samples and their effects in the low and high levels, namely the *A* effect in the low level of *B* is given by (3), in the other hand the *A* effect in the high level of *B* yields (4). Then the principal effect of *A* is obtained by the combination of (3, 4) and yields (5). A same procedure is used to obtain the principal effects of *B*, the *B* effect on low level of *A* is given by (6), the *B* effect on the high level of *A* yields (7) and the principal effect of *B* is given by (8) finally the effect of AB is given by (9).

$$\bar{y}_{A^-}=\frac{a-(1)}{2n} \tag{3}$$

$$\bar{y}_{A^+}=\frac{ab-b}{2n} \tag{4}$$

$$A=\bar{y}_{A^+}-\bar{y}_{A^-} \tag{5}$$

$$\bar{y}_{B^-}=\frac{a+(1)}{2n} \tag{6}$$

$$\bar{y}_{B^+}=\frac{ab+b}{2n} \tag{7}$$

$$B=\bar{y}_{B^+}-\bar{y}_{B^-} \tag{8}$$

$$AB=\bar{y}_{B^+}\bar{y}_{A^-} \tag{9}$$

In order to complete the CCD to obtain the coefficients to produce the regression model based on the obtained principal effects of the treatments. The correlation coefficients are obtained by (10-12).

$$\beta_0=\bar{X}=\sum_{i=1}^{n}x_i \tag{10}$$

$$\beta_1=\sum_{i=1}^{n}a_i \tag{11}$$

$$\beta_2=\frac{\sum_{i=1}^{n}b_i}{\sum_{i=1}^{n}a_i} \tag{12}$$

3.3 Prediction Model

The least squares (LS) method is used to predict the value of a function of time *t* in [24]. The linear multivariate model can be expressed as a single equation (13). This is a matrix series that produces the pseudo-inverse matrix that results in the final function and then can be predicted a value at time *t* (14). The obtained model is the same of the CCD and is used to predict some value at a point in time and evaluate their behavior.

$$\hat{\beta}=(X^TX)^{-1}X^Ty \tag{13}$$

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\cdots+\beta_kx_k+\varepsilon \tag{14}$$

4 Proposal

This work evaluates the methodologies to model fuzzy systems in the pure and hybrid forms (FLS, FLS/CCD [25], ANFIS, ANFIS/CCD [12] and FLS IBISPC [26]) to confirm the accuracy of these methods. The reduced rule base used in it saves computational time without decrement the capacity of prediction of the model.

The design of these systems is based on the fact that there are no actual methodologies in literature or only exist a few methods to model this kind of expert systems (ES). Basically the design of the ES is based on the knowledge and experience of the designer and he decides how many sets need to be created for every variable.

The models mentioned above are created with the use of the natural separations of the models to divide the universe of discourse. E.g. CCD produces a square when has 2 variables (see Figure 4). Statistical process control (SPC) uses the Gaussian distribution and it has seven divisions then if (2) is applied the CCD has 4 rules in the ES and SPC has 49 rules. These rules are produced by mixed combinations of the graphical models for 2 variables also if the model has more than 3 variables are impossible to generate a graphical model then is needed fix some parameters in order to obtain a 3 variable model to see this graphically. The 2^k factorial model gives the rule base for the ES (see Table 1). This is called principal effect or interaction matrix. It can represent a function that describes the model (15) as a multivariate function of first order.

$$a1x1@a2x2@...\ @anxn=y \qquad (15)$$

Where *a* is a scalar and @ is a mathematical operator.

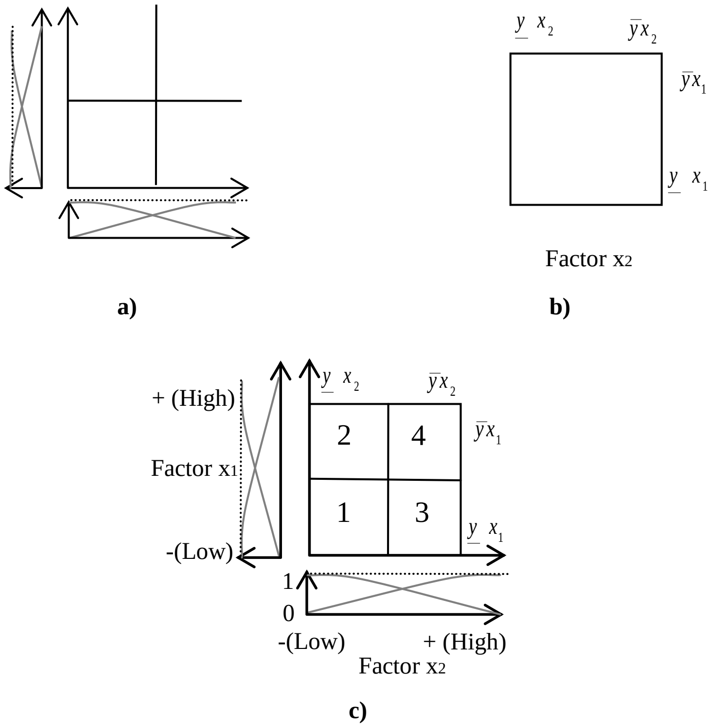


Fig. 4. Symbolic adaptation of CCD model. Adapted from [25]. a) Jang's ANFIS model, b) 2^k Factorial design, c) Hybrid model.

Table 1 Values for inputs of CCD states of every variable and combinatorial treatment.

Variable		Condition		Treatment
A	B	Low	Low	A_{low}, B_{low}
A	B	High	Low	A_{high}, B_{low}
A	B	Low	High	A_{low}, B_{High}
A	B	High	High	A_{high}, B_{High}

5 Method

The method used for modeling the ES is described in Figure 5. First, we need to use one of the described methodologies of individual base inference -such as IBISPC or

CCD- to create the universe of discourse (UOD). The partition of UOD need to be modeled with the antecedents for later obtaining the consequents (if they exists, if not create the consequents by LS). Finally, run the ES to predict any state or combination of inputs for the system.

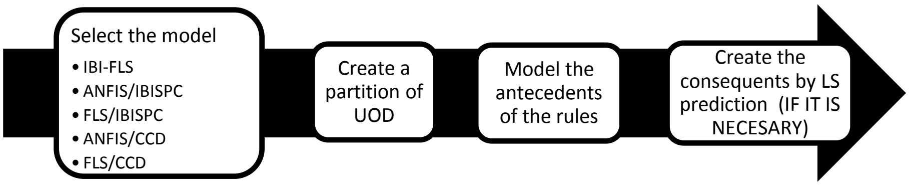

Select the model
• IBI-FLS
• ANFIS/IBISPC
• FLS/IBISPC
• ANFIS/CCD
• FLS/CCD
Create a partition of UOD
Model the antecedents of the rules
Create the consequents by LS prediction (IF IT IS NECESARY)

Fig. 5. Proposal for the methodologies comparison.

The next subsections describe the methods used to model-up the ES systems.

5.1 Tests

Test A

The IBI model uses as UOD of 11 sets for every variable and every data pair represents a rule. The final ES has 11 rules (see Table 2).

Table 2. Value for antecedents and consequents obtained by CCD (adapted from [9]).

Inputs		Output
X1, Tool feed, RPM	**X2, Spindle speed, Mm/min**	**Diameter (mm)**
2500.00	300.000	4.18
2750.00	275.000	4.08
3000.00	250.000	4.19
2750.00	239.645	4.07
2396.45	275.000	4.15
2750.00	310.355	4.10
2750.00	275.000	4.12
2500.00	250.000	4.17
3103.55	275.000	4.08
3000.00	300.000	3.97
2750.00	275.000	4.03

Test B

The IBISPC produces 7 sets for every variable and with (2), the missing consequents were obtained by prediction with LS, which uses the coefficients presented in Table 2 as their input, this represents the antecedents for the rules of FLS.

Test C

The CCD model produces only 2 sets per variable and their combination produces 4 rules for the system. Table 2 data was used to predict the output for the unknown states of the system without the need of more tests in the machining process that condition produces savings in resources because the missing states were predicted or simulated.

Test D
Evaluate the model (FLS, ANFIS, ANFIS/CCD or FLS/CCD). These models use the same UOD in both cases.

6 Experimental Results

Experiments confirm that the process works in the form of cycles such as a periodic function. The CCD model performs an evaluation to describe the states of the process as a control system. This state takes the minimum and maximum values of every variable and their permutations are called axial points in CCD. It can be transformed into fuzzy rules. Usually the number of fuzzy sets is calculated with the permutations of the variables as a thumb rule. It yields (16) and is a partition of UOD represented by the axial points in this case. The developed test runs 11 samples and 6 different numbers of epochs for every sample.

$$FS^{x} \tag{16}$$

Where $FS = \#\ of\ fuzzy\ sets$ and $X = \#\ variables.$

The rule reduction can be done with the study of performance of the process. This brings a periodic function behavior (see Figure 6). *Y* axis represents the output for the ES: the function represents the system and its units are the diameter of the drilling operation. *X* axis represents the different test (one was simulated and the remaining ones were actual physical tests). The frequency is given by the tool feed. The spindle speed explains the amplitude. The experiment can be made with a single period as a reduced base of rules. The results obtained by one period simplification in the form of DOE method also provide a novel model to perform a rule reduction.

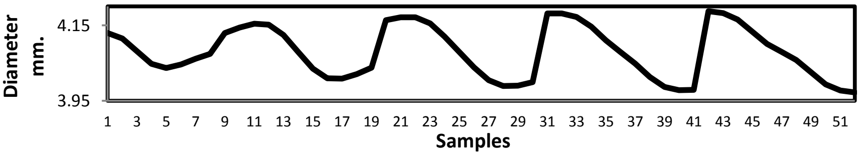


Fig. 6. Performance of machining process, response case (simulated).

The rule reduction is performed as following:

- Taking a period of the output
- As shown in Figure 3: the point in the left lower corner represents the lowest state of the system, the upper right corner represents the maximum state of the system as axial points for the spindle speed, the mean value represents the middle points for both axis. The crossover points are represented by intervals between the minimum or maximum and the mean.

The rest of rules can be obtained by linear combinations of the axial points with the scalar product and these combinations are not necessary for the systems because the activation of these systems is always done at least with a pair of rules. Figure 7 shows a period in the response function, this function only found a change in the amplitude when the speed spindle changes (RPM). If the speed arises the amplitude arises and the frequency is constant at the time.

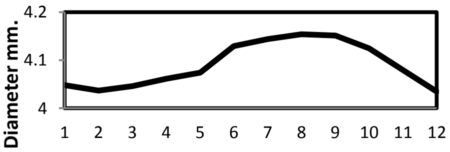


Fig. 7. Periodic performances for every fuzzy set of tool feed.

Figure 8 shows the performance of the systems. The chaotic behavior of the function precedes the nature of treatments of the CCD method. Tests need to be performed in a random order to avoid the appearance of either noise in the system or the cause of a trend. On the other hand, the models used (LS, FLS, ANFIS) are not exact methods and they can only make approximations of a function. Some variations of this performance can be explained by the wear and tear of the tool. Just as same, the similarity in the approximations generated by the purity or hybrid models and produced by the pre-processing of these methods is based in a fuzzification of the data and the variations in the processing of the following phases small variations due to differences in the calculated decimal by the nature of the model.

The sample number 10, of the table 2 is a special case because the variation can be caused by deterioration of the tool or can be considered as an atypical point. Anyway, this can be considered in a later study. Figure 8a shows the results obtained by the fuzzy systems (IBI and IBISPC) and hybrid form (FLSCCD). The differences between the error rate in FLS and optimized FLS are no significant in comparison of the savings in time and modeling (see Table 3). The results obtained by the hybrid ANFIS with the CCD show that this class of systems can be actually made. The error rate in ANFIS compared against the goal values is very similar to an FLS model (see Figure 8b). These models show good approximation capabilities. The computational time consumed in the CCD models is half in comparison of the IBI model and 10 % in comparison of IBISPC model all results are shown in Table 4.

Table 3. Computational time of different tests with 10 epochs of training (seconds).

FLSCCD	FLSIBI	FLSISIBISPC	ANFISCCD	ANFISIBI	ANFISIBISPC
0.026	0.053	0.267	**0.220**	0.039	0.125

Table 4. MSE of different models 10 epochs of training.

FLSCCD	FLSIBI	FLSISIBISPC	ANFISCCD	ANFISIBI	ANFISIBISPC
0.0024	**0.0018**	0.0030	0.0025	0.0031	0.0030

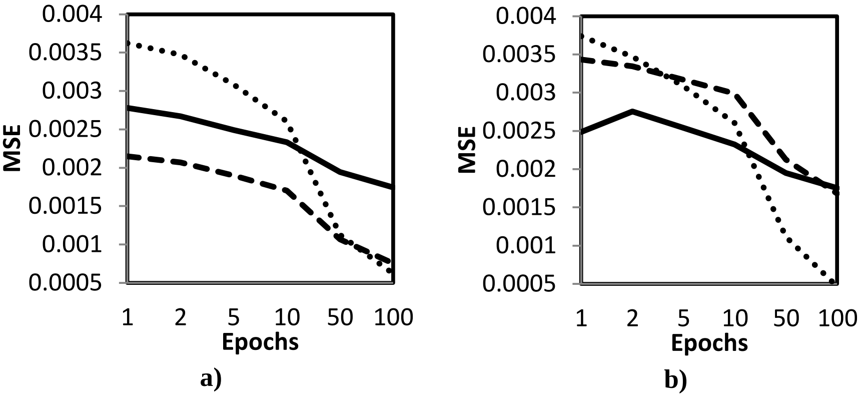


Figure 8. MSE of different models. a) Fuzzy models, b) ANFIS models. Where: solid lines are CCD, dotted lines are IBISPC, and dashed lines are IBI.

7 Conclusion

The DOE techniques provide a simplification for modeling SC systems. The interrelation generated by the CCD serves as functions that can be used to model the rules in the form described in (1). These design techniques can also reduce the modeling work, because the analysis made by CCD evaluates the interaction between variables and with these interactions non-correlated variables can be eliminated of the model.

These kinds of systems were quite useful in generating an approximation or prediction to study the possible performance of the process in different conditions using less resources. Tests can be made by simulations with SC techniques. A complete extended rule base SC model with hybrid forms (FLS/CCD OR ANFIS/CCD) can prove its performance against a pure form (IBISPC) at least in this case.

The performance of the systems shows that the CCD can be used as a good model to calculate an approximation to FLS and ANFIS systems with a reduced rule base without having to worry about losing reliability. The use of these hybrid models (FLS/DOE and ANFIS/DOE) can reduce the size of the model around 80% without decrement the capacity of prediction (in comparison with the individual base inference model). Also, it can reduce the training cycles depending on the precision required.

References

1. Vale, Z., Morais, H., Faria, P., Ramos, C.: Distribution system operation supported by contextual energy resource management based on intelligent SCADA. Renewable Energy, vol. 52, pp. 143–153 (2013)

2. Gutowski T.: Machining. Lectures of manufacturing processes and systems http://web.mit.edu/2.810/www/lecture09/4.pdf (2009)
3. Özel T., Davim P.: Intelligent Machining, control systems, robotics and manufacturing series. 1st edition. Wiley-ISTE. pp. 1–61 (2009)
4. Krishnamoorthy, A., Rajendra Boopathy, S., Palanikumar, K., Paulo Davim, J.: Application of grey fuzzy logic for the optimization of drilling parameters for CFRP composites with multiple performance characteristics. Measurement, 45(5), pp. 1286–1296 (2012)
5. Fujipress.: CALL FOR PAPERS, Special Issue on Machining of Difficult-to-cut Materials for the International Journal of Automation Technology (IJAT), Vol.7 No.3 (2013)
6. Haber, R. E., del Toro, R. M., Gajate, A.: Optimal fuzzy control system using the cross-entropy method. A case study of a drilling process. Information Sciences, 180(14), pp. 2777–2792 (2010)
7. Rajmohan, T., Palanikumar, K., Prakash, S.: Grey-fuzzy algorithm to optimise machining parameters in drilling of hybrid metal matrix composites. Composites Part B: Engineering, 50, pp. 297–308 (2013)
8. Jang, J. S.: ANFIS: adaptive-network-based fuzzy inference system. Systems, Man and Cybernetics, IEEE Transactions on, 23(3), pp. 665–685 (1993)
9. Kabini, K.: Review of ANFIS and its Application in Control of Machining Processes. Sustainable Research and Innovation Proceedings (2011)
10. Praga-Alejo R., González G. D., Pérez V. P, Cantú S. M., Flores H. B.: Modeling a Fuzzy Logic System Using Central Composite Design. Proceedings of 1st annual world Conference of the Society for Industrial and Systems Engineering. Washington D.C., USA, September 16-18 (2012)
11. Makadia, A. J., Nanavati, J. I.: Optimisation of machining parameters for turning operations based on response surface methodology. Measurement, 46(4), pp. 1521–1529 (2013)
12. Montes Dorantes P. N., Praga-Alejo R., Nieto Gonzalez J.P., Méndez G.M.: Modelado de Sistemas Adaptativos de Inferencia Neuro-Difusa Usando Diseño Central Compuesto. Research In Computing Science, vol. 62, pp. 259–269 (2013)
13. Borrotti M., Poli I.: Naïve Bayes Ant Colony Optimization for Experimental Design. Synergies of Soft Computing and Statistics for Intelligent Data Analysis Advances in Intelligent Systems and Computing Volume 190, 2013, pp 489–497, Springer-Verlag (2013)
14. Balasko B., Madar J., Abonyi J.: Additive Sequential Evolutionary Design of Experiments. Artificial Intelligence and Soft Computing – ICAISC 2006. Lecture Notes in Computer Science Volume 4029, 2006, pp 324–333. Springer-Verlag (2006)
15. Yu J., Tsung R.H, Juang J., Thibault F.: Design Optimization Using Soft Computing Techniques for Extrusion Blow Molding Processes. 2002 NRC-NSC Canada-Taiwan Joint Workshop on Advanced Manufacturing Technologies, London, Canada. pp. 73–84 (2002)
16. Zadeh L.A.: Fuzzy Sets. Information and control 8(3), pp.338–353 (1965)
17. Zadeh L.A.: The concept of a linguistic variable and its application to approximate reasoning I. Information Sciences, 8(1975). pp.199–249 (1975)
18. Mamdani, E. H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International journal of man-machine studies, 7(1), pp. 1–13 (1975)
19. Jang, J. S. R., Sun, C. T.: Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence. Prentice-Hall, Inc. (1996)
20. Hosseini, M. S., Zekri, M.: Review of Medical Image Classification using the Adaptive Neuro-Fuzzy Inference System. Journal of medical signals and sensors, 2(1), 49 (2012)
21. Mendez, G. M., de los Angeles Hernandez, M.: IT2 TSK NSFLS2 ANFIS. In Artificial Intelligence (MICAI), 2010 Ninth Mexican International Conference on. pp. 89–93, IEEE, (2010)

22. González M. A., & Ballarin V. L.: Segmentación de imágenes utilizando la transformada Watershed: obtención de marcadores mediante lógica difusa IEEE latin america transactions, 6(2), pp. 223—228 (2008)
23. Dutra R. G., & Martucci Jr. M.: Adaptive Fuzzy Neural Tree Network IEEE latin america transactions, 6(5). pp. 456—460 (2008)
24. Montgomery D.C: Diseño y Análisis de experimentos. 2ª Ed. Limusa-Wiley. pp.218—279 (2004)
25. Montes Dorantes P. N., Nieto González J.P., Praga-Alejo R., Guajardo Cosio K.L., & Méndez G.M.: Sistema inteligente para procesamiento de imágenes en control de calidad basado en el modelo difuso singleton tipo 1. RCS 74, pp. 117—130 (2014)
26. Montes Dorantes P. N., Nieto González J.P., Praga-Alejo R., & Méndez G.M.: Fault Detection Systems via a Novel Hybrid Methodology for Fuzzy Logic Systems Based on Individual Base Inference and Statistical Process Control. IEEE latin america transactions, 12(4), pp. 706—712 (2014)

Reviewing Committee

Alexander Gelbukh
Felix Castro Espinoza
Francisco Viveros Jiménez
Grigori Sidorov
Gustavo Arroyo Figueroa
Hugo Terashima Marín
Ildar Batyrshin
Jesús González Bernal
Luis Villaseñor Pineda
Maya Carillo Ruiz
Miguel González Mendoza
Noé Alejandro Castro Sánchez
Obdulia Pichardo Lagunas
Omar Montaño Rivas
Oscar Herrera Alcantara
Rafael Murrieta Cid
Sabino Miranda Jiménez
Sofía Galicia Haro

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Noviembre de 2014
Printing 500 / Edición 500 ejemplares