

# Modificación al algoritmo de Hooke-Jeeves para búsqueda local en variantes de evolución diferencial para resolver problemas de optimización con restricciones

Angel Juan Sánchez García<sup>1</sup>, Homero Vladimir Ríos Figueroa<sup>2</sup>, Gerardo Contreras Vega<sup>1</sup>, Juan Carlos Pérez Arriaga<sup>1</sup> y María Karen Cortés Verdín<sup>1</sup>

<sup>1</sup> Facultad de Estadística e Informática, Universidad Veracruzana, Xalapa, Veracruz, México

<sup>2</sup> Centro de Investigación en Inteligencia Artificial, Universidad Veracruzana, Xalapa, Veracruz, México

{angesanchez, hrios, gcontreras, juaperez, kcortes}@uv.mx

**Resumen.** Este trabajo presenta una modificación del algoritmo de Hooke-Jeeves implementado en variantes de Evolución Diferencial para resolver problemas de optimización con restricciones. El algoritmo de Hooke-Jeeves promueve una mejor exploración y explotación en zonas prometedoras para encontrar mejores soluciones. El algoritmo de Hooke-Jeeves modificado es implementado en 4 variantes de Evolución Diferencial: DE/rand/1/bin, DE/best/1/bin, Modified Differential Evolution (MDE) y Differential Evolution Combined Variants (DECV). Este enfoque es probado en un conjunto de problemas de referencia sobre optimización con restricciones. Los resultados son discutidos y algunas conclusiones son establecidas.

**Palabras clave:** Hooke-Jeeves, optimización, restricción, evolución diferencial, búsqueda local.

## 1. Introducción

Los algoritmos evolutivos han sido ampliamente utilizados para resolver problemas de optimización [1,2]. Sin embargo, en sus versiones originales carecen de mecanismos para hacer frente a problemas con restricciones [3], por ello se han hecho modificaciones a estos algoritmos.

Evolución diferencial (ED) es un algoritmo evolutivo propuesto por Storm y Price [4] para resolver problemas de optimización principalmente en espacios continuos. ED se diferencia de otros algoritmos evolutivos en que no utiliza una codificación binaria como los algoritmos genéticos [5] ni utiliza una función de densidad de probabilidad autoadaptativa como en la estrategia evolutiva [6].

Con el objetivo de obtener mejores resultados basados en el enfoque de ED, se han propuesto algunas variantes de este algoritmo caracterizadas por el tipo

de recombinación y el número y tipo de soluciones para calcular los valores de las mutaciones. A continuación se describen brevemente las variantes utilizadas para la incorporación de nuestro enfoque de búsqueda local.

La variante más popular es DE/rand/1/bin, donde *DE* significa Evolución Diferencial, *rand* indica que los individuos que son seleccionados para ser mutados se escogen de manera aleatoria, *1* es el número de pares de soluciones escogidos y *bin* significa que se utiliza una recombinación binomial [4]. En DE/best/1/bin la única diferencia con respecto a DE/rand/1/bin es que el vector base no se escoge aleatoriamente, si no que es el mejor vector de la población actual. Con respecto a Modified Differential Evolution (MDE) en [3], se propone que la incorporación de la información de la mejor solución y el padre actual, acoplado con un mecanismo que permita a cada padre generar más de un descendiente, aumenta la capacidad de exploración y explotación el algoritmo ED en problemas de optimización. También que la incorporación de un mecanismo de diversidad permite que ED se acerque a la región factible de una mejor manera como para alcanzar el óptimo global [3].

En Differential Evolution Combined Variants (DECV), los autores en [7] proponen la combinación de DE/rand/1/bin y DE/best/1/bin, donde DE/rand/1/bin es utilizado primero para generar un conjunto más diverso de direcciones de búsqueda y cambiar a DE/best/1/bin para centrar la búsqueda en la vecindad del mejor vector posible, con un criterio de cambio al obtener el 10% de los vectores factibles. La investigación sobre búsqueda local aplicada a Evolución Diferencial para resolver problemas de optimización numérica con restricciones, es la principal motivación de este documento, en el cuál una modificación al algoritmo de Hooke-Jeeves es presentado para mejorar la búsqueda local.

Este documento está organizado como sigue: en la sección 2 la definición de los problemas de optimización numérica con restricciones es presentada, en la sección 3 son presentados algunos trabajos relacionados que dan pie a la motivación de este trabajo, en la sección 4 la descripción de la búsqueda local es presentada. En la sección 5 es descrita nuestra propuesta, mientras que en la sección 6 los resultados y la discusión son presentados. Por último en la sección 7 se presentan las conclusiones y el trabajo futuro propuesto.

## 2. Planteamiento del problema

Los problemas que se pretenden resolver, son problemas en general no lineales que se definen como sigue:

Encontrar  $x$  tal que optimice  $f(x)$

Sujeta a:

$$g_i(x) \leq 0, i = 1, \dots, p \quad (1)$$

$$h_j(x) = 0, j = 1, \dots, q \quad (2)$$

donde  $x$  es el vector de parámetros  $= [x_1, x_2, \dots, x_n]^T$ ,  $p$  es el número de restricciones de desigualdad y  $q$  el número de restricciones de igualdad ( $p$  y  $q$

pueden ser lineales o no lineales). Para manejar las restricciones de igualdad, usualmente son transformadas a desigualdades como sigue [8]:  $|h_j(x) - \varepsilon| \leq 0$ , donde  $\varepsilon$  es una tolerancia permitida (un valor muy pequeño). El valor de  $\varepsilon = 1 \times 10^{-4}$  [9] fue usado en este trabajo.

### 3. Trabajos relacionados

Otros autores han propuesto enfoques que se acoplan con algoritmos evolutivos para obtener mejores resultados. Un nuevo criterio de selección para soluciones candidatas, conjugado con un operador basado en el método Nelder-Mead, es el enfoque *HDESMCO* propuesto en [10], donde se puede ver la idea de incorporar algún mecanismo que ayude a *ED* a encontrar mejores soluciones.

En [11] se propone la inclusión de una búsqueda local asistida por un meta modelo basado en máquinas de soporte de vectores donde se involucra el método de Hooke-Jeeves. Este mecanismo de búsqueda local es incluido en algoritmos evolutivos multi-objetivo.

La búsqueda local, ha tomado gran importancia recientemente para tratar de alcanzar óptimos globales. En [12], los autores desarrollan un marco de optimización multiobjetivo basado en la clasificación no dominado y búsqueda local NSLS (por sus siglas en inglés). El NSLS se basa en iteraciones. En cada iteración, dada una población  $P$ , un método simple de búsqueda local es usado para obtener una mejor población  $p'$ , y luego la clasificación no dominada se adopta en  $P \cup P'$  para establecer una nueva población en la próxima iteración. Además, el candidato más lejano se combina con la clasificación de los no dominados para elegir la nueva población y así mejorar la diversidad.

En [13] se propone un cómputo memético adaptativo como la sinergia de un algoritmo genético, una evolución diferencial y estimación de un algoritmo de distribución. La relación entre el número de soluciones producidas por los algoritmos en una generación definen características de adaptabilidad en la próxima generación. Lo importante de la relación de este trabajo con el nuestro, es que posteriormente, un subconjunto de las soluciones pasa por la evolución de búsqueda local utilizando el algoritmo de búsqueda de gradiente.

En [9] se incluye el método de Hook-Jeeves como mecanismo de búsqueda local aplicado a la modificación del algoritmo *Modified Bacterial Foraging Algorithm* (MBFOA).

### 4. Búsqueda local

El método de búsqueda local trabaja creando un conjunto de direcciones. Las direcciones de búsqueda deben ser independientes y cubrir todo el dominio de  $f(x)$ . Por lo tanto, en un problema  $N$ -dimensional, este método requiere por lo menos  $N$  direcciones de búsqueda linealmente independientes [14].

#### 4.1. Método de patrones de búsqueda de Hooke-Jeeves

El método de patrones de búsqueda de Hooke-Jeeves [15] crea un conjunto de direcciones de búsqueda de manera iterativa. Este método fue propuesto en 1966 y fue uno de los primeros algoritmos en incorporar la historia previa de una secuencia de iteraciones en la generación de una nueva dirección de búsqueda. El algoritmo combina movimientos *exploratorios* y movimientos de *patrones* con alguna heurística.

#### 4.2. Movimientos exploratorios

Los movimientos exploratorios examinan la vecindad del punto actual para encontrar el mejor punto alrededor del mismo [14]. Por lo tanto, los movimientos exploratorios examinan el comportamiento local de la función y buscan localizar la dirección de cualquier pendiente existente en la zona.

#### 4.3. Movimientos de patrones

Este movimiento es realizado por dos puntos, el original y el nuevo. El movimiento de patrones utiliza la información generada en la exploración para escalar rápidamente las pendientes. Si el movimiento de patrones no toma la solución a una mejor región, entonces no tiene éxito y se reduce el alcance de la búsqueda exploratoria. Esto se repite hasta converger, es decir, cuando el movimiento prácticamente es mínimo. El nuevo punto es encontrado por el salto del mejor punto actual en la dirección que conecte el anterior mejor punto  $x^{(k-1)}$  y el actual punto base  $x^{(k)}$  como se muestra en 3:

$$xp^{(k+1)} = x^{(k)} + (x^{(k)} - x^{(k-1)}) \quad (3)$$

### 5. Nuestra propuesta

Cuando se habla de una búsqueda local existen por lo menos tres preguntas que se deben responder: ¿A quién se le aplica la búsqueda local? ¿Con qué frecuencia? y ¿Cuál será la condición de paro de la búsqueda?. La idea de nuestra propuesta es explorar sólo en las mejores soluciones y que esta exploración se haga de manera medida, es decir, no dedicar muchas evaluaciones en explorar las soluciones prometedoras. Si estas soluciones prometedoras no son los caminos que nos llevan al óptimo global, podremos ir tomando otras alternativas mediante el algoritmo de *ED* (en cualquier variante). Pero si las mejores soluciones nos llevan al óptimo global, entonces seguirán siendo exploradas cada vez que se realice la búsqueda local, puesto que seguirán siendo mejores soluciones con el paso del tiempo.

El movimiento de exploración se toma como en la versión original y se muestra en el algoritmo 1, donde un valor de  $\alpha = 2$  es recomendado [14]. Nosotros proponemos que este movimiento exploratorio se le aplique al 3 % de la población.

---

**Algorithm 1** Movimiento Exploratorio

---

```

1: Inicializar parámetros
2: Escoger un punto de inicio  $x^{(0)}$ 
3:  $k = 0$ 
4: for  $i = 1$  a 10 do
5:   Realizar un movimiento exploratorio con  $x^{(k)}$  como punto base
6:   if el movimiento exploratorio = éxito then
7:      $x^{(k+1)} = x$ 
8:      $k = k + 1$ 
9:     realizar el movimiento de patrones:  $x_p^{(k+1)} = x^{(k)} + (x^{(k)} - x^{(k-1)})$ 
10:  else
11:     $\Delta_i = \Delta_i / \alpha$ 
12:  end if
13: end for

```

---

En este 3% se encuentran las mejores soluciones. Se aplica solo un 3% debido a que no tiene caso explorar más soluciones que no son tan prometedoras como las mejores. Así mismo, no recomendamos que el movimiento exploratorio sea solo en la mejor solución, sino tomar más de un camino prometedor. Para el tamaño de la exploración recomendamos que sea obtenido de la variable con menor rango como se muestra en (4):

$$(\text{Límite superior} - \text{límite inferior}) / 100 \quad (4)$$

Lo anterior es debido a que la exploración dependerá del espacio de búsqueda donde se pueda explorar. La frecuencia de la aplicación de la búsqueda local es aplicada cada generación, con el fin de explorar más veces aquellas mejores soluciones después de cada generación. La parte donde difiere nuestra propuesta a la de Hooke-Jeeves original, es la condición de paro. Nosotros fijamos un número de iteraciones máximo que el algoritmo puede explorar. Lo anterior es realizado con el fin que no explore demasiado, puesto que alguna buena solución aparente puede llevarnos a un óptimo local. Si la solución debe seguir siendo explorada después de las 10 iteraciones propuestas, la siguiente vez que se ejecute este algoritmo de búsqueda, esa solución volverá a ser seleccionada para ser explorada siempre y cuando esté dentro del 3% de las mejores soluciones. Además con esta restricción se evita la calibración de más parámetros.

El algoritmo de Hooke-Jeeves modificado es presentado en el algoritmo 2.

## 6. Experimentos y discusión de resultados

Dos experimentos fueron realizados para analizar el desempeño de nuestro enfoque de búsqueda local:

1. Una comparación entre las variantes de *ED* con y sin nuestro enfoque.
2. Una comparación contra los resultados de otro algoritmo que ocupa la búsqueda local de Hooke-Jeeves: MBFOA [9].

---

**Algorithm 2** Método de búsqueda de patrones de Hooke-Jeeves modificado

---

```

1:  $x = x^{(c)}$ 
2:  $i = 1$ 
3: Calcular  $f = f(x)$ ,  $f^+ = (x_i + \Delta_i)$  y  $f^- = (x_i - \Delta_i)$ 
4: Encontrar  $f_{min} = \min(f, f^+, f^-)$  basado en las reglas de factibilidad
5:  $x = f_{min}$ 
6: if  $i < N$  then
7:    $i = i + 1$ 
8:   ir al paso 3
9: end if
10: if  $x \neq x^c$  then
11:   Exito
12: else
13:   Fallo
14: end if

```

---

Ocho de los problemas conocidos [16] son usados en las comparaciones propuestas. Las características de los problemas abordados están resumidos en el cuadro 1. Las definiciones son descritas en [17] y son mostradas a continuación:

1. **g01:**

Minimizar  $f(\vec{x}) : 5 \sum_{i=1}^4 x_i - 5 \sum_{i=1}^4 x_i^2 - \sum_{i=5}^{13} x_i$  sujeta a:

$$g1(\vec{x}) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0$$

$$g2(\vec{x}) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0$$

$$g3(\vec{x}) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0$$

$$g4(\vec{x}) = -8x_1 + x_{10} \leq 0$$

$$g5(\vec{x}) = -8x_2 + x_{11} \leq 0$$

$$g6(\vec{x}) = -8x_3 + x_{12} \leq 0$$

$$g7(\vec{x}) = -2x_4 - x_5 + x_{10} \leq 0$$

$$g8(\vec{x}) = -2x_6 - x_7 + x_{11} \leq 0$$

$$g9(\vec{x}) = -2x_8 - x_9 + x_{12} \leq 0$$

Donde  $0 \leq x_i \leq 1 (i = 1, \dots, 9)$ ,  $0 \leq x_i \leq 100 (i = 10, 11, 12)$  y  $0 \leq x_{13} \leq 1$ .

2. **g02:**

Maximizar  $f(\vec{x}) : \left| \frac{\sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i)}{\sqrt{\sum_{i=1}^n ix_i^2}} \right|$  sujeta a:

$$g1(\vec{x}) = 0,75 - \prod_{i=i}^n x_i \leq 0$$

$$g2(\vec{x}) = \sum_{i=i}^n x_i - 7,5n \leq 0$$

Donde  $n = 20$  y  $0 \leq x_i \leq 10 (i = 1, \dots, n)$ .

3. **g04:**

Minimizar  $f(\vec{x}) : 5,3578547x_3^2 + 0,8356891x_1x_5 + 37,293239x_1 - 40792,141$  sujeta a:

$$g1(\vec{x}) = 85,334407 + 0,0056858x_2x_5 + 0,0006262x_1x_4 - 0,0022053x_3x_5 - 92 \leq 0$$

$$g2(\vec{x}) = -85,334407 - 0,0056858x_2x_5 - 0,0006262x_1x_4 + 0,0022053x_3x_5 \leq 0$$

$$g3(\vec{x}) = 80,51249 + 0,0071317x_2x_5 + 0,0029955x_1x_2 + 0,0021813x_3^2 - 110 \leq 0$$

$$g4(\vec{x}) = -80,51249 - 0,0071317x_2x_5 - 0,0029955x_1x_2 - 0,0021813x_3^2 + 90 \leq 0$$

$$g5(\vec{x}) = 9,300961 + 0,0047026x_3x_5 + 0,0012547x_1x_3 + 0,0019085x_3x_4 - 25 \leq 0$$

$$g6(\vec{x}) = -9,300961 - 0,0047026x_3x_5 - 0,0012547x_1x_3 - 0,0019085x_3x_4 + 20 \leq 0$$

Donde  $78 \leq x_1 \leq 102$ ,  $33 \leq x_2 \leq 45$  y  $27 \leq x_i \leq 45 (i = 3, 4, 5)$ .

4. **g06:**

Minimizar  $f(\vec{x}) : (x_1 - 10)^3 + (x_2 - 20)^3$  sujeta a:

$$g1(\vec{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$$

$$g2(\vec{x}) = (x_1 - 6)^2 - (x_2 + 5)^2 - 82,81 \leq 0$$

Donde  $13 \leq x_1 \leq 100$  y  $0 \leq x_2 \leq 100$ .

5. **g07:**

Minimizar  $f(\vec{x}) : x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + 45$  sujeta a:

$$g1(\vec{x}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0$$

$$g2(\vec{x}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0$$

$$g3(\vec{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0$$

$$g4(\vec{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0$$

$$g5(\vec{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0$$

$$g6(\vec{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0$$

$$g7(\vec{x}) = 0,5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_1^2 - x_6 - 30 \leq 0$$

$$g8(\vec{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0$$

Donde  $-10 \leq x_1 \leq 10 (i = 1, \dots, 10)$ .

6. **g08:**

Maximizar  $f(\vec{x}) : \frac{\sin^3(2\pi x_1) \sin(2\pi x_2)}{x_1^3(x_1x_2)}$  sujeta a:

$$g1(\vec{x}) = x_1^2 - x_2 + 1 \leq 0$$

$$g2(\vec{x}) = 1 - x_1 + (x_2 - 4)^2 \leq 0$$

7. **g09:**

Minimizar  $f(\vec{x}) : (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 + 4x_6x_7 - 10x_6 - 8x_7$  sujeta a:

$$g1(\vec{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0$$

$$g2(\vec{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0$$

$$g3(\vec{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0$$

$$g4(\vec{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0$$

Donde  $-10 \leq x_1 \leq 10 (i = 1, \dots, 7)$ .

8. **g10:**

Minimizar  $f(\vec{x}) : x_1 + x_2 + x_3$  sujeta a:

$$g1(\vec{x}) = -1 + 0,0025(x_4 + x_6) \leq 0$$

$$g2(\vec{x}) = -1 + 0,0025(x_5 + x_7 - x_4) \leq 0$$

$$g3(\vec{x}) = -1 + 0,01(x_8 - x_5) \leq 0$$

$$g4(\vec{x}) = x_1x_6 + 833,33252x_4 + 100x_1 - 83333,333 \leq 0$$

$$g5(\vec{x}) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0$$

$$g6(\vec{x}) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0$$

Donde  $100 \leq x_1 \leq 10000$ ,  $1000 \leq x_i \leq 10000 (i = 2, 3)$  y  $10 \leq x_i \leq 1000 (i = 4, \dots, 8)$ .

En los dos experimentos se realizaron 30 ejecuciones independientes por cada variante (con y sin búsqueda). Los parámetros utilizados fueron para todas las

variantes: tamaño de población = 200, CR= 0.5.

Para DE/rand/1 bin y DE/best/1/bin F = 0.5. Para MDE Fa = .4 y Fb = 0.6. Para DECV el porcentaje de vectores factibles = 10%. Para el método modificado de Hooke-Jeeves  $\alpha = 2$  y  $\Delta$  como se propone en la sección 5 fueron utilizados. El número de evaluaciones máximas fueron 220,000.

**Tabla 1.** Características principales de los problemas de prueba,  $n$  indica la dimensión del problema,  $\rho$  es el tamaño estimado de la región factible con respecto a todo el espacio de búsqueda,  $LI$  y  $NI$  son el número de restricciones de desigualdades lineal y no lineal respectivamente y  $LE$  y  $NE$  son el número de restricciones de igualdades lineales y no lineales respectivamente.

Num	N	Función	$\rho$	LI	NI	LE	NE
g01	13	Cuadrática	0.0111%	9	0	0	0
g02	20	No Lineal	99.9971%	0	2	0	0
g04	5	Cuadrática	52.1230%	0	6	0	0
g06	2	Cúbica	0.0000%	0	2	0	0
g07	10	Cuadrática	0.0003%	3	5	0	0
g08	2	No Lineal	0.8560%	0	2	0	0
g09	7	Polinomial	0.5121%	0	4	0	0
g10	8	Lineal	0.0010%	0	3	0	0

Los resultados del primer experimento son mostrados en los cuadros 2 y 3, donde algunas variantes de ED, principalmente MDE fueron capaces de encontrar soluciones factibles y soluciones óptimas en la mayoría de los problemas. En los cuadros 2 y 3 se puede observar que se encontraron los óptimos globales cuando se le acopló nuestro método propuesto a alguna variante de ED. Las variantes que más óptimos globales encontraron fueron MDE y DE best/1/bin con la búsqueda local propuesta en  $g01$ ,  $g04$  y  $g08$ , y en  $g01$ ,  $g08$  y  $g09$  respectivamente.

Con base en los cuadros 2 y 3, también se puede notar que en prácticamente todas las funciones, la variante de ED con la búsqueda local propuesta supera a la variante de ED sin búsqueda. También se puede notar en los resultados de este enfoque, que es sensible a los problemas con restricciones de igualdad, puesto que es donde no pudo desempeñarse de mejor manera. Con base en los cuadros 2 y 3, los mejores resultados son obtenidos con las variantes de ED cuando existen más restricciones que cuando tiene menos ( $g01$ ,  $g04$ ,  $g09$ ).

También se afirma con base en las tablas que entre más sea el rango de las variables del problema, las desviación estándar suele ser mayor en nuestro enfoque con en el caso de  $g04$  y  $g06$ .

En el cuadro 4 la comparación contra MBFOA es presentada. Se observa que en general, la búsqueda local ayuda a obtener el óptimo global. Se aprecia de igual manera en el cuadro 4 que en unos problemas de prueba MBFOA es

mejor, y en otros problemas nuestra búsqueda con *ED* es mejor (principalmente con *MDE*).

## 7. Conclusiones y trabajo futuro

La implementación de una modificación al algoritmo de búsqueda local de Hooke-Jeeves acoplado con variantes de Evolución Diferencial fue presentada en este documento, con el objetivo de mejorar su desempeño en problemas de optimización numérica con restricciones. La búsqueda local propuesta restringe el número de movimientos exploratorios y de soluciones seleccionadas para aplicarles la búsqueda local, con el fin realizar más evaluaciones con el algoritmo de Evolución Diferencial y no enfocar demasiado la búsqueda en soluciones que aparentemente son prometedoras pero pudieran llevarnos a un óptimo local. Esta propuesta fue probada con las variantes de ED: *DE rand/1/bin*, *DE best/1/bin*, *MDE* y *DECV* sobre once de los problemas bien conocidos.

Los resultados sugieren que una búsqueda local en soluciones prometedoras, ayudan a explorar y explotar de mejor manera esas soluciones, lo que lleva a los algoritmos de Evolución Diferencial a encontrar mejores soluciones. Esta explotación y exploración es debido a que la búsqueda se hace en las  $N$  direcciones del problema. Además, la búsqueda local ayuda que explorar y explotar las mejores soluciones debido que no se tiene que esperar a que el algoritmo de *ED* vuelva a explorar todas las soluciones.

Como trabajo futuro proponemos lo siguiente: La aplicación de búsqueda local en este trabajo se aplica en cada generación al 3% de la población, siendo ese 3% las mejores soluciones con base en las reglas de factibilidad. Entonces se podrían hacer pruebas donde no se aplique la búsqueda local hasta que se encuentren soluciones factibles. Lo anterior debido a que cuando se aplica la búsqueda local a soluciones no factibles, se tomará como segundo criterio la aptitud, entonces se irán minimizando los valores de las aptitudes de las soluciones no factibles, y en ese caso de deberá seguir explorando con el algoritmo de Evolución Diferencial.

Por último se propone implementar búsquedas locales en otros paradigmas como inteligencia colectiva o sistemas inmunes artificiales.

## Referencias

1. Zbigniew, M.: Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, third edition (1996)
2. Coello, C., Van Veldhuizen, D., Lamont, G.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, New York (2002)
3. Mezura, E., Velásquez, J., Coello, C.: Modified Differential Evolution for Constrained Optimization. In: Proceedings of the Congress on Evolutionary Computation, pp. 332–339 (2006)
4. Price, K.: An introduction to differential evolution. In: David Corne, Marco Dorigo, and Fred Glover, editors, New Ideas in Optimization, pp. 79–108, Mc Graw-Hill, UK (1999)

5. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Co., Reading, Massachusetts (1989)
6. Schwefel, H.: Evolution and Optimization Seeking. John Wiley y Sons, New York (1995)
7. Mezura, E., Miranda, M., Gomez, R.: Differential Evolution in constrained numerical optimization: An empirical study. *Information Sciences*, vol. 180, no. 22, pp. 4223–4262 (2010)
8. Runarsson, T., Yao, X.: Stochastic Ranking for Constrained Evolutionary Optimization. In: *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294 (2000)
9. Mezura, E., López, E.: Adaptation and local search in the modified bacterial foraging algorithm for constrained optimization. In: *Evolutionary Computation (CEC), IEEE Congress on* (2012)
10. Menchaca, A., Coello, C.: A new Proposal to hybridize the Nelder-Mead Method to a differential evolution algorithm for constrained optimization. In: *Evolutionary Computation, CEC'09, IEEE Congress on* (2009)
11. Zapotecas, S., Coello, C.: A multi-objective Meta-model assisted memetic algorithm with non gradient-based local search. In: *GECCO'10, Portland, Oregon, USA* (2010)
12. Chen, B., Zeng, W., Lin, U., Zhang, D.: A New Local Search-Based Multiobjective Optimization Algorithm. *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, pp. 50–73 (2015)
13. Ann Shim, V., Chen Tan, K., Tang, H.: Adaptive Memetic Computing for Evolutionary Multiobjective Optimization. *Cybernetics, IEEE Transactions on*, vol. 45, no. 4, pp. 610–621 (2015)
14. Deb, K.: *Optimization for Engineering Design, Algorithms and Examples*. Prentice-Hall, India (1995)
15. Hooke, R., Jeeves, T.: Direct search, solution of numerical and statistical problems. *J. ACM*, vol. 8, no. 2, pp. 212–229 (1961)
16. Mezura, E., Coello, C.: A Simple Multimembered Evolution Strategy to Solve Constrained Optimization Problems. In: *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 1, pp. 1–17 (2005)
17. Runarsson, T., Yao, X.: Stochastic Ranking for Constrained Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294 (2000)

**Tabla 2.** Comparación de variantes de *ED*: *DE/rand/1/bin* y *DE/best/1/bin* sin el método propuesto contra variantes *ED*: *DE/rand/1/bin* y *DE/best/1/bin* con el método de búsqueda propuesto. *MLS* significa (*ModifiedLocalSearch*), *B* es el mejor, *M* es la media, *W* es el peor, *SD* es la desviación estándar, *PE* es el promedio de evaluaciones.

Óptimo	DE-rand/1/bin	DE-rand/1/bin (MLS)	DE-best/1/bin	DE-best/1/bin (MLS)
<b>g01</b> <b>-15.000</b>	B: -13.799804 M: -13.365909 W: -12.414840 SD: 0.606441 PE: 220000	<b>B:-15.000</b> M:-14.9999 W: -14.999997 SD: 1E-6 PE: 165574.67	<b>B:-15.000</b> M:-14.352087 W: -12.814961 SD:0.053651 PE:207069.32	<b>B:-15.000</b> M:-14.99999 W: -14.99999 SD: 5E-6 <b>PE: 110638.708</b>
<b>g02</b> <b>-.8036191</b>	B: -.7952141 M: -.7845126 W: -.775135 SD: .3E-4 PE: 220000	B: -.7997510 M: -.795364 W: -.782245 SD: .5E-5 PE: 220000	B: -.8012429 M: -.7824035 W: -.726512 SD: .03481 PE: 220000	B: -.8032455 M: -.8015653 W: -.7920110 SD: .0056 PE: 220000
<b>g04</b> <b>-30665.5386</b>	B: -30659.5210 M: -30653.3678 W: -30650.3250 SD: 4.5864 PE: 220000	B: -30663.2145 M: -30655.2752 W: -30653.558 SD: 5.2926 PE: 220000	B: -30662.5412 M: -30660.5547 W: -30655.3690 SD: 3.6074 PE: 220000	B: -30664.4368 M: -30662.1547 W: -30660.5498 <b>SD: 2.0016</b> PE: 220000
<b>g06</b> <b>-6961.81387</b>	B: -6068.3124 M: -5732.3644 W: -5521.9127 SD: 275.5913 PE: 220000	<b>B: -6961.81387</b> M: -6325.3249 W: -5783.1378 SD: 589.3315 <b>PE: 219989.6584</b>	B: -6038.00426 M: -5953.3654 W: -5682.9346 SD: 185.8932 PE: 220000	B: -6573.4572 M: -6181.7630 W: -5848.7244 SD: 362.6028 PE: 220000
<b>g07</b> <b>24.3062</b>	B: 33.7865 M: 34.2301 W: 34.6255 SD: 0.4197 PE: 220000	B: 33.2865 M: 33.6777 W: 34.3665 SD: 0.5467 PE: 220000	B: 33.8963 M: 33.2665 W: 34.7456 SD: 0.7422 PE: 220000	B: 32.9634 M: 33.4112 W: 34.3512 SD: 0.7082 PE: 220000
<b>g08</b> <b>-0.09582504</b>	B: -0.0904421 M: -0.09445202 W: -0.0726796 SD: 0.0115 PE: 220000	B: -0.09582503 M: -0.09582250 W: -0.09572035 SD: 5E-5 PE: 220000	B: -0.0955922 M: -0.0953254 W: -0.0949350 SD: 3E-4 PE: 220000	<b>B: -0.09582504</b> M: -0.0957852 W: -0.0955210 SD: 1E-4 <b>PE: 219923.265</b>
<b>g09</b> <b>680.63005</b>	B: 681.502212 M: 681.736654 W: 681.874212 SD: 0.1880 PE: 220000	B: 681.168569 M: 681.39645 W: 681.46584 SD: 0.1555 PE: 220000	B: 681.079512 M: 681.23487 W: 681.632254 SD: 0.285 PE: 220000	<b>B: 680.63005</b> M: 680.98856 W: 681.08658 SD: 0.2403 <b>PE: 219956.015</b>
<b>g10</b> <b>7049.248020</b>	B: 7210.65585 M: 7246.5684 W: 7259.22447 SD: 25.1952 PE: 220000	B: 7125.32963 M: 7169.54523 W: 7256.56478 SD: 66.7708 PE: 220000	B: 7217.56321 M: 681.23487 W: 7232.21001 SD: 34.9738 PE: 220000	B: 7163.325122 M: 7178.563324 W: 7210.25485 SD: 23.9407 PE: 220000

**Tabla 3.** Comparación de nuestros mejores resultados con los trabajos relacionados con variantes de *ED*, *MDE* y *DECV* sin el método propuesto, contra variantes de *ED*, *MDE* y *DECV* con el método de búsqueda propuesto. *MLS* significa (*ModifiedLocalSearch*), *B* es el mejor, *M* es la media, *W* es el peor, *SD* es la desviación estándar, *PE* es el promedio de evaluaciones.

Óptimo	MDE	MDE (MLS)	DECV	DECV (MLS)
<b>g01</b> <b>-15.000</b>	<b>B: -15.000</b> <b>M: -15.000</b> <b>W: -15.000</b> <b>SD: 0</b> PE: 200819.306	<b>B: -15.000</b> <b>M: -15.000</b> <b>W: -15.000</b> <b>SD:0</b> PE: 179356.675	<b>B: -15.000</b> M:-14.352087 W: -12.814961 SD:0.053651 PE: 208596.124	<b>B: -15.000</b> <b>M: -15.000</b> W: -14.99999 SD: 5E-10 <b>PE: 156814.301</b>
<b>g02</b> <b>-.8036191</b>	B: -.7964518 M: -.791150 W: -.7851235 SD: .3E-5 PE: 220000	B: -.8034857 M: -.8025412 W: -.8015456 SD: 0.009108 PE: 220000	B: -.7955210 M: -.7845223 W: -.77619085 SD: .0845 PE: 220000	B: -.80154552 M: -.7999514 W: -.7922214 <b>SD: 7E-8</b> PE: 220000
<b>g04</b> <b>-30665.5386</b>	B: -30633.3541 M: -30621.3547 W: -30583.3508 SD: 26.104 PE: 220000	<b>B: -30665.5386</b> M: -.30642.2457 W: -30635.9288 SD: 13.272 <b>PE: 219.669</b>	B: -30638.2133 M: -30627.3641 W: -30615.3665 SD:11.5053 PE: 220000	B: -30663.2154 M: -30645.368 W: -30638.3545 SD:12.8983 PE: 220000
<b>g06</b> <b>-6961.81387</b>	B: -6228.8601 M: -5763.1574 W: -5322.3407 SD: 452.7075 PE: 220000	B: -6558.4572 M: -5836.7596 W: -5543.8102 SD: 522.1340 220000	B: -6018.2014 M: -5953.3654 W: -5472.5421 SD: 274.2695 PE: 220000	B: -6032.5844 M: -5925.3564 W: -5793.2654 <b>SD: 119.4095</b> PE: 220000
<b>g07</b> <b>24.3062</b>	B: 33.6332 M: 34.0253 W: 34.2557 SD: 0.3147 PE: 220000	B: 28.3687 M: 33.6366 W: 34.1954 SD: 3.2148 PE: 220000	B: 33.5632 M: 33.7445 W: 33.9654 <b>SD: 0.2014</b> PE: 220000	B: 33.2765 M: 33.7963 W: 34.5221 SD: 0.6256 PE: 220000
<b>g08</b> <b>-0.09582504</b>	B: -0.09582503 M: -0.09582501 W: -0.09581520 SD: 5 E -6 PE: 220000	<b>B: -0.09582504</b> M: -0.09582502 W: -0.09582501 <b>SD: 1 E-8</b> <b>PE: 219967.124</b>	B: -0.09582503 M: -0.09582369 W: -0.09575301 SD: 4E-5 PE: 220000	B: -0.09582501 M: -0.0956724 W: -0.09436584 SD: 8 E-4 PE: 220000
<b>g09</b> <b>680.63005</b>	B:680.949523 M: 681.25548 W: 681.23542 SD:0.1711 PE: 220000	B: 680.88060 M: 680.99542 W: 681.09951 SD: 0.1094 PE: 220000	B: 680.999535 M: 681.00953 W: 681.116574 <b>SD: 0.0648</b> PE: 220000	B: 681.003254 M: 681.15464 W: W: 681.65217 SD: 0.3394 PE: 220000
<b>g10</b> <b>7049.248020</b>	B:7265.33478 M:7272.76654 W:7259.22447 SD:16.7426 PE: 220000	B:7122.48525 M: 7189.6542 W: 7201.1745 SD:42.4978 PE: 220000	B:7277.35665 M:7283.12589 W:7296.54752 SD: 9.8464 PE: 220000	B:B:7284.34526 M:7288.45221 W:72.96.54244 <b>SD: 2.9040</b> PE: 220000

**Tabla 4.** Comparación de nuestros mejores resultados contra *MBFOA – AS – LS*.

Óptimo	MBFOA-AS-LS [9]	Mejores resultados (MLS)
<b>g01</b> <b>-15.000</b>	<b>B: -15.000</b> M: -14.685 W: -12.833 SD: 7.2E-01	<b>B: -15.000</b> <b>M: -15.000</b> <b>W: -15.000</b> <b>SD: 0</b> <i>MDE(MLS)</i>
<b>g02</b> <b>-.8036191</b>	B: -.7925284 M: -0.62220 W: -12.833 SD: 1.09E-1	B: -.8034857 M: -.8025412 W: -.8015456 <b>SD: .009108</b> <i>MDE(MLS)</i>
<b>g04</b> <b>-30665.5386</b>	B: -30665.539 M: -30665.539 W: -30665.539 <b>SD: 1.48E-11</b>	<b>B: -30665.5386</b> M: -.30642.2457 W: -.8015456 SD: 13.272 <i>MDE(MLS)</i>
<b>g06</b> <b>-6961.81387</b>	<b>B: -6961.814</b> <b>M: -6961.814</b> <b>W: -6961.814</b> <b>SD: 3.13E-6</b>	B: -6961.81387 M: -6325.3249 W: -.8015456 SD: 589.3315 <i>DErand/1/bin(MLS)</i>
<b>g07</b> <b>24.3062</b>	B: 24.349 M: 24.461 W: 24.623 <b>SD: 8.14E-2</b>	B: 28.3687 M: 33.6366 W: -.8015456 SD: 3.2148 <i>MDE(MLS)</i>
<b>g08</b> <b>-0.09582504</b>	B: -0.095825 M: -0.095825 W: -0.095825 <b>SD: 4.23E-17</b>	<b>B: -0.09582504</b> M: -0.0957852 W: -0.0955210 SD: 1E-4 <i>DEbest/1/bin(MLS)</i>
<b>g09</b> <b>680.63005</b>	B: 680.633 M: 680.690 W: 680.837 <b>SD: 6.61E-02</b>	<b>B: 680.63005</b> M: 680.98856 W: 681.08658 SD: 0.2403 <i>DEbest/1/bin(MLS)</i>
<b>g10</b> <b>7049.248020</b>	B: 7051.648 M: 7077.555 W: 7142.653 <b>SD: 2.5E-1</b>	B: 7122.48525 M: 7189.6542 W: 7201.1745 SD: 42.4978 <i>MDE(MLS)</i>