Un estudio de operadores de variación en algoritmos inspirados en la naturaleza para optimización restringida

Como requerimiento para obtener el grado de Doctora en Inteligencia Artificial

Adriana Cervantes Castillo

Universidad Veracruzana
Centro de Investigación en Inteligencia Artificial

Diciembre 2018

Documento maquetado con TEXIS v.1.0+.

Un estudio de operadores de variación en algoritmos inspirados en la naturaleza para optimización restringida

Como requerimiento para obtener el grado de
Doctora en Inteligencia Artificial

Dirigida por el Doctor
Efrén Mezura Montes

Universidad Veracruzana
Centro de Investigación en Inteligencia Artificial

Diciembre 2018

Agradecimientos

He aquí, tu amas la verdad en lo íntimo,
y en lo secreto me has hecho comprender
sabiduría.

Salmos 51:6

El término de este trabajo no hubiera sido posible sin el apoyo recibido por todas las personas que en el momento oportuno estuvieron siempre dispuestas para mi. Es por esto que agradezco a mi Dios por haberme puesto en el lugar y el momento indicado para que esto sucediera. Agradezco a mis padres los cuales directa e indirectamente se preocuparon en darme la educación necesaria, a mi madre en particular por su constancia e insistencia en todo el tiempo que duró esto. Agradezco a mi esposo porque siempre fue paciente y supo esperar y apoyarme hasta el último momento. Agradezco a mi director de tesis, sin él este trabajo no hubiera sido posible. Siempre dispuesto para brindarme asesoría, para escuchar y resolver mis dudas. Deseo bendiciones y éxito para él y para toda su familia. Agradezco a mis profesores su entrega y conocimiento brindados en cada clase. Agradezco a mis compañeros de clase grandes amigos dispuestos a brindarme su apoyo cuando los necesité. Agradezco a mi institución y a Conacyt por el apoyo brindado durante estos cuatro años. Hay mucho y a muchos que agradecer por finalizar este trabajo, pero agradezco a mi Dios que hizo posible en mi esto que hoy presento. Un año de bendiciones y entre ella mi pequeña hija Aurora Hefzi-ba que será desde hoy en adelante mi inspiración para esforzarme y ofrecerle lo mejor de mi, mis éxitos y mis bendiciones son desde hoy por ella y para ella.

Resumen

Y dijo al hombre: He aquí que el temor
del Señor es la sabiduría, y el apartarse
del mal, la inteligencia.

Job 28:28

En problemas de optimización numérica con restricciones, referidos en este trabajo, el enfoque de optimización difiere del utilizado en optimización sin restricciones. De entrada, la presencia de restricciones divide el espacio de búsqueda en dos regiones: la región factible y la región no factible, siendo las soluciones localizadas en el área factible las de interés en este tipo de problemas. Por otro lado, si al evaluar la restricción su valor es cero, se dice que, ésta es una restricción activa, la cual, de acuerdo a la literatura especializada, dificultará el proceso de búsqueda al aumentar la concentración de soluciones en la frontera de la región factible, un espacio difícil de explorar por un método de búsqueda global. Esas condiciones posibilitan la introducción de métodos especiales que permiten guiar la búsqueda hacia las dos zonas de interés en problemas de optimización restringida: el borde de la región factible y el interior de la región factible.

En este trabajo se emplean Algoritmos Inspirados en la Naturaleza (NIAs, por sus siglas en inglés) reforzados con operadores especiales de frontera y de factibilidad que permitirán concentrar la búsqueda en las dos regiones de interés en optimización restringida. Como algoritmo global de búsqueda se emplea el algoritmo de Inteligencia Colectiva inspirado en el proceso de tormenta de ideas (BSO, por sus siglas en inglés *Brain Storm Optimization*)reforzado con dos operadores especiales. En primer lugar, BSO es reforzado con el operador especial de frontera BS (por sus siglas en ingles *Boundary Search*) utilizado para guiar la búsqueda hacia la frontera de la región factible. Por otra parte, el método de consenso de restricciones (CC, por sus siglas en inglés *Constraint Consensus*) es agregado al algoritmo para alcanzar con rapidez la zona factible y reducir el número de evaluaciones utilizadas en el proceso. Inspirados en estos dos operadores especiales, dos nuevos operadores de frontera son propuestos en este trabajo: el primero (BS-r), basa su comportamiento en el tópico de la división de un segmento

de línea propuesto en Álgebra Lineal; el segundo (BS-G), trata de alcanzar la frontera de las restricciones a través de la información proporcionada por el gradiente, obtenido éste de forma numérica, a partir de la restricción con mayor grado de violación. Para guiar la búsqueda hacia la región factible se propone una variante al método CC denominada R+V (Restricción mas Violada), mediante la cual se intenta alcanzar con rapidez la zona factible calculando un único vector de factibilidad para la restricción con mayor grado de violación. La nueva variante R+V logra resultados similares a otras variantes existentes con la ventaja de reducir el costo computacional calculando el gradiente sólo para una restricción sin importar el número de restricciones violadas en el punto actual.

Al final se tiene un nuevo algoritmo reforzado con operadores especiales mostrando resultados competitivos en problemas de optimización con restricciones comparado con algoritmos representativos del estado del arte y enfatizando un ahorro computacional medido por el número de evaluaciones. Además de contribuir al área con dos nuevos operadores especiales de frontera y un nuevo operador especial de factibilidad, los cuales pueden ser integrados a otros algoritmos del estado del arte sin necesidad de cambios adicionales.

Índice

Índice de figuras

Índice de Tablas

Capítulo 1

Introducción

Un término cada vez más común en el área de optimización es el referido a optimización restringida y aunque existe un gran número de métodos clásicos que pueden ser empleados para dar solución a esta problemática, en los últimos años los Algoritmos Inspirados en la Naturaleza (*Nature Inspired Algorithms*, NIA) se han convertido en una alternativa muy utilizada para este fin logrando resultados alentadores y competitivos en el área [1], [2], [13], [15].

El enfoque utilizado al resolver problemas restringidos difiere del utilizado en espacios sin restricciones. En primer lugar, ahora el espacio de búsqueda se encuentra dividido principalmente en dos zonas: la zona factible y la zona no factible, siendo la zona factible la región de interés en optimización restringida. Para todo método o algoritmo de optimización empleado en resolver problemas de optimización con restricciones, el primer reto es encontrar la zona factible para después mantenerse dentro de ésta y encontrar el óptimo global localizado en alguna parte de la misma [11] [34]. Dependiendo del tipo de problema a optimizar, la zona factible puede ser muy pequeña en comparación con la zona no factible dentro del espacio de búsqueda, además de existir la posibilidad de tratarse de una zona factible disjunta.

Otro reto a vencer en optimización restringida es el tipo de restricciones asociadas al problema, que pueden generalizarse en dos tipos: restricciones de igualdad y restricciones de desigualdad. A su vez, una restricción de desigualdad puede clasificarse en dos tipos, en el primero, cuando al evaluar la restricción en el punto actual el valor de la restricción es positivo, se dice que es una restricción inactiva que se satisface dentro de la zona factible. En contraste, si al evaluar la restricción en el punto actual el valor de la restricción es cero, se le conoce como una restricción activa la cual se encuentra en el límite entre la región factible y la región no factible. Por definición, a toda restricción de igualdad se le considera como restricción activa [15]. Además, está documentado en la literatura especializada [41], [43], [55], que la presencia de restricciones activas agrega mayor dificultad al proceso de

optimización, ya que es muy probable que el óptimo global se localice justo en la frontera de la región factible, un lugar no fácil de explorar por cualquier método de búsqueda global sea a través de programación matemática o meta-heurística.

Por todo lo anterior, se deriva que en optimización restringida el óptimo global de una función puede localizarse ya sea dentro de la zona factible o en la frontera de ésta, convirtiendo el proceso de optimización en una tarea no trivial para cualquier método utilizado en el proceso y abriendo la posibilidad del surgimiento de métodos especiales que ayuden a guiar la búsqueda global hacia las zonas de interés.

Los operadores especiales son métodos que permiten posicionarse en una región de interés dentro del espacio de búsqueda. De esta manera, en combinación con un algoritmo de búsqueda global se espera obtener algoritmos robustos ofreciendo resultados competitivos con respecto a los obtenidos por algoritmos del estado del arte al enfrentar exitosamente problemas de optimización con restricciones.

1.1. Antecedentes

Distintos algoritmos inspirados en la naturaleza han mostrado resultados exitosos cuando se enfrentan a problemas restringidos [42], [56], [60]. Algunos de ellos han sido reforzados con operadores especiales mostrando un mejor desempeño en los problemas donde han sido probados. Por ejemplo, en 1996 Z. Michalewicz y M. Schoenauer utilizaron un Algoritmo Genético junto con dos operadores especiales para explorar la frontera de la región factible de dos problemas en particular [45]. Basados en la idea de Michalewicz y Schoenauer, en 2006 G. Leguizamón y C. A. Coello-Coello presentaron un operador de frontera en combinación con un algoritmo de Inteligencia Colectiva de optimización basado en colonia de hormigas (Ant Colony Optimization, ACO) para optimizar el conjunto de 24 problemas presentados en el Congreso de Cómputo Evolutivo CEC2006 [38]. En el 2014, M. R. Bonyadi and Z. Michalewicz [3] presentaron un método para guiar la búsqueda hacia la frontera de la región factible reduciendo el espacio de búsqueda mediante un factor de relajación controlado por el usuario. El algoritmo de Inteligencia Colectiva PSO es utilizado como algoritmo global de búsqueda. Por otro lado, el algoritmo evolutivo de Evolución Diferencial es utilizado desde 2011 hasta 2016 por Hamza [31] en combinación con el método de consenso de restricciones con el objetivo de alcanzar con rapidez la zona factible y mejorar el rendimiento final del algoritmo. Como se puede observar, la literatura especializada muestra que la combinación de operadores especiales con Algoritmos Inspirados en la Naturaleza puede mejorar el rendimiento global del algoritmo. Cabe destacar que, de inicio, estos algoritmos por sí mismos eran ya exitosos en espacios restringidos.

En el 2011 surge un nuevo algoritmo de Inteligencia Colectiva inspirado en el proceso de tormenta de ideas (Brainstorming: proceso utilizado por el ser humano para generar ideas que den solución a problemas que difícilmente se resolverían de forma individual). El algoritmo es conocido como Algoritmo de Optimización basado en tormenta de Ideas (*Brain Storm Optimization algorithm*, BSO) [62]. Desde su aparición, BSO ha atraído la atención de varios investigadores y ha sido exitosamente utilizado para resolver problemas sin restricciones [9], sin tener conocimiento hasta el momento, de si el éxito presentado por BSO en espacios sin restricciones puede replicarse en espacios restringidos.

1.2. Justificación

El éxito reportado por el algoritmo BSO en espacios sin restricciones y el desempeño mostrado por los operadores especiales aplicados a algoritmos bioinspirados en optimización restringida ofrecen la posibilidad de adaptar este nuevo algoritmo a espacios restringidos e intentar mejorar su desempeño a través de operadores especiales.

1.3. Definición del problema

Con el objetivo de reforzar el área en optimización restringida y operadores especiales, en este trabajo se analiza el desempeño del algoritmo BSO en espacios restringidos, reforzándolo además con dos operadores especiales. Por un lado se refuerza al algoritmo con operadores especiales de frontera (Boundary Search, BS, [38]) guiando su búsqueda hacia la frontera de la región factible esperando encontrar soluciones factibles aún cuando se trate de restricciones activas. Por otro lado, se intenta acelerar la llegada a la región factible utilizando el método de consenso de restricciones (*Constraint Consensus*, CC) propuesto en [10], de esta manera, el algoritmo concentrará su búsqueda dentro o muy cerca de la región factible de interés tratando de mejorar los resultados y gastando un número menor de evaluaciones en el proceso.

1.4. Hipótesis

La incorporación de dos operadores especiales, búsqueda de frontera (BS) y consenso de restricciones (CC), al algoritmo BSO, hará de éste un algoritmo robusto en alcanzar la zona factible y competitivo contra algoritmos del estado del arte para resolver problemas de optimización numérica con restricciones.

1.5. Objetivos

1.5.1. Objetivo Principal

Enriquecer el estado del arte en el área de optimización numérica restringida agregando el algoritmo basado en tormenta de ideas BSO reforzado con operadores especiales, mostrándolo como un algoritmo capaz de obtener resultados competitivos con otros algoritmos del estado del arte y enfatizando un ahorro en el número de evaluaciones.

1.5.2. Objetivos específicos

- Analizar el desempeño del algoritmo BSO en espacios restringidos mediante un comparativo entre las primeras versiones surgidas del algoritmo BSO con diferentes técnicas para el manejo de restricciones.
- Analizar el desempeño del algoritmo BSO reforzado con operadores especiales de frontera comparando los resultados finales contra la versión del algoritmo sin operador.
- Contribuir al área de operadores especiales de frontera con dos nuevos operadores, incluyendo ventajas y desventajas de uso en problemas restringidos.
- Contribuir al área de operadores especiales de factibilidad con una nueva variante al método de consenso de restricciones con base en la restricción más violada.
- Comparar el desempeño del algoritmo BSO reforzado con operadores especiales contra algoritmos representativos del estado del arte. Aplicando medidas de tendencia central y dispersión a los resultados obtenidos en la función objetivo.

1.6. Contribuciones

- Un nuevo algoritmo basado en el proceso de tormenta de ideas reforzado con operadores especiales capaz de obtener resultados competitivos comparados con algoritmos del estado del arte para resolver problemas de optimización numérica con restricciones.
- Dos nuevos operadores de frontera: el primero, (BS-r), basado en el concepto de división de un segmento de línea de Álgebra lineal y el segundo, (BS-G), basado en la proyección del gradiente eliminando la condición de entrada de puntos factibles necesaria en BS.

- Una nueva variante del método de CC basada en la restricción más violada. La cual permitirá obtener resultados competitivos con las variantes de CC existentes, enfatizando un ahorro de costo computacional al calcular un solo gradiente y sin la necesidad de métodos para generar vectores consenso.

1.7. Publicaciones

Los productos obtenidos durante el desarrollo de esta tesis son mencionados en este apartado.

1.7.1. Conferencias Internacionales

- A. Cervantes-Castillo and E. Mezura-Montes. A study of constraint-handling techniques in brain storm optimization, in Proceedings of 2016 IEEE Congress on Evolutionary Computation (CEC 2016). Vancouver, Canada: IEEE, 2016, pp. 3740-3746.

- A. Cervantes-Castillo, G. Leguizamón, E. Mezura-Montes. Exploring boundary constraints using the modified brain storm optimization algorithm and a boundary search operator: MBSO-BS. IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), 2017, 1-6.

1.7.2. Artículo de revista

A. Cervantes-Castillo and E. Mezura-Montes. A Modified Brain Storm Optimization Algorithm with a Special Operator to Solve Constrained Optimization Problems, International Journal of Computational Intelligence Systems, 2018. SUBMITTED, SCI Impact factor 2.0.

1.8. Estructura del documento

La estructura de este documento se encuentra dividida en 7 capítulos, de los cuales, los primeros tres describen los conceptos básicos que sirven de base para entender la dirección y los aportes de este trabajo descritos en los capítulos restantes.

En el Capítulo 1 se presenta la introducción y definición del problema, así como la hipótesis, los objetivos, contribuciones y publicaciones obtenidas del trabajo de tesis. En el Capítulo 2 se presentan los conceptos básicos de optimización con y sin restricciones, así como los métodos clásicos utilizados en optimización restringida. Posteriormente, como una alternativa a los métodos clásicos para resolver problemas de optimización, en el Capítulo 3

se describen los conceptos básicos de un Algoritmo Evolutivo y sus principales paradigmas. Además, se describe el concepto de Inteligencia Colectiva y se finaliza mencionando algunas técnicas utilizadas en el manejo de restricciones. Una vez que se conocen las alternativas a los métodos clásicos de optimización, en el Capítulo 4 se analiza el desempeño de un algoritmo de inteligencia colectiva conocido como BSO el cual es inspirado en el concepto de tormenta de ideas. En este Capítulo se presenta un comparativo de técnicas para el manejo de restricciones en tres versiones del algoritmo con el objetivo de obtener la mejor versión en problemas de optimización con restricciones y ser utilizado como algoritmo global de búsqueda en capítulos posteriores. Por otra parte, dado que en optimización restringida los óptimos a las funciones pueden localizarse dentro de la zona factible o en la frontera de ésta, en los siguientes dos capítulos se centra la búsqueda en estas regiones de interés mediante la aplicación de operadores especiales. En primer lugar, en el Capítulo 5 se analiza el desempeño del algoritmo BSO reforzado con operadores especiales de frontera. De esta manera, además de una búsqueda global, con los operadores especiales la búsqueda se dirigirá hacia la frontera de las restricciones esperando localizar puntos óptimos que difícilmente pueden ser localizados por un algoritmo de búsqueda global. Se proponen dos nuevos operadores de frontera: uno basado en el concepto de la división de un segmento de linea de Álgebra Lineal y uno más basado en la proyección del gradiente. Por otra parte, en el Capítulo 6 se describe el desempeño del algoritmo BSO reforzado con una nueva variante del método de consenso de restricciones conocida como R+V. Al igual que en las variantes existentes, la variante R+V intenta alcanzar con rapidez la región factible, la única diferencia es que lo hace tomando en cuenta la restricción con mayor grado de violación y no el consenso de todas las restricciones violadas, enfatizando así, un ahorro computacional al calcular el gradiente sólo para una restricción y no para todas las restricciones violadas en el punto actual. Se finaliza en el Capítulo 7 con las conclusiones y trabajos futuros al trabajo aquí presentado.

Capítulo 2

Optimización

En este capítulo, el concepto de optimización hace referencia a métodos de búsqueda aplicados para obtener la mejor solución a un problema propuesto, el cual es aplicado en distintas áreas de la vida real como lo es la ingeniería, las matemáticas, la minería de datos, entre otras [1], [14], [15].

De acuerdo con Deb en [15], un problema de optimización necesita ser reflejado a través de un modelo matemático perfectamente diseñado. El modelo matemático consistirá entonces, en elegir las variables de diseño o variables de decisión a optimizar, formular las restricciones asociadas al problema y establecer una función objetivo en términos de las variables definidas, la cual se resolverá mediante la optimización del vector conformado por las variables de decisión que minimizan o maximizan la función objetivo en cuestión.

2.1. Conceptos de optimización

Los conceptos generales de optimización son descritos en este apartado. Se definirá un problema de optimización y sus elementos, además de los conceptos de mínimos locales y globales en una función.

2.1.1. Definición del problema

Sin pérdida de generalidad, un problema de optimización sin restricciones se define en la Ecuación 2.1 y consiste en:
Encontrar el vector $\vec{x}$

$$\vec{x} = x_1, x_2, ...x_D \tag{2.1}$$

que minimice

$$f(\vec{x})$$

donde x_d, es el vector conformado por las variables de decisión del problema, $d = 1, \ldots, D$ se encuentra delimitado por un límite inferior L_d y un límite

superior U_d, $L_d \leq x_d \leq U_d$.

2.1.2. Mínimo Global

En la minimización de funciones un mínimo global es el mejor valor conocido en la función objetivo. Dentro del espacio de búsqueda es el punto $\vec{x}$ que minimiza el valor de $f(\vec{x})$, también conocido como punto óptimo de la función.

2.1.3. Mínimo Local

Un mínimo local se define como el mejor punto encontrado relativo a una vecindad específica en la función que rodea este punto. En ocasiones un óptimo local puede ser también el óptimo global [15].

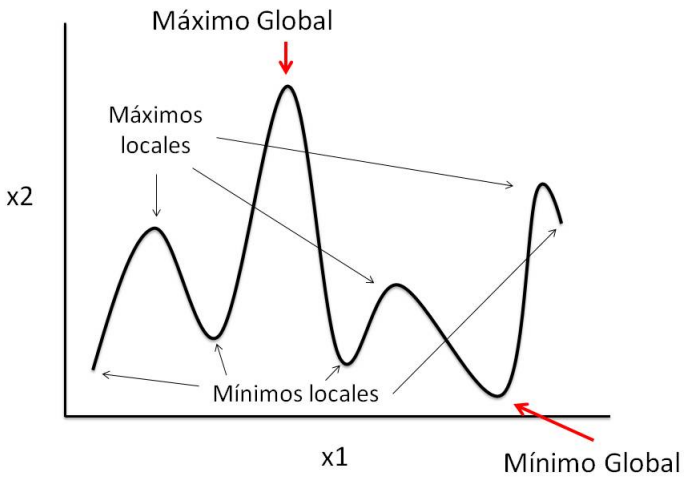

Figura 2.1: Máximos y mínimos en una función.

Cabe destacar que dependiendo de si el problema a optimizar tiene o no restricciones asociadas, la optimización puede ser clasificada en dos tipos: Optimización no restringida y optimización restringida. En este trabajo nos centraremos en la minimización de funciones mono-objetivo sujetas a restricciones.

2.2. Optimización con Restricciones

En Problemas de Optimización Numérica con Restricciones, CNOP,(por sus siglas en inglés) el enfoque de optimización difiere del enfoque no restrin-

gido, ya que ahora es necesario satisfacer en primer lugar las restricciones del problema y en un segundo paso optimizar la función objetivo. Este tipo de problemas divide en dos áreas principales el espacio de búsqueda: el área factible y el área no factible. Siendo muchas veces, la mayor dificultad en un problema con restricciones localizar el área factible y más aún mantenerse dentro de ésta, ya que esta zona puede ser muy pequeña en comparación con la región no factible y en ocasiones estar disjunta dentro del espacio de búsqueda constituyendo un reto para todo algoritmo asociado con este tipo de problemas.

Matemáticamente, un problema de optimización con restricciones puede ser definido como sigue: Encontrar

$$f(\vec{x}) \tag{2.2}$$

Sujeto a:

$$g_i(\vec{x}) \leq 0; i = 1, 2, \ldots, m$$

$$h_j(\vec{x}) = 0; j = 1, 2, \ldots, p$$

donde $f(\vec{x})$ representa la función objetivo, $g_i(\vec{x})$, $i = 1, \ldots, m$ y $h_j(\vec{x})$, $j = 1, \ldots, p$ son el conjunto de restricciones de desigualdad e igualdad, respectivamente, $\vec{x} = [x_1, x_2, \ldots, x_D]$ es el vector de variables de decisión del problema, donde cada x_d, $d = 1, \ldots, D$ es delimitado por un limite inferior L_d y un limite superior U_d, $L_d \leq x_d \leq U_d$. Los límites en las variables, definidas por el usuario, definen el espacio de búsqueda S y la región factible $F \subseteq S$ es definida por aquellas soluciones que satisfacen el conjunto de restricciones tanto de igualdad como de desigualdad.

De acuerdo a la definición anterior, un punto o solución se define como factible si tanto las restricciones de igualdad, desigualdad y de límite son satisfechas por dicho punto. Este punto se considera óptimo si además de lo anterior también es un mínimo global. Cualquier otro punto que no cumpla con esta condición se conoce como no factible y por lo tanto nunca será un punto óptimo.

De acuerdo a la definición del problema, hay dos maneras de satisfacer las restricciones de desigualdad, ya sea que el punto se encuentre sobre la superficie (o límite) de la restricción $g_i(\vec{x}) = 0$. En este caso se dice que la restricción es activa en este punto. Por otro lado, si el punto se encuentra sobre el lado factible de la superficie de la restricción, donde la restricción es positiva, entonces se dice que la restricción es inactiva en este punto. Por definición, todas las restricciones de igualdad son activas. Dada esta situación es muy común que el óptimo de la función se encuentre ya sea en el límite

de la región factible o dentro de ésta, por lo tanto, concentrar la búsqueda en estas dos regiones es vital para poder alcanzar los resultados deseados.

2.3. Algoritmos clásicos en optimización restringida

Con referencia en el libro de Deb [15], en este apartado se clasificará a los algoritmos utilizados en optimización restringida en dos tipos: métodos directos y métodos basados en gradiente.

2.4. Métodos directos

Como parte de los métodos directos, en este apartado se menciona a la penalización de funciones y el método de Complex.

2.4.1. Penalización de funciones

La penalización de funciones es un método de transformación, en el cual un problema con restricciones se convierte en un problema sin restricciones con sólo agregar factores de penalización a las restricciones violadas como se muestra en la Ecuación 2.3.

$$P(\vec{x}, R) = f(\vec{x}) + \Omega(R, g(\vec{x}), h(\vec{x})) \tag{2.3}$$

donde $f(\vec{x})$ es la función a optimizar, R es un conjunto de parámetros de penalización, Ω es el término de penalización (interior o exterior) y por último g, h son las restricciones de desigualdad e igualdad respectivamente. En el algoritmo la condición de paro puede ser un número máximo de iteraciones, un número máximo de evaluaciones o un valor determinado de la función objetivo.

De acuerdo en como se aplica el término de penalización Ω, en la literatura se habla de dos tipos de penalización de funciones:
Penalización interior que consiste en penalizar aquellas soluciones factibles que se encuentran muy cerca de la frontera de la restricción. Por lo tanto puntos factibles son requeridos en cada iteración del algoritmo. Estos puntos pueden obtenerse después de optimizar la Ecuación 2.3 utilizando cualquier método de optimización sin restricciones.

Penalización exterior. Consiste en penalizar soluciones no factibles. El método no requiere de soluciones factibles ya que el punto no factible se mejora en cada iteración del algoritmo.

El valor del factor de penalización varía de acuerdo al grado de violación de la restricción actual, por lo tanto, entre mayor sea la violación de la restricción, mayor será el valor que tomará el factor de penalización.

De acuerdo al tipo de restricción es la forma en que se elige el factor de penalización. Por ejemplo para restricciones de igualdad, dicho factor se calcula de acuerdo a la Ecuación 2.4

$$\Omega = R\{h(\vec{x})\}^2 \tag{2.4}$$

Como se puede observar cualquier punto no factible es penalizado con el cuadrado de la violación de la restricción actual. Al penalizar solo puntos no factibles, se está refiriendo a una penalización exterior. Una forma de seleccionar la función de penalización para restricciones de desigualdad es usando la Ecuación 2.5

$$\Omega = R\sum_{i=1}^{m} |g_i(\vec{x})| \tag{2.5}$$

donde, m es el conjunto de todas las restricciones de desigualdad violadas por ese punto. A continuación se describe el procedimiento de penalización de funciones en el Algoritmo 1.

Algoritmo 1 Penalización de funciones

1: Elegir un punto inicial x^t, un valor inicial de Ω, un valor inicial de R^t, un parámetro c que servirá para actualizar el valor de R, $t = 0$;
2: **while** No se cumpla una condición de paro **do**
3: Formar $P(\vec{x^t}, R^t) = f(\vec{x^t}) + \Omega(R^t, g(\vec{x^t}), h(\vec{x^t}))$;
4: Encontrar $\vec{x}^{t+1}$ tal que $P(\vec{x}^{t+1}, R^t)$ es mínimo para un valor fijo de R^t;
5: **if** $|P(\vec{x}^{t+1}, R^t) - P(\vec{x}^t, R^{t-1})| \leq \varepsilon$ **then**
6: $\vec{x}^t = \vec{x}^{t+1}$;
7: **else**
8: $R^{t+1} = cR^t$;
9: $t = t + 1$;
10: **end if**
11: **end while**

Una de las ventajas del método es que puede utilizarse para cualquier tipo de restricciones. Su mayor desventaja es la calibración de los factores de penalización que deben ser definidos por el usuario final.

2.4.2. Método Complex

El método fue propuesto por M. J. Box en 1965 [4] y es muy similar al método simplex utilizado en optimización sin restricciones; la diferencia es que las restricciones son tomadas en cuenta para generar el complex. El algoritmo inicia con un conjunto P de puntos factibles generados de forma aleatoria. Si en el proceso de búsqueda se encuentra que algún punto es no factible éste es movido hacia el centroide del conjunto hasta lograr la

factibilidad del punto. Una vez formado el complex se pasa al proceso de optimización en el cual se elegirá el peor de los puntos para ser reflejado hacia el centroide del conjunto actual y generar un nuevo punto factible. Si el nuevo punto es factible, éste reemplazará al peor del conjunto actual. Los pasos a seguir para implementar el método se encuentran en el Algoritmo 2.

El algoritmo muestra un buen desempeño cuando la región factible es convexa y cuando el óptimo se encuentra dentro de la zona factible, es decir el óptimo no está en la frontera de la zona factible.

Algoritmo 2 Complex

1: $x^L \leq x \leq x^U$; Inicializar α; ε, δ;
2: **while** $i < P$ **do**
3: Generar N números aleatorios en el rango permitido de las variables para generar el punto $\vec{x}_i^p$
4: **if** $\vec{x}_i^p$ es no factible **then**
5: **repeat**
6: calcular el centroide $\bar{x}$ del conjunto P;
7: $\vec{x}_i^p = \vec{x}_i^p + \frac{1}{2}(\bar{x} - \vec{x}_i^p)$;
8: **until** $\vec{x}_i^p$ sea factible
9: **end if**
10: **end while**
11: Evaluar los puntos generados
12: *//Reflexión*
13: Seleccionar $\vec{x}^R = max f(\vec{x}^p)$
14: calcular $\bar{x}$ del conjunto P sin contar $\vec{x}^R$, generar el nuevo punto $\vec{x}^m$;
15: $\vec{x}^m = \bar{x} + \alpha(\bar{x} - \vec{x}^R)$;
16: **if** $\vec{x}_m$ es factible y $f(\vec{x}^m) \geq f(\vec{x}^R)$ **then**
17: **repeat**
18: retraer la mitad de la distancia del centroide $\bar{x}$
19: $\vec{x}^m = \bar{x} + \alpha(\bar{x} - \vec{x}^R)$;
20: **until** $f(\vec{x}^m) < f(\vec{x}^R)$
21: **else**
22: **if** $\vec{x}^m$ es factible y $f(\vec{x}^m) < f(\vec{x}^R)$ **then**
23: $\vec{x}^R = \vec{x}^m$
24: Calcular $\bar{f} = \frac{1}{P}\sum_{i=1}^{P} f(\vec{x}_i^p)$ y $\bar{x} = \frac{1}{P}\sum_{i=1}^{P} \vec{x}_i^p$
25: **if** $\sqrt{\sum_{i=1}^{P}(f(\vec{x}_i^p) - \bar{f})^2} \leq \varepsilon$ y $\sqrt{\sum_{i=1}^{P} \parallel \vec{x}_i^p - \bar{x} \parallel^2} \leq \delta$ **then**
26: Terminar
27: **else**
28: $k = k + 1$, regresar al paso 13
29: **end if**
30: **end if**
31: **else**
32: **if** $\vec{x}^m$ es no factible **then**
33: Verificar que las variables de diseño se encuentren en el rango permitido;
34: **if** $\vec{x}_m$ es no factible **then**
35: **repeat**
36: retraer la mitad de la distancia del centroide $\bar{x}$
37: $\vec{x}^m = \bar{x} + \alpha(\bar{x} - \vec{x}^R)$;
38: **until** $f(\vec{x}^m)$ sea factible
39: Regresar al paso 16
40: **end if**
41: **end if**
42: **end if**

2.5. Métodos basados en gradiente

Los métodos que se describen en esta sección buscan acercarse a la región factible utilizando la información del gradiente tanto de la función objetivo como de las restricciones del problema. Por lo tanto, aunque son métodos en teoría mas rápidos en alcanzar su objetivo, tienen la desventaja de cumplir con la condición de continuidad y ser diferenciales para un mejor resultado.

2.5.1. Método de direcciones factibles

El método consiste en la búsqueda de direcciones locales a través de aproximaciones lineales de la función objetivo y las restricciones asociadas a ésta. Partiendo de un punto factible y una dirección de búsqueda factible se genera un nuevo punto con la finalidad de que éste sea mejor que el actual y más cercano al punto óptimo. El método requiere que se cumplan dos condiciones que son: a) factibilidad (a partir de las restricciones activas se encuentra una dirección de búsqueda) y b) descendencia dada por $\nabla f(\vec{x}^t) \cdot d \leq 0$. Los pasos necesarios para implementar el método se presentan en el Algoritmo 3.

Algoritmo 3 Método basado en direcciones factibles

1: $t = 0, \varepsilon = 10^{-3}$. Elegir un punto factible $\vec{x}^t$, $\theta^t = -\nabla f(\vec{x}^t)$;
2: **while** $\theta^t \geq 0$ **do**
3: $I^t = \{i : 0 \leq g_i(\vec{x}^t) \leq \varepsilon, i = 1, 2, \ldots, m\}$
4: **if** I^t es vacío **then**
5: usar $d^t = \theta^t = -\nabla f(\vec{x}^t)$
6: normalizar d^t
7: **if** $\theta^t \leq 0$ **then**
8: Terminar;
9: **else**
10: $\bar{\alpha} = min[\alpha : g_i(\vec{x}^t + \alpha d^t) = 0, i = 1, 2, \ldots, m, \alpha > 0]$;
11: **end if**
12: **if** no existe $\bar{\alpha} > 0$ **then**
13: $\bar{\alpha} = \infty$;
14: **end if**
15: **end if**
16: Resolver el LP problem presentado en 2.6
17: Etiquetar d^t, θ^t
18: Encontrar α^t tal que: $f(\vec{x}^t + \alpha^t d^t) = min[f(\vec{x}^t + \alpha d^t) : 0 \leq \alpha \leq \bar{\alpha}]$
19: $\vec{x}^{t+1} = \vec{x}^t + \alpha^t d^t$
20: $t = t + 1$
21: **end while**

$$Maximizar \quad \theta \tag{2.6}$$

sujeto a:

$$\nabla f(\vec{x}^t)d \leq -\theta,$$

$$\nabla g_i(\vec{x}^t)d \geq \theta, \quad i \subset I^t$$

$$-1 \leq d_i \leq 1 \quad i = 1, 2, \ldots, N; \quad N = total \quad de \quad variables$$

Dado que sólo restricciones activas son consideradas para determinar las direcciones de búsqueda, el algoritmo puede padecer de convergencia lenta ya que no hace una búsqueda directa en linea recta, sino que el algoritmo zigzaguea en el espacio de búsqueda tratando de llegar a su objetivo.

2.5.2. Método de proyección del gradiente

La idea del algoritmo es partir de un punto no factible que al ser proyectado a la superficie de las restricciones activas, éste se torna un punto factible. En cada iteración se encuentran las restricciones activas, luego, a través del algoritmo de Newton-Rapson (algoritmo utilizado en optimización clásica sin restricciones) se encuentra una dirección de búsqueda para proyectarla sobre dichas restricciones y así obtener una dirección factible de búsqueda. Una vez que se ha encontrado una dirección factible de búsqueda, se continúa con una búsqueda unidireccional para obtener el punto mínimo en dicha dirección. Cuando se trata de restricciones no lineales, la búsqueda unidireccional es modificada con el objetivo de hacer factibles los puntos de búsqueda. En este sentido, cada punto creado es proyectado hacia la superficie de la intersección de las restricciones utilizando el método de interpolación cuadrática para tener información de donde se encuentra el punto óptimo. Los pasos necesarios para implementar este método se describen en el Algoritmo 4.

Algunos inconvenientes del método es que se necesita calcular y evaluar la matriz A, la cual contiene los gradientes de las restricciones activas, en cada iteración, además de que cada punto no factible debe proyectarse sobre la superficie de las restricciones activas, lo que no siempre es una tarea fácil y puede necesitar un gran número de evaluaciones.

2.6. Conclusiones

En este capítulo se presentó un resumen muy general sobre los conceptos básicos de optimización. Se hizo énfasis en la optimización restringida donde el espacio de búsqueda se encuentra dividido en dos áreas principales, el área factible y el área no factible, siendo de interés en este tipo de problemas el área factible, ya que es ahí donde se encontrará el óptimo global de la función. Para lograr este objetivo, se presentaron algunos métodos clásicos

Algoritmo 4 Método de Proyección del gradiente

1: $t = 0, \varepsilon_1 = \varepsilon_2 = 0{,}001$. Elegir un punto inicial $\vec{x}^t$;
2: Evaluar las restricciones de desigualdad e identificar las restricciones activas I^t en el punto $\vec{x}^t$. $I^t = \{i : |g_i(\vec{x}^t)| \leq \varepsilon_1, i = 1, 2, \ldots, m\}$
3: $u_m = min\{u_l : l \subset I^t\}$
4: **while** $u_m > \varepsilon_2$ **do**
5: **if** $t \geq 1$ y $I^t = I^{t-1}$ or I^t es vacío **then**
6: $s = -\nabla f(\vec{x}^t)$
7: **if** $\parallel s^t \parallel > \varepsilon_2$ **then**
8: Determinar el tamaño de paso máximo α_{max} de tal manera que $g_l(w(\alpha)) \geq 0$ para todo $l \not\subset I^t$, donde cualquier punto s^t es representado por $w(\alpha)$;
9: **else**
10: $(v, u) = (AA^T)^{-1}A\nabla f$
11: **if** $|u_m| \leq \varepsilon_2$ **then**
12: Terminar
13: **else**
14: Borrar la restricción de desigualdad m de I^t e ir al paso 4.
15: **end if**
16: **end if**
17: **else**
18: Formar la matriz A cuyas filas son ∇h_k y ∇g_i, donde, $i \subset I^t$
19: Calcular la matriz de proyección $P = I - A^T(AA^T)^{-1}A$
20: $s = -P\nabla f(\vec{x}^t)$
21: **end if**
22: **if** El algoritmo de pasos descendiente(algoritmo de Newton-Raphson) se lleva a cabo **then**
23: Hacer una búsqueda direccional en el rango$(0, \alpha_{max})$ para calcular α^*
24: **else**
25: Concentrar la búsqueda sobre α en el intervalo $(0, \alpha_{max})$ para determinar tres puntos $w(\alpha_1), w(\alpha_2), w(\alpha_3)$. Donde cada valor de $\alpha, w(\alpha)$ es la solución final en cada iteración sobre las restricciones activas de $w^{t+1} = w^t - A^T(AA^T)^{-1}H$. Donde H es el vector que contiene los valores de las restricciones de igualdad y las restricciones de desigualdad activas al punto w^t
26: Calcular la matriz A en el punto w^t
27: Utilizar una interpolación cuadrática para estimar α^*
28: **end if**
29: $\vec{x}^{t+1} = w(\alpha^{ast})$
30: $t = t + 1$
31: Evaluar las restricciones de desigualdad e identificar las restricciones activas I^t en el punto $\vec{x}^t$. $I^t = \{i : |g_i(\vec{x}^t)| \leq \varepsilon_1, i = 1, 2, \ldots, m\}$
32: $u_m = min\{u_l : l \subset I^t\}$
33: **end while**

de búsqueda utilizados en optimización restringida, los cuales, como se pudo observar, aunque no son nada triviales, permiten obtener resultados alentadores en este tipo de problemas. En el siguiente capítulo se presentarán algoritmos alternativos a los métodos clásicos para dar solución a problemas restringidos. Estos métodos son conocidos con el nombre de Algoritmos Inspirados en la Naturaleza y se mencionarán tanto sus componentes como sus principales paradigmas.

Capítulo 3

Algoritmos Inspirados en la Naturaleza

En este Capítulo se presentarán algunos métodos alternativos a los métodos clásicos utilizados en la solución a problemas de optimización con restricciones. Los métodos aquí presentados, enmarcados dentro de los algoritmos meta-heurísticos, imitan el comportamiento natural de los seres vivos y son identificados como Algoritmos Evolutivos (EA) o Algoritmos Inspirados en la Naturaleza (NIA). Por su simplicidad, en las últimas décadas han sido utilizados en distintas áreas de las ciencias ofreciendo resultados competitivos y atrayendo cada vez más la atención de los investigadores. Los algoritmos Inspirados en la Naturaleza imitan el comportamiento y evolución natural de los seres vivos. Existen dos grandes grupos de estos algoritmos: los Evolutivos y los de Inteligencia Colectiva. Ambos grupos se mencionarán en este trabajo.

3.1. Algoritmos Evolutivos

Los Algoritmos Evolutivos (EAs, por sus siglas en inglés), son métodos inspirados en la naturaleza, los cuales imitan el comportamiento de los seres vivos [60] y basan su función en la evolución de las especies y la supervivencia del más apto. Surgen de la experiencia de observar la naturaleza, de analizar como los organismos que la habitan son capaces de ya sea individualmente o grupalmente realizar ciertas actividades que los llevan a la supervivencia. Dicho análisis es sistematizado y utilizado para dar solución a una gran gama de problemas en en el campo de la optimización.

Los EAs son poblacionales y estocásticos, es decir, puede ser lanzado al espacio de búsqueda más de un punto a la vez y obtener diferentes soluciones al problema. A cada punto en el espacio de búsqueda se le conoce como individuo o solución y dependiendo del algoritmo seleccionado, estos términos

pueden variar de acuerdo al ente natural de inspiración en que se base el algoritmo. La mayoría de los EAs comparten los mismos componentes los cuales se mencionan a continuación.

3.2. Elementos de un Algoritmo Evolutivo

En esta sección se mencionan los elementos básicos de un algoritmo evolutivo. Se describe la población de soluciones, los mecanismos de selección y de reemplazo así como a los operadores de variación.

3.2.1. Población de soluciones

La población es el conjunto de individuos generados en el espacio de búsqueda. Cada individuo perteneciente a la población se genera, en la mayoría de las veces, de forma aleatoria con distribución uniforme dentro del espacio de búsqueda determinado por los límites de las variables de decisión, ver Figura 3.1. Una vez generada la población inicial, cada individuo es evaluado de acuerdo a la función $f(\vec{x})$ y entra en un ciclo conocido como proceso evolutivo, en el cual, mediante la intervención de ciertos operadores de selección y de variación, se obtiene una nueva generación de individuos que reemplazarán a algunos o a todos los elementos de la población inicial. Este proceso continúa hasta que se cumpla una condición de paro, la cual es definida por el usuario y que puede ser alcanzar el valor deseado de la función $f(\vec{x})$ o un cierto número de iteraciones o evaluaciones ejecutadas del algoritmo. Es en cada iteración, también conocida como generación, donde se llevan a cabo todos los procesos del algoritmo

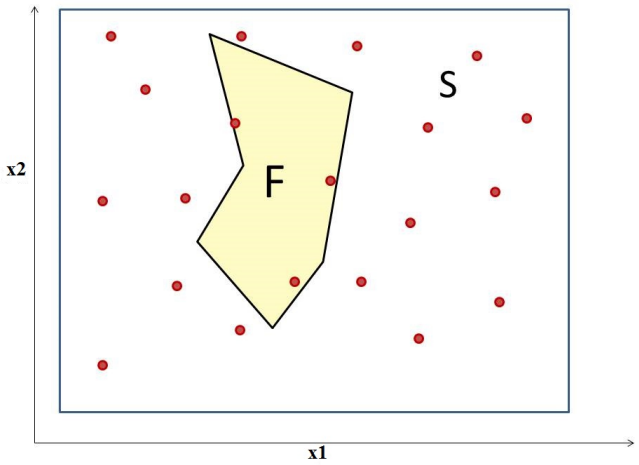

Figura 3.1: Población de soluciones.

3.2.2. Mecanismos de Selección

Un operador de selección separa aquéllos individuos de la población con mejores características, como puede ser un mejor valor en la función $f(\vec{x})$ o un menor valor en la violación de las restricciones. Los individuos seleccionados compartirán sus características a la nueva generación a partir de los operadores de variación. Existen diferentes mecanismos de selección entre ellos se mencionan los basados en selección proporcional como la ruleta, sobrante estocástico y muestreo determinista. Por otro lado, esta la selección mediante torneo y la selección de estado uniforme [12], [27].

3.2.3. Operadores de variación

Los operadores de variación son los encargados de generar nuevos individuos, conocidos como hijos, en la población. Las nuevas soluciones son creadas a partir de los individuos seleccionados por los mecanismos de selección, que en este paso toman el rol de padres. Aunque la mayoría de los operadores de variación utiliza operaciones matemáticas básicas sobre los padres para generar los nuevos descendientes, es necesario aclarar que también existen operadores en los cuales se requiere un grado matemático más avanzado para su implementación [21], [25]. Dentro de los operadores de variación se encuentra a los operadores de cruza y los operadores de mutación. La idea general es que los operadores de cruza transmitan a los nuevos individuos aquellas características que permitan explotar el espacio de búsqueda, mientras que los operadores de mutación son aplicados para aumentar la capacidad de exploración del espacio de búsqueda. Existe una gran variedad de operadores de cruza y mutación dependiendo del tipo de representación utilizada, binaria, entera, real o permutaciones, entre otras [12], [36]. La Figura 3.2 muestra un ejemplo de la cruza de un punto en representación binaria donde a partir de dos padres seleccionados se crean dos descendiente después de aplicar el operador de cruza sobre los padres. En la cruza de un punto, se selecciona un punto aleatorio de corte de acuerdo al tamaño de las soluciones, la parte izquierda del punto de corte pasa directamente a cada uno de los hijos, la parte derecha del punto de corte se intercambia en los hijos, es decir, el padre 1 le aporta esa parte al hijo 2 y el padre 2 al hijo 1.

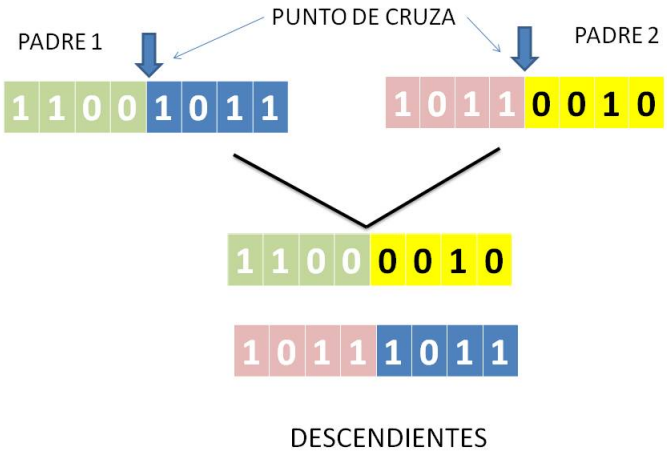

Figura 3.2: Cruza de un punto.

3.2.4. Mecanismos de reemplazo

El mecanismo de reemplazo es el utilizado para decidir qué individuos ya sea de la población actual, de la nueva población o de la combinación de éstas formarán la población que continuará para la siguiente generación. El reemplazo puede ser poblacional, en el que la población nueva sustituye a la actual. Puede ser elitista, en la que sólo los mejores individuos de la combinación entre la población nueva y la actual serán los que permanecerán para la siguiente generación. El concepto elitista alude al sentido de que la mejor solución de la población actual pasará intacta a la siguiente generación.

Los elementos presentados constituyen los componentes generales de un Algoritmo Evolutivo (ver Figura 3.3). Gracias al éxito reportado por el uso de los mismos la literatura presenta una gran variedad de estos algoritmos donde éstos componentes pueden observarse. En este capítulo sólo se presentarán los paradigmas más utilizados en el área.

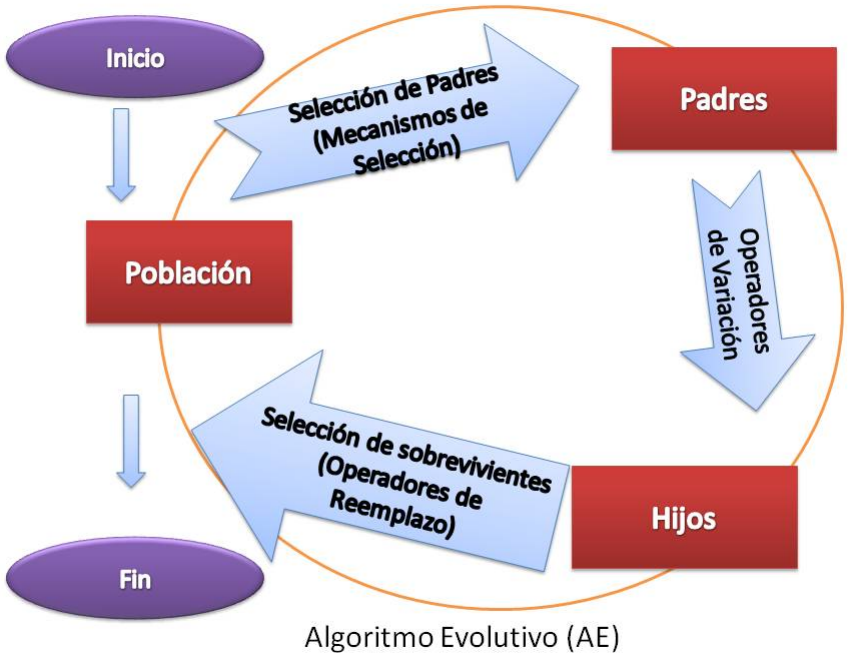

Figura 3.3: Esquema general de un Algoritmo Evolutivo.

3.3. Paradigmas de Algoritmos Evolutivos

Los primeros paradigmas de algoritmos evolutivos surgidos en la década de 1960 son mencionados a continuación.

3.3.1. Programación Evolutiva

En la década de l960, Fogel et al., [23] propuso la técnica de Programación Evolutiva, la cual contempla sólo al operador de mutación para generar los nuevos individuos, esto con el fin de simular el proceso evolutivo a nivel de las especies donde especies diferentes no se cruzan. El Algoritmo 5 presenta los pasos generales de la programación evolutiva.

Algoritmo 5 Programación Evolutiva

1: Generar aleatoriamente una población inicial;
2: Evaluar la población inicial;
3: Aplicar mutación a la población;
4: Evaluar la nueva población;
5: Seleccionar a los individuos que pasará a la siguiente generación;
6: Regresar al paso 3 hasta que se cumpla una condición de paro;

3.3.2. Estrategias Evolutivas

Igno Rechenberg por el año de 1964 [2] introduce las Estrategias Evolutivas (EE), inspirado en el mecanismo de mutación observado en la naturaleza donde las mutaciones pequeñas son más frecuentes a diferencia de mutaciones grandes. El primer algoritmo de este tipo no fue poblacional, la evolución se generaba a partir de un individuo que figura como padre al cual se le aplica una mutación con ruido Gaussiano y se obtenía un nuevo descendiente (1+1)-EE. Al final, se aplica un proceso de selección extintiva ya que el mejor entre los dos individuos (padre e hijo) será el que pase a la siguiente generación. El nuevo individuo es generado a partir de la Ecuación (3.1).

$$\vec{x}_{t+1} = \vec{x}_t + N(0, \sigma) \tag{3.1}$$

donde t indica la generación actual y $N(0, \sigma)$ es un vector de números Gaussianos con media cero y desviación estándar σ.

En años posteriores el algoritmo se vuelve poblacional y se agregan las siguientes versiones:

- $(\mu + 1)$-EE, en la cual se presentan μ padres, los cuales generan un solo descendiente. El proceso de selección continúa siendo extintiva, ya que el peor de los padres se elimina.

- $(\mu + \lambda)$-EE, en la que μ padres generan λ hijos. Los μ mejores entre la combinación de padres e hijos pasan a la siguiente generación.

- (μ, λ)-EE, en la que μ padres generan λ hijos. Los μ mejores hijos pasan a la siguiente generación.

Por último, es importante destacar que las estrategias evolutivas introducen el término de auto-adaptación de parámetros.

3.3.3. Algoritmos Genéticos

Los Algoritmos Genéticos (GAs) surgen del interés de John H. Holland en la década de los 1960 por estudiar procesos de adaptación [33]. Los GAs hacen énfasis en los operadores de cruza sexual, ya que a partir de dos individuos que participan como padres se obtiene un nuevo descendiente. El Algoritmo 6 presenta de manera general los pasos necesarios para implementar un Algoritmo Genético. El elitismo es muy utilizado por estos algoritmos al momento del reemplazo ya que se recomienda conservar siempre al mejor individuo de cada generación para acelerar el proceso de convergencia, la cual puede ser prematura o exitosa.

Algoritmo 6 Algoritmo Genético

1: Generar aleatoriamente una población inicial;
2: Evaluar la población inicial;
3: Seleccionar de forma probabilista con base en su aptitud los individuos que tomarán el rol de padres;
4: Aplicar operadores de variación a los padres, para generar la nueva población;
5: Evaluar la nueva población;
6: La nueva población sustituye a la anterior y se aplica elitismo;
7: Regresar al paso 3 hasta que se cumpla una condición de paro;

3.4. Inteligencia Colectiva

Al conjunto de algoritmos que emulan el comportamiento social de organismos y resolver problemas complejos, se le conoce con el nombre de Inteligencia Colectiva IC [22].

3.4.1. Fundamento Biológico

La IC surge del análisis del comportamiento social en los organismos vivos. Parte del análisis de pequeños insectos como hormigas y abejas que trabajan de forma colectiva para obtener un bien común, como por ejemplo, la obtención de alimento y/o la construcción de sus nidos. Cuatro reglas pueden observarse en este comportamiento [61]:

1. Cada individuo se rige mediante reglas simples.

2. No existe un control o controlador central que mueva o rija a cada individuo.

3. Estos individuos pueden auto-organizarse y colectivamente mostrar habilidades que les permiten resolver problemas complejos. A este comportamiento colectivo, se le conoce como inteligencia.

4. El comportamiento colectivo surge a través de la interacción entre los organismos, el cual puede ser por comunicación local o indirecta mediante cambios en el ambiente.

A partir de estas reglas se define que la inteligencia se obtiene en el proceso de interacción o cooperación entre individuos. Por lo tanto, el establecer un modelo que imite estos comportamientos de la naturaleza, establece las pautas necesarias para aplicarlo en la resolución de problemas de optimización complejos. A estos modelos sistematizados se les conoce con el nombre de IC. Algunos algoritmos representativos de IC son presentados a continuación.

3.4.2. Algoritmo ACO

Ant Colony Optimization (ACO), un algoritmo de optimización basado en colonia de hormigas, fue propuesto por Marco Dorigo en [18], e imita el comportamiento de las hormigas al llevar a cabo la tarea de forrajeo. Cuando una hormiga encuentra una fuente de alimento, lleva parte del alimento a la colonia, dejando en su trayecto el rastreo de una sustancia conocida como feromona. La feromona se evapora con el tiempo. De acuerdo a la concentración de esta sustancia en el camino, el resto de las hormigas decidirá si tomar éste o algún otro camino. El camino con mayor concentración de feromona tiene mayor posibilidad de ser transitado y por lo tanto, el camino más transitado suele ser el camino más corto entre la fuente de alimento y el nido. Siguiendo esta metáfora es como surge ACO, donde el problema es representado mediante un grafo con aristas que representan los posibles caminos a seguir. Las hormigas serán generadas artificialmente y se encargarán de recorrer dichos caminos con el objetivo de encontrar una solución competitiva al problema de optimización. Los pasos necesarios para implementar ACO se encuentran en el Algoritmo 7.

Algoritmo 7 Algoritmo ACO

1: Inicializar de forma aleatoria el depósito de feromonas en cada nodo;
2: Colocar a las hormigas en un nodo inicio ;
3: **while** no se cumpla la condición de paro **do**
4: Basado en la concentración de feromona en cada nodo, decidir el nodo a visitar, marcando los caminos recorridos con mayor cantidad de feromona;
5: Actualizar la cantidad de feromona de cada nodo;
6: Computar la longitud de la ruta de cada hormiga;
7: **end while**
8: Al final quedarse con la ruta más corta;

Aunque ACO es un algoritmo muy diferente a otros algoritmos de IC, suele ser muy competitivo particularmente en problemas de optimización combinatoria.

3.4.3. Algoritmo PSO

Particle Swarm Optimization (PSO) propuesto por Kennedy and Eberhart in 1995 [35], es un algoritmo que surge a través de la simulación del vuelo de las aves partiendo de la sincronía entre los pájaros, el cambio de dirección repentino, la dispersión y su reagrupamiento. En este algoritmo particular, a la población se le conoce como cúmulo y a los individuos como partículas. El cúmulo puede dividirse en vecindarios lo cual favorece la exploración del espacio de búsqueda. Cada partícula está formada por una

posición actual, una memoria que guarda la mejor posición por donde ha pasado (*Pbest*) y una velocidad para indicar la dirección de búsqueda. Así, las partículas vuelan por el espacio de soluciones tratando de encontrar la mejor de ellas para un determinado problema, actualizando su posición a partir de su experiencia *Pbest* y de la posición del líder en el cúmulo. El Algoritmo 8 muestra los pasos necesarios para implementar PSO.

PSO es un algoritmo de convergencia muy rápida por lo tanto, es posible

Algoritmo 8 Algoritmo PSO

1: Generar de forma aleatoria un cúmulo inicial de partículas;
2: Calcular la aptitud de cada partícula ;
3: **while** no se cumpla la condición de paro **do**
4: Seleccionar al líder del cúmulo;
5: Aplicar el operador de vuelo a cada partícula;
6: Evaluar cada partícula;
7: Actualizar el valor de *Pbest* de cada partícula;
8: **end while**
9: Del cúmulo final, reportar la mejor solución obtenida;

quedar atrapado en óptimos locales debido a esta característica.

3.4.4. Algoritmo ABC

Artificial Bee Colony (ABC), es un algoritmo propuesto por Karaboga en el 2005 [37] que emula el comportamiento de forrajeo de las abejas meliferas. Tres tipos de abejas son consideradas en este algoritmo: abejas empleadas, abejas observadoras y abejas exploradoras. Las abejas empleadas son asignadas a una fuente de alimento y comparten su información mediante una danza a las abejas observadoras, las cuales reciben la información y eligen la fuente con mayor cantidad de néctar. Por último, las abejas exploradoras se encargan de buscar nuevas fuentes de alimento. En este algoritmo cada fuente de alimento, representa una solución potencial al problema y la cantidad de néctar representa su aptitud. El algoritmo general de ABC se muestra en el Algoritmo 9.

Algoritmo 9 Algoritmo ABC

1: Generar de forma aleatoria un conjunto inicial de fuentes de alimento;
2: Evaluar cada fuente de alimento;
3: **while** no se cumpla la condición de paro **do**
4: Las abejas empleadas visitan su fuente de alimento y generan una nueva, la cual si es mejor sustituirá a la actual;
5: Las abejas observadoras seleccionarán la fuente de alimento que visitarán con base en su aptitud;
6: Las abejas observadoras visitan la fuente elegida y crearán una nueva fuente, que de ser mejor sustituirá a la actual;
7: Las abejas exploradoras crearán fuentes de alimento que sustituirán aquéllas que no hayan sido mejoradas después de un límite de ciclos;
8: **end while**
9: Obtener la mejor fuente de alimento del conjunto final;

3.5. Técnicas para el manejo de restricciones

Originalmente los algoritmos inspirados en la naturaleza no fueron pensados para trabajar en espacios restringidos, es por ello que al enfrentarse a problemas con restricciones se hizo necesaria la adición de métodos que permitieran manejar las restricciones asociadas al problema. Algunos de estos métodos son mencionados a continuación.

3.5.1. Penalización de funciones

La penalización de funciones consiste en transformar la función original con restricciones a una función sin restricciones agregando a la función objetivo factores de penalización que consideran el valor de las restricciones violadas.

Una función de penalización puede representarse de manera general como sigue:

$$f_p(\vec{x}) = f(\vec{x}[\sum_{i=1}^{n} r_i \times g_i + \sum_{j=1}^{p} c_j \times h_j]) \tag{3.2}$$

donde, f_p es la nueva función sin restricciones a optimizar, g, h son las restricciones de desigualdad e igualdad respectivamente, y r_i, c_j son los factores de penalización.

El mayor inconveniente en este método es la calibración de los factores de penalización, ya que valores grandes favorecen una rápida aproximación a la zona factible, pudiendo quedar atrapado en óptimos locales. Por otra parte, valores pequeños favorecen una aproximación más lenta a la zona factible necesitando un número mayor de evaluaciones para encontrar un buen valor factible de la función objetivo.

3.5.2. Reglas de Factibilidad

Las Reglas de Factibilidad (FR, por sus siglas en inglés Feasible Rules), fueron propuestas por Deb [16] en el 2000. Ejercen presión al momento de la selección, ya que soluciones factibles son preferibles a soluciones no factibles, permitiendo una rápida convergencia hacia la región factible. El método consiste en comparar dos soluciones tomando en cuenta tanto el valor de violación de las restricciones como el valor de la función objetivo. Las reglas que se aplican en el proceso de comparación se mencionan a continuación:

1. Entre dos soluciones factibles, aquélla con mejor valor en la función objetivo es seleccionada.
2. Entre dos soluciones no factibles, aquélla con menor grado de violación de restricciones es seleccionada.
3. Entre dos soluciones, una factible y otra no factible, la solución factible es preferible.

3.5.3. Jerarquización estocástica

El método fue propuesto por Runarsson y Yao en [49], (SR, por sus siglas en inglés Stochastic Ranking). El objetivo fue mitigar los inconvenientes de la calibración de los factores de penalización, en la penalización de funciones. SR sólo utiliza un parámetro Pf definido por el usuario para controlar el criterio utilizado al comparar soluciones no factibles. Estos criterios son: 1) basarse en la suma de la violación de las restricciones, $\phi(\vec{x})$ o 2) basarse sólo en el valor de la función objetivo $f(\vec{x})$. Los pasos para implementar SR son descritos en el Algoritmo 10.

La presión de selección ejercida por SR se puede calibrar, ya que de acuerdo al valor del parámetro Pf es posible incorporar a la población soluciones no factibles con un buen valor en la función objetivo, favoreciendo así la diversidad de la misma. Un proceso adaptativo puede incorporarse a SR, de tal manera que al inicio del ciclo evolutivo el parámetro Pf obtenga valores altos que favorezcan la diversidad en la población y al mismo tiempo la exploración del espacio de búsqueda. Por otro lado, al final del ciclo evolutivo usar valores pequeños de Pf permitirá favorecer a la convergencia de la población y a su vez a la explotación del espacio de búsqueda.

Algoritmo 10 Algoritmo SR

```
1: I_j = j ∀j ∈ {1,...,λ}
2: for i=1 to NP do
3:   for j=1 to λ − 1 do
4:     u = rand (0,1)
5:     if (φ(I_j) = φ(I_{j+1})) or (u < Pf) then
6:       if (f(I_j) > f(I_{j+1})) then
7:         intercambiar(I_j, I_{j+1}))
8:       end if
9:     else
10:      if (φ(I_j) > φ(I_{j+1})) then
11:        intercambiar(I_j, I_{j+1}))
12:      end if
13:    end if
14:  end for
15: end for
```

3.5.4. Método $\varepsilon-$Constrained

El método fue propuesto por Takahama y Sakai [57]. Al igual que la penalización de funciones forma parte de los métodos de transformación, en el cual un problema restringido es transformado en un problema sin restricciones. Consiste principalmente de dos partes: 1) un método de relajación, en el que se considera a una solución como factible con base en la suma de violación de sus restricciones y el valor de una tolerancia ε, y, 2) un mecanismo en el cual la minimización de la suma de violación de restricciones precede a la minimización de la función objetivo. El parámetro ε determina el nivel de comparación entre dos soluciones $\vec{x}_1$ y $\vec{x}_2$ como se muestra a continuación:

$$f(\vec{x_1}), \phi(\vec{x_1}) <_\varepsilon f(\vec{x_2}), \phi(\vec{x_2}) \iff \begin{cases} f(\vec{x_1}) < f(\vec{x_2}), \text{if } \phi(\vec{x_1}), \phi(\vec{x_2}) \leq \varepsilon; \\ f(\vec{x_1}) < f(\vec{x_2}), \text{if } \phi(\vec{x_1}) = \phi(\vec{x_2}); \\ \phi(\vec{x_1}) < \phi(\vec{x_2}), \text{otherwise} \end{cases}$$

donde, $\phi(\vec{x})$ se refiere a la suma de violación de restricciones y $f(\vec{x})$ es el valor de la función objetivo.

Los valores de ϵ son controlados de acuerdo a la Ecuación (3.3).

$$\epsilon(0) = \phi(\vec{x_\theta})$$
$$\epsilon(t) = \begin{cases} \epsilon(0)(1 - \frac{t}{Tc})^{cp} & 0 < t < Tc; \\ 0 & t \leq Tc. \end{cases} \tag{3.3}$$

donde t representa la iteración actual; Tc es el máximo de iteraciones permitidas al algoritmo and $\phi(\vec{x_\theta})$ es la solución con mayor violación de restric-

ciones de la θ-th solución, $\theta = 0.2\text{NP}$ y cp es el parámetro que controla el decremento de la tolerancia de la violación.

Al principio del ciclo evolutivo, el algoritmo permite mayor diversidad y mayor relajación en el proceso de selección ya que soluciones no factibles son permitidas en la nueva población de acuerdo al valor de ε. La presión de selección aumenta al final de proceso evolutivo, ya que cuando ε es cero, el método se convierte automáticamente en las reglas de factibilidad propuestas por Deb.

3.5.5. Operadores especiales

Los operadores especiales son métodos que actúan en un área específica de interés en el espacio de búsqueda en combinación con el algoritmo principal. Debido a que en problemas de optimización restringida se requiere concentrar la atención en la región factible y en la frontera de ésta, los operadores especiales pueden clasificarse como sigue.

- **Operadores de Frontera**. La existencia de restricciones activas hace probable que el óptimo de una función se encuentre en el área comprendida entre la región factible y la región no factible, por lo tanto, los operadores de frontera intentan explorar y mantener soluciones en esta área de interés (ver Figura 3.4). A continuación se mencionan algunos de los operadores de frontera encontrados en la literatura.

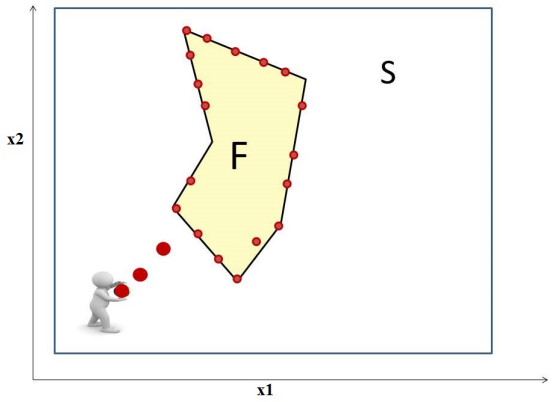

Figura 3.4: Operadores de Frontera.

  - Cruza Geométrica y Esférica. Estos dos operadores especiales de frontera fueron propuestos por Schoenauer y Michalewicz en [51]. A partir de un proceso de inicialización de la población ad hoc para dos problemas específicos donde se generan soluciones que de inicio se posicionan justo en la frontera de la región factible, los

operadores de cruza especiales son utilizados para generar nuevas soluciones que permanezcan en esta región de interés.

- En 2002 Wu, Z.Y. y Simpson, A.R. [59] utilizaron la penalización de funciones para explorar la frontera de la región factible con ayuda de un Algoritmo Genético. Mediante la co-evolución y la auto-adaptación de los factores de penalización intentan guiar la búsqueda hacia la frontera factible.

- Boundary Search Operator. En 2006 [38] Guillermo Leguizamón presentó un operador especial capaz de explorar la frontera factible de las restricciones aplicando una búsqueda binaria. Partiendo de dos puntos, uno factible y uno no factible, se adentra en un ciclo en el cual se genera el punto medio entre los dos puntos, cuando el punto medio se encuentra sobre o muy cerca de la frontera factible el proceso termina.

- En 2014 Bonyadi y Michalewicz [3] presentaron una técnica llamada Subset Constraints Boundary Narrower (SCBN) que permite ajustar el área factible de cualquier problema restringido a su frontera factible definida por el conjunto de restricciones contempladas en SCBN. La técnica propuesta utiliza un parámetro que permite definir el tamaño de la frontera a explorar. Dado un punto x, el parámetro tomará un valor menor a 0 si x es factible y alguna de las restricciones contenidas en SCBN se encuentra dentro de un rango establecido.

De estos operadores el propuesto en [38] se mencionará en capítulos posteriores.

- **Operadores de Factibilidad**. En este tipo de operadores se intenta mantener o buscar la factibilidad de las soluciones en todo momento ya sea a partir de puntos iniciales no factibles o en su defecto, mantener la factibilidad de las soluciones (ver Figura 3.5).

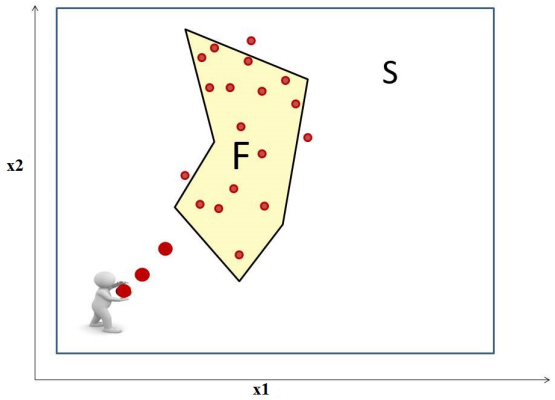

Figura 3.5: Operadores de Factibilidad.

Los operadores especiales que inician el proceso de búsqueda a partir de soluciones no factibles tratan de alcanzar el punto más cercano a la zona factible con la mayor rapidez posible, para después concentrar la búsqueda en optimizar la función objetivo y no en satisfacer las restricciones asociadas a dicha función. En esta categoría se encuentran los operadores basados en vectores de proyección de factibilidad que utilizan el gradiente de cada restricción como la dirección de búsqueda, así podemos mencionar el método de consenso de restricciones propuestos por John W. Chineck, [10], [11]. De los operadores que intentan mantener la factibilidad de las soluciones podemos mencionar a GENOCOP III [44], el cual parte de dos poblaciones donde el desenvolvimiento de una afecta el desarrollo de la otra. La primera población contiene sólo puntos que satisfacen las restricciones lineales cuya factibilidad es mantenida por operadores especiales. La segunda población contiene puntos de referencia completamente factibles, es decir, que satisfacen todas las restricciones. Estos puntos son evaluados directamente en la función objetivo. Los puntos no factibles son reparados en cada generación haciendo uso de los puntos de referencia. Cuando alguno de los puntos reparados es mejor que alguno de los puntos de referencia, el punto reparado sustituye al punto de referencia con el objetivo de generar convergencia hacia mejores zonas del espacio de búsqueda. Como salida el método ofrece sólo puntos factibles.

3.6. Conclusiones

Algunos de los métodos alternativos a los métodos clásicos utilizados en optimización restringida fueron expuestos en este capítulo. Se describieron los elementos básicos que los caracterizan, así como los principales paradigmas

surgidos en la década de 1960. Entre los principales paradigmas, el más utilizado en la literatura es sin duda el Algoritmo Genético propuesto por John H. Holland, este auge se debe principalmente a su capacidad de admitir varias representaciones ya sea binaria, real, entera, entre otras, además de poder ser utilizado tanto en optimización numérica como en optimización combinatoria. En el capítulo se hizo referencia a las técnicas para el manejo de restricciones ya que en espacios restringidos forman la dupla perfecta con los NIAs para tratar las restricciones asociadas al problema.

Una área de interés dentro de los NIAs es la IC, la cual surge de analizar el comportamiento e interacción de organismos sociales dentro de la naturaleza. El siguiente capítulo, se centra en un nuevo algoritmo de IC conocido como algoritmo BSO, el cual está inspirado en el proceso de tormenta de ideas utilizado por el ser humano para dar solución a problemáticas difíciles de resolver de forma individual. Se hace un comparativo de tres versiones del algoritmo con tres técnicas para el manejo de restricciones con el objetivo de determinar si alguna versión presenta ventajas competitivas sobre otra y utilizarla como algoritmo base en capítulos posteriores.

Capítulo 4

Algoritmo BSO en espacios restringidos

En el esfuerzo por dar solución a problemas restringidos, los NIAs han sido exitosamente utilizados y el área de IC ha sido particularmente activa en este sentido [15], [22], [36]. Recientemente el algoritmo BSO, ha mostrado resultados competitivos al ser utilizado en problemas de optimización sin restricciones. El algoritmo fue propuesto por Yuhui Shi en 2011 [62], analizando su comportamiento en dos funciones sin restricciones obteniendo resultados finales competitivos con algoritmos del estado del arte. En 2012 Zhan et al. [63] propuso algunas modificaciones al algoritmo BSO y presentó una nueva versión conocida como: Modified Brain Storm Optimization (MBSO). En esta versión, dos modificaciones fueron hechas al algoritmo BSO, la primera de ellas fue utilizar el método conocido como: Simple Grouping Method, en lugar del algoritmo k-means en el operador de agrupamiento. La segunda modificación fue descartar la estrategia de ruido Gaussiano aleatorio para generar nuevas ideas proponiendo una nueva estrategia en su lugar. En 2013, un estudio de parámetros fue presentado en BSO [29], en este trabajo, las dos versiones anteriores, BSO y MBSO, fueron tomadas en cuenta para analizar los tres parámetros utilizados en los principales operadores del algoritmo, estos parámetros son: $p_{replace}$, p_{one}, y p_{center}. El primer parámetro es una probabilidad para dar mayor capacidad de exploración al algoritmo y así reducir la posibilidad de quedar atrapado en óptimos locales. El segundo parámetro se relaciona a la probabilidad de usar uno o dos clusters para generar las nuevas ideas, por último, el tercer parámetro es una probabilidad para generar las nuevas ideas basados ya sea en la mejor idea o en una idea aleatoria del cluster seleccionado. Los mejores resultados se obtuvieron al omitir el parámetro $p_{replace}$ y al utilizar sólo un cluster para generar las nuevas ideas con un valor obtenido de una distribución Gaussiana con media de 0.4 y desviación estándar de 0.1 para el parámetro p_{center}, de acuerdo a los autores, estos parámetros les permitieron obtener mejores resultados en

el conjunto de prueba utilizado, además, permiten un balance entre la exploración y explotación en el espacio de búsqueda. Desde entonces, BSO ha atraído la atención de diferentes investigadores en el área [9]. Sin embargo, la literatura muestra que su desempeño en problemas de optimización numérica con restricciones ha sido poco explorado. Por lo tanto, siendo BSO un algoritmo de reciente aparición y motivados por el éxito mostrado en espacios sin restricciones, es este capítulo se presenta un estudio de tres versiones del algoritmo BSO: Brain Storm Optimization Algorithm (BSO) [62], Modified Brain Storm Optimization Algorithm (MBSO) [63] y Simple Modified Brain Storm Optimization Algorithm (SMBSO) [29] en problemas de optimización numérica con restricciones utilizando tres diferentes manejadores de restricciones que son las reglas de factibilidad, el método de $\varepsilon - constrained$ y jerarquización estocástica. El desempeño de cada versión se analiza al resolver los 24 problemas de optimización propuestos en el Congreso de Cómputo Evolutivo celebrado en el 2006 (CEC2006) [39].

4.1. Algoritmo de Optimización Basado en el proceso de Tormenta de ideas, BSO

Brain Storm Optimization Algorithm (BSO) propuesto por Yuhui Shi en el 2011 [62] es un algoritmo de inteligencia colectiva inspirado en el método de tormenta de ideas, utilizado por los seres humanos para generar soluciones a problemas que difícilmente podrían resolverse individualmente. La motivación del algoritmo parte de que al ser inspirado indirectamente en el ser humano y siendo éste el animal más inteligente en la naturaleza, BSO debería ser, sino el mejor, uno de los mejores algoritmos del estado del arte en resolver problemas de optimización. A continuación se muestra el proceso de tormenta de ideas el cual sirve de base para el algoritmo BSO.

4.1.1. Tormenta de ideas (Brainstorming Process)

El proceso de tormenta de ideas consiste en reunir a un grupo de personas con diferentes especialidades con la finalidad de generar el mayor número posible de ideas para tratar de resolver una problemática.

Cuatro reglas propuestas en [47], son tomadas en cuenta para facilitar el proceso de generación de ideas, las cuales se definen a continuación.

1. No criticar, no enjuiciar las ideas generadas.

2. Toda idea es bienvenida.

3. Generar el mayor número de ideas como sea posible.

4. Combinar las ideas para generar nuevas ideas.

El seguimiento de las cuatro reglas anteriores permite a cada integrante pensar libremente y genera el ambiente propicio para reunir el mayor número posible de ideas.

A continuación se mencionan los pasos necesarios para generar una tormenta de ideas:

1. Reunir un grupo de personas con diferentes especialidades.
2. Generar el mayor número posible de ideas, tomando en cuenta las cuatro reglas mencionadas anteriormente.
3. Seleccionar mínimo dos personas que tomarán el rol de propietarios del problema, cuya función principal es seleccionar las mejores ideas que solucionen dicho problema.
4. Utilizar las ideas seleccionadas como pistas para generar nuevas ideas.
5. Una vez generadas las nuevas ideas, los propietarios del problema seleccionan las mejores ideas que resuelvan su problemática.
6. Para evitar quedarse atrapados en las mismas ideas, puede seleccionarse aleatoriamente un grupo de ideas y tomarse como base para generar las nuevas ideas.
7. Nuevamente los propietarios del problema seleccionan las mejores ideas.
8. El proceso continua hasta cumplirse un limite de tiempo proporcionado.

Imitando los pasos de la tormenta de ideas, es como surge el algoritmo de optimización BSO, el cual se describe en el Algoritmo 11. Dentro del algoritmo cada idea se corresponde con una posible solución al problema donde al evaluarse en la función objetivo se determina con base en su aptitud cual de esas soluciones es la que mejor satisface al problema actual. En BSO existen tres operadores principales: operador de agrupamiento, operador de reemplazo y operador de cruza, los cuales se describen a continuación.

4.1.2. Operador de Agrupamiento

El proceso de agrupar las ideas generadas en el algoritmo BSO simula el rol de los propietarios del problema de seleccionar las mejores ideas en el proceso de tormenta de ideas. El objetivo de este operador es guiar la búsqueda del algoritmo hacia zonas promisorias dentro del espacio de búsqueda. Aunque distintas técnicas de agrupamiento pueden ser utilizadas en este operador, el algoritmo de k-means es utilizado en el algoritmo original de BSO [62]. En la actualidad hay diferentes trabajos en este sentido [8], [63], [64].

Algoritmo 11 BSO algoritmo

1: De forma aleatoria generar NP ideas y evaluarlas;
2: **while** $iter < Max_{iter}$ **do**
3: *// operador de agrupación*
4: Agrupar las NP ideas en M clusters;
5: Identificar la mejor idea en cada cluster, con base en su aptitud, como el centroide;
6: *// Operador de reemplazo*
7: **if** $random(0,1) < p_{replace}$ **then**
8: Aleatoriamente seleccionar un cluster y reemplazar el centroide por una idea generada aleatoriamente;
9: **end if**
10: *// Operador de variación*
11: **for** i=1 to NP **do**
12: **if** $random(0,1) < p_{one}$ **then**
13: Aleatoriamente seleccionar un cluster
14: **if** $random(0,1) < p_{oneCenter}$ **then**
15: Utilizar el centroide del cluster seleccionado y generar la nueva idea Y_i;
16: **else**
17: Utilizar una idea aleatoria del cluster seleccionado y generar la nueva idea Y_i;
18: **end if**
19: **else**
20: Aleatoriamente seleccionar dos clusters;
21: **if** $random(0,1) < p_{twoCenter}$ **then**
22: Generar la nueva idea Y_i, tomando como base la idea generada después de combinar los dos centroides de los clusters seleccionados;
23: **else**
24: Generar la nueva idea Y_i, tomando como base la idea generada después de combinar dos ideas aleatorias de los clusters seleccionados;
25: **end if**
26: **end if**
27: Evaluar la nueva idea Y_i. Si Y_i es mejor que X_i, $X_i = Y_i$;
28: **end for**
29: **end while**

4.1.3. Operador de Reemplazo

El objetivo general del operador de reemplazo es evitar que el algoritmo quede atrapado en óptimos locales y a su vez generar mayor diversidad en la población actual y poder cubrir la mayor parte del espacio de búsqueda [29]. Lo anterior se emula reemplazando la mejor idea de un cluster seleccionado aleatoriamente por una nueva idea generada de forma aleatoria en el rango de las variables de decisión.

Existen algunos trabajos que revelan la poca utilidad de este operador e incluso deciden no utilizarlo o proponen rediseñar la forma en que se aplica el operador dentro del algoritmo, por ejemplo reemplazar la peor idea y no necesariamente la mejor [29].

4.1.4. Operador de cruza

La finalidad del operador de cruza es generar las nuevas ideas tomando como base las ideas seleccionadas ya sea a partir de uno o más clusters. La generación de ideas a partir de un cluster favorece la habilidad de explotación del algoritmo refinando las áreas de interés en el espacio de búsqueda. Al contrario, el generar ideas a partir de más de un cluster favorece la diversidad de la población y por lo tanto, la capacidad de exploración del algoritmo [7]. Tomando en consideración este análisis, se podría diseñar una nueva version del algoritmo BSO aumentando o disminuyendo la probabilidad para generar las nuevas ideas a partir de uno o más clusters de acuerdo al estatus de la población actual en cada generación.

En el algoritmo BSO [62] se generan las nuevas ideas agregando un ruido Gaussiano a la idea seleccionada como se muestra en la Ecuación (4.1) y (4.2).

$$Y_i = X_s + \xi * n(\mu, \sigma) \tag{4.1}$$

$$\xi = logsig(\frac{(0{,}5 * T - t)}{k}) * rand() \tag{4.2}$$

donde Y_i es la nueva idea, X_s es la idea seleccionada; $n(\mu, \sigma)$ es una función Gaussiana con media μ y varianza σ; T es el máximo número de iteraciones, t es la iteración actual y k es usado para cambiar la pendiente de la función logsig () y establece el tamaño de paso en el operador de cruza.

4.1.5. Parámetros de control en el algoritmo BSO

- $p_{replace}$: Es un parámetro que favorece la diversidad del algoritmo y evitar quedar atrapado en óptimos locales al cambiar el centroide por una solución generada aleatoriamente. El autor propone un valor inicial de 0.2 para este parámetro.

- p_{one}: Parámetro que determina si las nuevas ideas serán creadas a partir de uno o más clusters. El autor propone un valor inicial de 0.8 para este parámetro.

- $p_{oneCenter}$, $p_{twoCenter}$: Parámetros que determinan si las nuevas ideas se crearán a partir de la mejor idea (centroide) o a partir de ideas aleatorias del cluster seleccionado. El autor propone un valor inicial de 0.4 para $p_{oneCenter}$ y un valor inicial de 0.5 para $p_{twoCenter}$.

- **M**: Parámetro numérico positivo que indica el número de clusters que se formarán en el algoritmo. Su valor inicial propuesto por el autor es de M = 5.

- **k**: Parámetro numérico que permite definir el tamaño de paso en la función logsig () en el operador de cruza del algoritmo BSO. El valor propuesto por el autor para este parámetro es k = 20.

A pesar de ser un algoritmo relativamente joven, el algoritmo BSO ha llamada la atención de los investigadores en los recientes años, muestra de esto se encuentra en el survey publicado por Shi Cheng en 2017 [9]. Distintas variantes al algoritmo BSO han sido presentadas en la literatura especializada, en este apartado sólo se presentan dos de las primeras modificaciones al algoritmo por ser sujetas de estudio en este capítulo.

4.2. Algoritmo MBSO

Modified Brain Storm Optimization algorithm (MBSO) es una versión del algoritmo BSO que surge en 2012 [63], en la cual se sustituye el algoritmo k-means utilizado en el operador de clustering por un método simple de agrupación (Simple Grouping Method, SGM). Además de lo anterior, sustituye el ruido Gaussiano utilizado en BSO para generar las nuevas ideas por una estrategia de diferencia de ideas (Idea Difference Strategy, IDS).

El proceso de agrupamiento utilizado en MBSO se describe a continuación.

1. Seleccionar M ideas aleatoriamente (diferentes entre sí) de la población actual, las cuales se convertirán en la semilla de cada cluster.

2. Por cada idea en la población calcular la distancia Euclidiana a cada cluster.

3. Comparar las distancias y asignar la idea al cluster más cercano.

4. Regresar al paso 2 hasta que todas las ideas hayan sido agrupadas.

Introduce una nueva estrategia para generar las ideas tomando en cuenta la diferencia generada por dos ideas seleccionadas de forma aleatoria de la población actual. De esta manera la nueva idea es generada de acuerdo a la Ecuación (4.3).

$$Y_i = \begin{cases} rand(L,H) & \text{if } rand(0,1) < pr; \\ X_i + rand(0,1) \times (X_a - X_b) & \text{otherwise.} \end{cases} \tag{4.3}$$

donde Y_i es la nueva idea, L, H representan el limite inferior y superior de las variables de decisión respectivamente. X_i es la idea seleccionada a partir de uno o más clusters, X_a, X_b son dos ideas diferentes entre si seleccionadas de forma aleatoria de la población actual; pr es un parámetro que simula la libertad de la generación de ideas simulando la regla 2 del proceso de Osborn [47].

4.3. Algoritmo SMBSO

El algoritmo SMBSO (Simple Modified Brain Storm Optimization) surge de un estudio sobre los parámetros del algoritmo BSO, en el cual determinan la afectación de los parámetros en el rendimiento del algoritmo. Tres son los parámetros en cuestión: $p_{replace}$, p_{one} and p_{center}. De este estudio se concluye que la ausencia o el menor uso del operador de reemplazo $p_{replace}$ favorece el rendimiento del algoritmo, por lo que el no utilizarlo hace al algoritmo más simple, y de ser utilizado debería rediseñarse para darle un mejor uso. Respecto del parámetro p_{one} se concluye que valores altos favorecen al rendimiento del algoritmo, por lo tanto, deciden crear las nuevas ideas utilizando sólo un cluster y no dos como en los algoritmos MBSO y BSO. Por otra parte, se demostró que al crear nuevas ideas utilizando la mejor idea del cluster o una idea aleatoria, los resultados eran favorecidos si se utilizaba la mejor idea con una probabilidad basada en una distribución Gaussiana con media = 0.4 y desviación estándar de 0.1, $p_{center} = N(0{,}4, 0{,}1)$ [29]. Por lo tanto, el algoritmo SMBSO es una version simplificada del algoritmo MBSO, en el cual se omite el operador de reemplazo $p_{replace}$, se crean las nuevas ideas a partir de un solo cluster y el parámetro p_{center} toma valores centrados en 0.4 con varianza de 0.1.

4.4. Diseño Experimental

Dos experimentos se llevaron a cabo. El primero de ellos consistió en analizar el desempeño de las tres versiones presentadas del algoritmo BSO con cada manejador de restricciones con el propósito de mostrar evidencia acerca de los efectos de un manejador de restricciones con una versión en particular del algoritmo BSO. De acuerdo a los resultados obtenidos en el

primer experimento un segundo experimento se llevó a cabo comparando las tres versiones del algoritmo BSO con el manejador de restricciones que reportó los mejores resultados en el primer experimento. El objetivo del segundo experimento fue obtener la mejor combinación de BSO/manejador de restricciones para ser utilizada posteriormente. Se aplicó la prueba estadística de suma de rango de Wilcoxon con 95 % de confianza a los resultados finales. Los resultados se obtuvieron después de hacer 30 ejecuciones independientes por cada función por cada variante. El conjunto de funciones presentado en la Tabla 4.1 fue utilizado como conjunto de prueba en este capítulo. Los valores para cada parámetro utilizado se presentan en la Tabla 4.2. El equipo de cómputo utilizado para llevar a cabo los experimentos cuanta con un sistema operativo *Windows 7 home premium*, un procesador Intel(R) Core(TM) i5 CPU M 460 2.53GHz con una memoria de 4G.

Considerando que el manejador de restricciones de jerarquización estocástica requiere un proceso de ordenamiento, se decidió combinar la población actual y la nueva en un solo conjunto para aplicar el método en cada variante del algoritmo y así mantener el tamaño de la población en cada iteración.

4.5. Resultados y Discusión

Los resultados de los experimentos son presentados en las Tablas 4.3-4.6, donde la primer columna identifica el nombre de la función, las siguientes tres columnas incluyen el resultado obtenido por la versión de BSO con cada manejador de restricciones ε-constrained (ε), Reglas de Factibilidad (FR) y jerarquización estocástica (SR). La última columna presenta el resultado de la prueba estadística (ST, por sus siglas en inglés Statistical Test) Rank Sum de Wilcoxon, donde (=) significa que no existe diferencia significativa entre el algoritmo de referencia y el algoritmo comparado, (+) significa que el algoritmo de referencia supera al algoritmo comparado y (-) significa que el algoritmo comparado supera al algoritmo de referencia. Un espacio vacío en las tablas (columnas 2-4) significa que el algoritmo no encontró soluciones factibles para el problema de prueba, en esos casos no se reportan pruebas estadísticas. Los mejores resultados son marcados en negritas y los problemas omitidos son aquellos en donde la versión de BSO no consiguió obtener soluciones factibles en ninguna ejecución.
En el experimento 1, las versiones de BSO con el método de ε-constrained fueron tomadas como algoritmo de referencia para realizar las pruebas estadísticas ya que fue la combinación que presentó mejores resultados con base en estadística descriptiva. En el experimento 2, la variante de MBSO fue utilizada como algoritmo de referencia por razones similares.

Tabla 4.1: Conjunto de funciones de prueba. CEC2006 . "n" es el número de variables del problema, "LI" es el número de restricciones de desigualdad lineal, "NI" es el número de restricciones de desigualdad no lineal, "LE" es el número de restricciones de igualdad lineal, "NE" es el número de restricciones de igualdad no lineal, "a" es el número de restricciones activas y "ρ" es el tamaño estimado de la región factible con respecto de todo el espacio de búsqueda [45]

Función	n	**Tipo de función**	ρ	**LI**	**NI**	**LE**	**NE**	a
g01	13	cuadrática	0.0003 %	9	0	0	0	6
g02	20	no lineal	99.9973 %	2	0	0	0	1
g03	10	no lineal	0.0026 %	0	0	0	1	1
g04	5	cuadrática	27.0079 %	4	2	0	0	2
g05	4	no lineal	0.0000 %	2	0	0	3	3
g06	2	no lineal	0.0057 %	0	2	0	0	2
g07	10	cuadrática	0.0000 %	3	5	0	0	6
g08	2	no lineal	0.8581 %	0	2	0	0	0
g09	7	no lineal	0.5199 %	0	4	0	0	2
g10	8	lineal	0.0020 %	6	0	0	0	6
g11	2	cuadrática	0.0973 %	0	0	0	1	1
g12	3	cuadrática	4.7697 %	0	1	0	0	0
g13	5	no lineal	0.0000 %	0	0	1	2	3
g14	10	no lineal	0.0000 %	0	0	3	0	3
g15	3	cuadrática	0.0000 %	0	0	1	1	2
g16	5	no lineal	0.0204 %	4	34	0	0	4
g17	6	no lineal	0.0000 %	0	0	0	4	4
g18	9	cuadrática	0.0000 %	0	13	0	0	6
g19	15	no lineal	33.4761 %	0	5	0	0	0
g20	24	lineal	0.0000 %	0	6	2	12	16
g21	7	lineal	0.0000 %	0	1	0	5	6
g22	22	lineal	0.0000 %	0	1	8	11	19
g23	9	lineal	0.0000 %	0	2	3	1	6
g24	2	lineal	79.6556 %	0	2	0	0	2

4.5.1. Resultados y Discusión: Experimento 1

De acuerdo a los resultados obtenidos para la variante de BSO en la Tabla 4.3, la prueba estadística indica un empate entre la versión de ε y FR ya que resultados similares fueron obtenidos en 15 de las 19 funciones de prueba donde se obtuvieron soluciones factibles. ε superó a FR sólo en uno de los problemas mientras que FR lo superó en dos de ellos. Por otra parte, ε fue claramente mejor que SR dado que ε encontró mejores resultados en 15 funciones, y solo en una función fue mejor SR, obteniendo resultados similares en dos funciones de prueba.

En la Figura 4.1 se presenta la gráfica de convergencia para las tres versiones de BSO en el problema de prueba representativo g01. Como se puede observar, ε es capaz de evitar óptimos locales presentando mejora

Tabla 4.2: Configuración de parámetros. Los valores de los parámetros fueron tomados de las referencias originales de cada versión de BSO [62], [63], [29] y también de las referencias de los manejadores de restricciones [56, 49].

Algoritmo	Parámetros	Valor
BSO, MBSO, SMBSO	NP	100
	M	5
	Evaluaciones	500,000
BSO	k	20
	μ	0
	σ	1
MBSO	pr	0.005
SMBSO	$p_{oneCenter}$	N(0.4,0.1)
BSO , MBSO	$p_{replace}$	0.2
	p_{one}	0.8
	$p_{oneCenter}$	0.4
	$p_{twoCenter}$	0.5
ε-constrained	cp	0.5
stochastic ranking	Pf	0.45
Test Problems	CEC2006 [39]	

durante todo el proceso de evolución. Este comportamiento contrasta del obtenido por los otros dos manejadores de restricciones que presentan un comportamiento de convergencia prematura.

Respecto a MBSO, los resultados estadísticos presentados en la Tabla 4.4 indican que ε superó a FR en 11 de 22 funciones en las que se obtuvieron soluciones factibles, obteniendo resultados similares en 9 y mostrando que FR no logró superarlo en ninguno de ellos. SR fue mejor en 2 funciones de prueba siendo superado por ε en 13 y obteniendo resultados similares en 5 de las 22 funciones.

La gráfica de convergencia para las tres versiones de MBSO se muestran en la Figura 4.2 para el problema representativo g01. ε presenta la mejor convergencia y a su vez obtiene los mejores resultados. Por otra parte, FR y SR requirieron de mayor tiempo para obtener los resultados mostrados. Únicamente se presentan las primeras 250 generaciones para poder identificar las diferencias en cada version.

Finalmente, en la Tabla 4.5, la variante SMBSO presenta un patrón similar a los resultados reportados por la variante MBSO, donde ε supera a FR en 13 de 21 problemas de prueba en los que se obtuvieron soluciones factibles, presentando resultados similares en 5 y empeora sólo en una función de prueba. Por otra parte, ε superó a SR en 15 funciones de prueba mostrando resultados similares en 4. SR no logró superar a ε en ninguna función de prueba.

Tabla 4.3: Valor de la Mediana reportada por el algoritmo BSO con cada manejador de restricciones. ε es tomado como referencia sobre FR y SR para mostrar los resultados de la prueba estadística (ST).

		BSO		
Función	ε	**FR**	**SR**	**ST**
g01	**-13.7259834**	-13.4356792	-12.1210574	(=)(+)
g02	-0.78487125	**-0.78544357**	-0.75745431	(=)(+)
g03	-0.28321891	-0.21301742	**-0.28573573**	(=)(=)
g04	**-30665.2927**	-30665.0269	-30653.4706	(=)(+)
g05	5294.00651	**5178.08668**		
g06	-6961.46576	**-6961.7764**	-6939.50205	(-)(+)
g07	**24.6005362**	24.6343637	27.2056687	(=)(+)
g08	-0.09582504	-0.09582504	-0.09582504	(=)(=)
g09	680.666694	**680.663085**	681.246018	(=)(+)
g10	**13318.7426**	14411.6122	18149.3709	(-)(+)
g11	**0.74997285**	0.7499743	0.75171115	(=)(+)
g12	-1	-1	-1	(=)(+)
g13	**0.61567416**	0.85296517	0.99999849	(+)(+)
g14	-43.6473924	**-43.6896006**	-42.7105425	(=)(-)
g15	**961.729572**	961.910793	967.332323	(=)(+)
g16	**-1.90423112**	-1.90402015	-1.88560212	(=)(+)
g18	-0.86579247	**-0.86599032**	-0.820099	(=)(+)
g19	**62.1777691**	63.8393044	84.1350502	(=)(+)
g23		**-97.9634101**		
g24	-5.50801327	-5.50801327	-5.50656964	(=)(+)

Un comportamiento muy similar al observado por la variante MBSO se presenta en la variante SMBSO en la Figura 4.3 para el problema de prueba g01. ε presentó mejor convergencia que las otras variantes, su gráfica es la que se encuentra pegada al eje de las x en la Figura 4.3. Sólo se presentan las primeras 250 generaciones para poder identificar las diferencias entre versiones.

Un descubrimiento importante en el experimento 1 es que el manejador de restricciones ε fue el más competitivo en las tres variantes del algoritmo BSO. Además, de favorecer al algoritmo BSO en sus tres versiones con una rápida convergencia obteniendo resultados competitivos. Este comportamiento se atribuye a las características del manejador de restricciones, que al permitir soluciones ligeramente no factibles en la población mediante una tolerancia ε que va disminuyendo hasta acercarse a cero favorece en mayor grado a la exploración del espacio de búsqueda en las primeras generaciones del algoritmo. Ésto hace que se cubra una mayor parte del espacio de búsqueda abriendo la posibilidad de encontrar zonas prometedoras con mayor

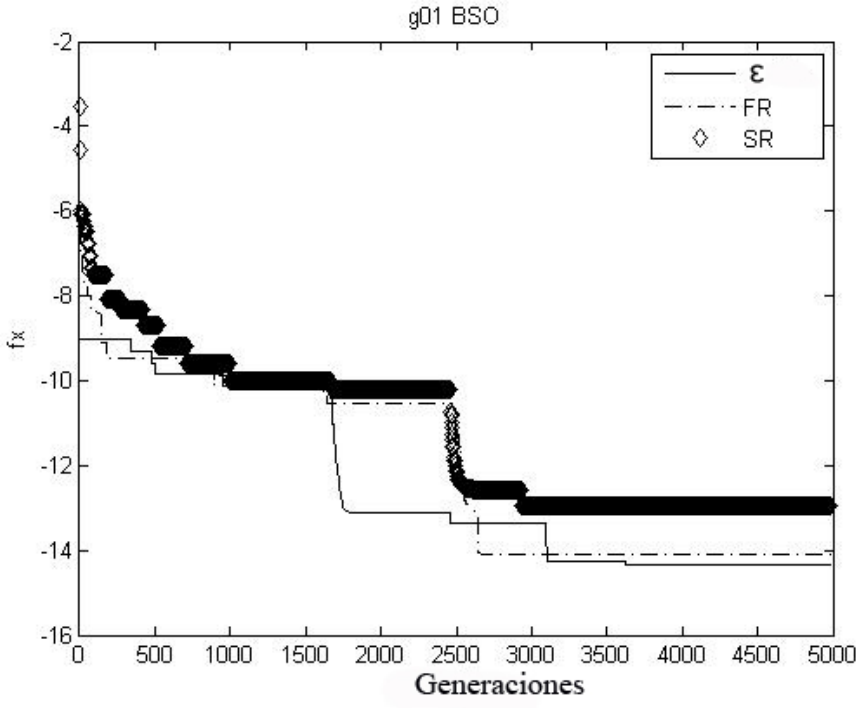

Figura 4.1: Gráfica de convergencia para la función g01 usando BSO con los tres manejadores de restricciones.

rapidez en comparación con los otros manejadores utilizados donde se ejerce una mayor presión de selección. Por lo tanto, en el siguiente experimento se utilizará ε como manejador de restricciones en las tres versiones del algoritmo BSO con el fin de obtener la versión más competitiva.

Tabla 4.4: Valor de la Mediana reportada por el algoritmo MBSO con cada manejador de restricciones. ε es tomado como referencia sobre FR y SR para mostrar los resultados de la prueba estadística (ST).

		MBSO		
Función	ε	**FR**	**SR**	**ST**
g01	-15	-15	-15	(=)(=)
g02	**-0.72905855**	-0.72195232	-0.70338773	(=)(=)
g03	**-1.00029549**	-0.99932567	-0.99689488	(+)(+)
g04	-30665.5387	-30665.5387	-30665.5387	(=)(-)
g05	**5126.57055**	5182.01052	5220.21686	(+)(+)
g06	-6961.81388	-6961.81388	-6961.81387	(=)(+)
g07	**24.3420728**	24.4072619	24.5270199	(+)(+)
g08	-0.09582504	-0.09582504	-0.09582504	(=)(=)
g09	**680.633326**	680.639841	680.652859	(+)(+)
g10	**7148.88021**	7267.76964	7327.29121	(=)(+)
g11	**0.74990035**	0.7546776	0.76300509	(+)(+)
g12	-1	-1	-1	(=)(=)
g13	**0.05395068**	0.95770084	0.88762401	(+)(+)
g14	**-47.6011839**	-43.3648135	-42.9230471	(+)(+)
g15	**961.716101**	964.030798	964.857912	(+)(+)
g16	-1.90515526	-1.90515526	-1.90515526	(=)(+)
g17	**8853.57277**	8957.76168	8957.47279	(+)(+)
g18	**-0.8658459**	-0.8643457	-0.86453845	(+)(-)
g19	**33.1293739**	36.4198197	40.4967993	(+)(+)
g21		329.575325		
g23		78.705459	242.962969	
g24	-5.50801327	-5.50801327	-5.50801327	(=)(=)

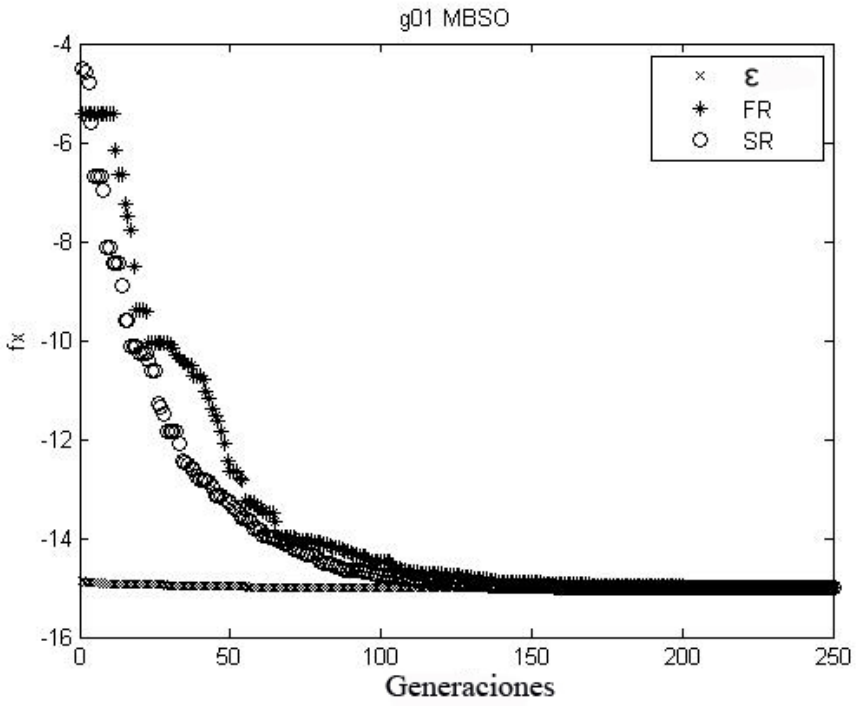

Figura 4.2: Gráfica de convergencia para la función g01 usando MBSO con los tres manejadores de restricciones.

Tabla 4.5: Valor de la Mediana reportada por el algoritmo SMBSO con cada manejador de restricciones. ε es tomado como referencia sobre FR y SR para mostrar los resultados de la prueba estadística (ST).

		SMBSO		
Función	ε	**FR**	**SR**	**ST**
g01	**-14.9999992**	-15	-14.9993215	(=)(+)
g02	-0.67133279	**-0.71850795**	-0.69465066	(-)(=)
g03	**-1.00027242**	-0.99144829	-0.67607676	(+)(+)
g04	**-30665.5387**	-30665.431	-30654.787	(+)(+)
g05	**5126.75765**	5205.27027	5346.15283	(+)(+)
g06		-6659.64967	-6619.74874	
g07	**24.3630671**	24.5280728	25.1630264	(+)(+)
g08	-0.09582504	-0.09582504	-0.09582504	(+)(+)
g09	**680.636058**	680.649006	680.71517	(+)(+)
g10	7322.8129	**7309.84216**	7525.8942	(=)(=)
g11	**0.7499004**	0.75636641	0.76383497	(+)(+)
g12	-1	-1	-1	(=)(=)
g13	**0.05394833**	0.86163753	0.92297506	(+)(+)
g14	**-47.6608268**	-41.9986648	-42.523399	(+)(+)
g15	967.519963	965.925127	**965.770638**	(=)(=)
g16	**-1.90514953**	-1.90454387	-1.83348496	(+)(+)
g17	**8855.91728**	8954.23605	8955.62533	(+)(+)
g18	**-0.8658068**	-0.86331758	-0.77768727	(=)(+)
g19	**33.7540853**	39.8877045	52.0551162	(+)(+)
g23		119.152469	145.598656	
g24	**-5.50801327**	-5.50798652	-5.5002577	(+)(+)

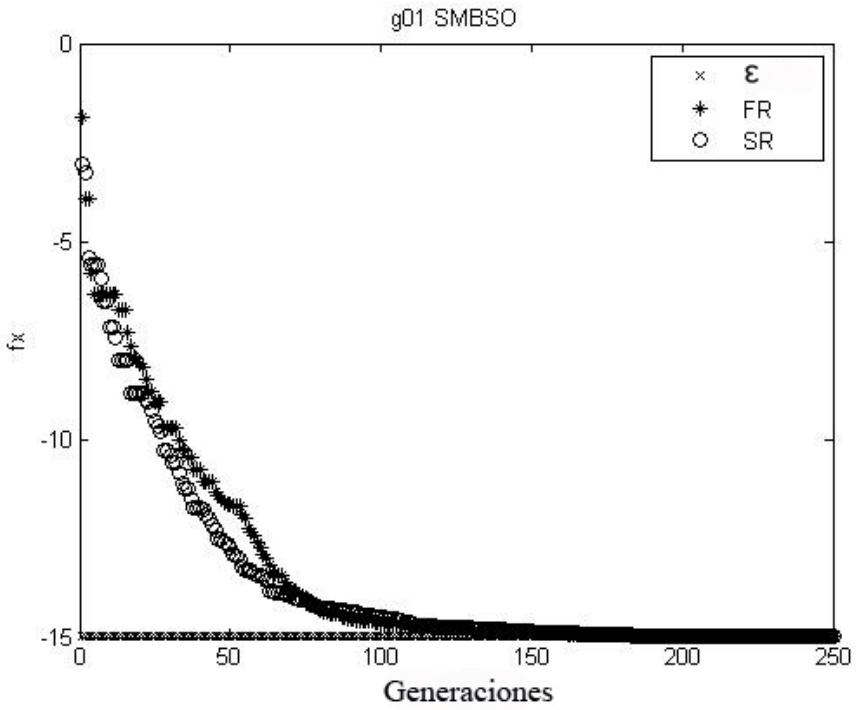

Figura 4.3: Gráfica de convergencia para la función g01 usando SMBSO y los tres manejadores de restricciones.

Tabla 4.6: Valor de las Medianas obtenidas por MBSO, SMBSO and BSO usando ε-constrained como manejador de restricciones. MBSO es tomando como referencia sobre SMBSO y BSO para mostrar los resultados estadísticos.

	MBSO, SMBSO y BSO			
Función	**MBSO**	**SMBSO**	**BSO**	**ST**
g01	**-15**	-14.99999925	-13.72598342	(+)(+)
g02	-0.729058554	-0.671332787	**-0.784871252**	(+)(-)
g03	**-1.000295492**	-1.000272424	-0.283218907	(=)(+)
g04	-30665.53867	**-30665.53867**	-30665.29271	(+)(+)
g05	**5126.570551**	5126.757652	5294.006507	(=)(+)
g06	**-6961.813876**		-6961.46576	
g07	**24.34207279**	24.36306708	24.60053615	(=)(+)
g08	-0.095825041	-0.095825041	-0.095825041	(+)(=)
g09	**680.6333264**	680.6360583	680.6666943	(=)(+)
g10	**7148.880214**	7322.812904	13318.74261	(=)(+)
g11	**0.749900347**	0.749900398	0.749972845	(=)(+)
g12	-1	-1	-1	(=)(=)
g13	0.053950678	**0.053948325**	0.615674157	(=)(+)
g14	-47.60118387	**-47.6608268**	-43.64739239	(=)(+)
g15	**961.7161006**	967.5199633	961.7295719	(+)(=)
g16	**-1.905155259**	-1.905149534	-1.904231122	(+)(+)
g17	**8853.572769**	8855.917276		
g18	**-0.865845899**	-0.8658068	-0.865792468	(=)(=)
g19	**33.12937392**	33.75408534	62.17776907	(=)(+)
g24	-5.508013272	-5.508013272	-5.508013272	(+)(=)

4.5.2. Resultados y Discusión: Experimento 2

La Tabla 4.6 presenta la comparación de las tres versiones del algoritmo BSO con ε como manejador de restricciones. La version MBSO se toma como referencia para realizar la prueba estadística de Wilcoxon en la cual se observa que MBSO supera la versión SMBSO en 7 problemas de prueba, alcanzando resultados similares en 11. Además, MBSO supera a la version BSO en 12 funciones de prueba obteniendo resultados similares en 5 y siendo superado por BSO sólo en 1 de los problemas de prueba.

La Figura 4.4 presenta la gráfica de convergencia para las tres versiones del algoritmo BSO con ε como manejador de restricciones en la función representativa g01. Como se puede observar las versiones MBSO y SMBSO presentan una rápida convergencia obteniendo resultados competitivos (Su gráfica se presenta cercana al eje de las x). En contraste, BSO necesita mucho más tiempo para converger quedando atrapado en óptimos locales.

De acuerdo a los resultados finales presentados en la Tabla 4.6 la combi-

nación MBSO y ε como manejador de restricciones presentaron los mejores resultados en el conjunto de problemas de prueba utilizados en este trabajo. Por otra parte, se observa que la versión SMBSO, siendo la versión más simple con menos parámetros que calibrar por el usuario, presentó un comportamiento muy competitivo en este tipo de problemas, por lo tanto, la convierte en centro de atención para un análisis posterior.

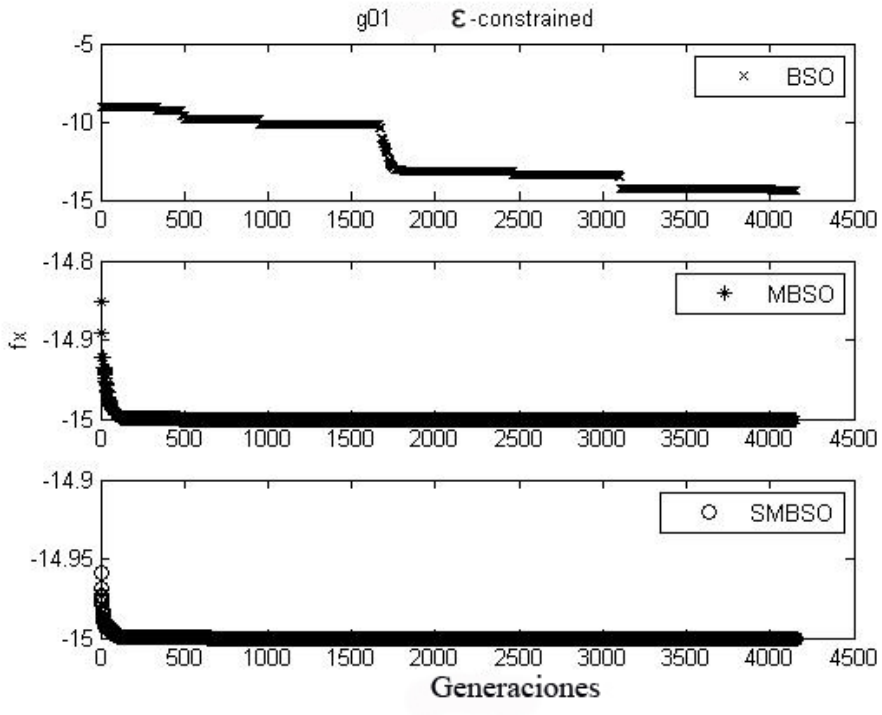

Figura 4.4: Gráfica de convergencia para la función g01 usando ε-constrained como manejador de restricciones en las tres versiones del algoritmo BSO.

4.6. Conclusiones

En este capítulo se hizo mención al algoritmo BSO, el cual es inspirado en el proceso de tormenta de ideas utilizado por el ser humano para dar solución a problemas que difícilmente pudieran ser solucionados de forma individual. BSO ha sido un algoritmo bastante explorado en espacios sin restricciones pero no así en espacios con restricciones, por tal motivo, en este capítulo se habilitó al algoritmo BSO para explorar espacios restringidos con la ayuda de los manejadores de restricciones. Tres técnicas para el manejo de restricciones fueron utilizadas en este estudio con el propósito de analizar sus efectos sobre tres versiones recientes del algoritmo en problemas de optimización

numérica con restricciones. Se realizaron dos experimentos. En el primero, se analizó el efecto de los tres manejadores de restricciones sobre cada versión del algoritmo BSO. De los resultados obtenidos se presenta a ε-constrained como el manejador de restricciones más competitivo en las tres versiones de BSO sujetas de estudio. Tomando como base los resultados obtenidos en el experimento 1, en el experimento 2 se consideró al manejador de restricciones ε-constrained con las tres versiones del algoritmo BSO. Los resultados finales muestran a la version MBSO como la más competitiva quedando la versión SMBSO en segundo lugar. Los resultados indican que algoritmos como BSO requieren de un manejador de restricciones como ε-constrained que permita soluciones ligeramente no factibles como factibles dentro de la población para obtener mejores resultados finales.

El algoritmo MBSO con el manejador de restricciones ε-constrained será el algoritmo base en los siguientes capítulos por mostrar una mayor competitividad en el conjunto de prueba utilizado. En el siguiente capítulo se reforzará este algoritmo con operadores especiales de frontera con el fin de guiar la búsqueda hacia la frontera de la región factible y poder localizar los puntos óptimos que pudieran encontrarse en esta área de interés.

Capítulo 5

Explorando el límite de las restricciones con operadores especiales de frontera

La existencia de restricciones activas así como la división del espacio de búsqueda en regiones factible y no factible son obstáculos a vencer en optimización numérica con restricciones. Así mismo, las restricciones activas incrementan la posibilidad de que puntos óptimos sean localizados sobre el límite de tales restricciones, haciendo necesario la adición de métodos especiales que permitan explorar esta área [3], [38], [51].

En los últimos años, los NIAs han sido exitosamente aplicados en la solución de este tipo de problemas. Aunque inicialmente no fueron pensados para tratar con restricciones, la adición de una o más técnicas para el manejo de las mismas les ha permitido ser exitosos en espacios restringidos [15], [42],[45]. Hoy en día existe una gran variedad de técnicas para el manejo de restricciones, entre las cuales, los operadores especiales, conocidos como operadores de frontera, son sujetos de estudio en este capítulo. El principal objetivo es concentrar las soluciones en la frontera de una o todas las restricciones asociadas al problema. La literatura especializada reporta el éxito obtenido cuando un operador de frontera es combinado con un NIA, algunos ejemplos de ésto se describen a continuación.

En 1996, Schoenauer y Michalewicz, [51] propusieron dos operadores de cruza especial (cruza esférica y cruza geométrica) para muestrear el límite la restricción en dos problemas específicos usando un Algoritmo Genético en codificación binaria. Después de un proceso de inicialización ad-hoc, en la cual los puntos son colocados sobre el límite de la superficie de la región factible, la cruza es aplicada a cada solución para preservar a los nuevos puntos en el límite de cada restricción. A pesar del éxito de esta combinación, la principal desventaja es el proceso de inicialización ad-hoc para un problema en particular. En 2006 un nuevo operador de frontera, Boundary Search (BS)

fue publicado por Leguizamón y Coello-Coello [38], en el cual a través de dos puntos (uno factible y uno no factible) y operaciones matemáticas simples se intenta guiar la búsqueda hacia el límite de la región factible de la restricción. Una búsqueda binaria es aplicada a los parámetros de entrada obteniendo un punto medio con el propósito de que el nuevo punto permanezca sobre la frontera de la restricción actual. Si éste es el caso, el proceso termina, de lo contrario, el nuevo punto es sustituido por el punto factible o no factible según corresponda y el proceso continúa. El algoritmo basado en Colonia de Hormigas (ACO) fue utilizado para actualizar las direcciones de búsqueda en el proceso de búsqueda global. El inconveniente en este operador es la necesidad de puntos factibles y el uso de una política para decidir sobre cual restricción es aplicado en caso de existir más de una en el problema actual. En 2014, Bonyady y Michalewicz [3], propusieron un método conocido como Subset Constraints Boundary Narrower (SCBN), capaz de ajustar el área factible de un problema restringido a su frontera definiendo el grosor de ésta área a través de un parámetro σ, permitiendo al algoritmo controlar la cercanía al área de interés. El algoritmo basado en partículas (PSO) fue utilizado como algoritmo de búsqueda global. Pobres resultados fueron obtenidos cuando el parámetro σ tomaba valores cercanos a cero y mejoraban al utilizar valores altos de σ. Por lo tanto, sugieren un ajuste adaptativo o auto-adaptativo del parámetro σ como trabajo futuro.

Como se muestra en la literatura, un gran potencial es generado por la combinación de algoritmos inspirados en la naturaleza y operadores de frontera. Siguiendo con esta línea de investigación en este capítulo se propone mejorar el rendimiento del algoritmo MBSO [6] con el uso de operadores especiales de frontera. De esta manera se habilita al algoritmo MBSO para explorar la frontera de la restricción con mayor grado de violación en el punto actual. Se proponen además, dos nuevos operadores de frontera, el primero de ellos (BS-r), inspirado en el método algebraico de la división de un segmento de línea en un determinado radio y el segundo (BS-G), basado en la información del gradiente de la restricción. El desempeño del algoritmo es analizado en el conjunto de treinta y seis funciones de prueba propuesto en el Congreso de Cómputo Evolutivo del 2010 (CEC2010) [40].

5.1. Operador de frontera Boundary Search (BS)

En 2009 Guillermo Leguizamón y Carlos. A. Coello-Coello [38] presentan un operador de frontera denominado Boundary Search (BS). BS puede ser utilizado tanto por algoritmos evolutivos como por algoritmos de inteligencia colectiva, además de estar diseñado para ser aplicado a diferentes problemas de optimización con restricciones. Aunque puede ser aplicado a cualquier tipo de restricciones, al ser un operador de frontera se espera mayor éxito en problemas con restricciones activas. El pseudocódigo de BS se presenta en el

Algoritmo 12.

Algoritmo 12 Algoritmo BS

1: x, un punto factible;
2: y,un punto no factible;
3: m = puntoMedio(x,y) *//Obtener el punto medio entre* x, y
4: **while** $g(m) \neq 0$, o, dist-frontera(m) $> \delta$ **do**
5: **if** factible(m) **then**
6: $x = m$
7: **else**
8: $y = m$;
9: **end if**
10: m = puntoMedio(x,y)
11: **end while**
12: regresa m

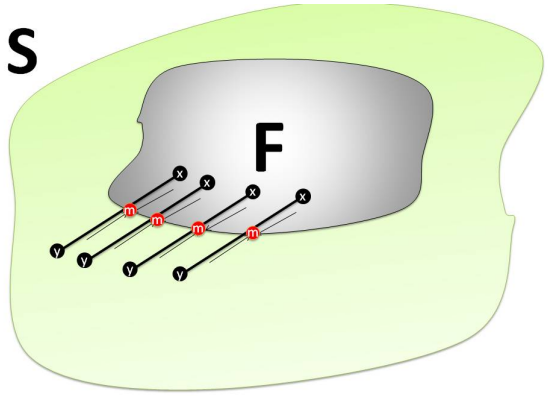

Figura 5.1: Operador de frontera BS.

A partir de dos puntos x, y, donde x es un punto factible, y es un punto no factible (ambos en la restricción actual), se aplica una búsqueda binaria donde se obtiene el punto medio m entre los puntos x, y. m es evaluado y si resulta un punto sobre o muy cerca de la frontera de la región factible, el método termina, de lo contrario entra en un ciclo hasta que m se encuentre lo mas cerca posible a la frontera de la restricción, ver Figura 5.1.

Dos reglas son necesarias para decidir si un punto se encuentra en la frontera, las cuales se mencionan a continuación:

1. Si la restricción i es una restricción de desigualdad, el punto m estará en la frontera si y solo si (-$\delta \leq g_i(m) \leq 0$),

2. Si la restricción j es una restricción de igualdad, el punto m indicará que se encuentra en la frontera si y solo si $(0 \leq |h_j(m)| \leq \delta)$.

Donde δ es un parámetro establecido por el usuario con un valor muy cercano a cero, que indicará qué tan cercano se encuentra el punto m del objetivo, en este caso, la frontera de la región factible.

La simplicidad del método y su buen desempeño en problemas restringidos son su principal atracción [38]. Aunque como todo método, tiene sus desventajas, y una de las principales es el requerimiento de puntos factibles, que, de entrada, obtenerlos se convierte en un problema de optimización en espacios restringidos. Por otra parte, cuando más de una restricción es asociada al problema se debe establecer una política adecuada para decidir sobre qué restricción se aplicará el operador. En este sentido, el autor menciona las siguientes:

- De existir restricciones activas, iniciar con dichas restricciones.
- Aplicar el operador a una restricción en turno durante un cierto número de iteraciones.
- Aplicar el operador a una restricción en turno, pero sólo contemplar las restricciones activas.

Además de estas opciones en [5], se propuso aplicar el operador BS a la restricción más difícil hasta el momento, es decir, la restricción con mayor grado de violación en el punto actual, fuese o no una restricción activa, obteniendo resultados competitivos.

5.2. Operador de frontera basado en el método de división de segmento de línea (BS-r)

Si tamaños de paso variables, largos o cortos, pudieran utilizarse para alcanzar la frontera de las restricciones, cabe entonces la posibilidad de disminuir el número de evaluaciones utilizados en dicho objetivo. Basados en esa posibilidad un nuevo operador de frontera es explorado en este capítulo. El nuevo operador está diseñado con base en el método de división de un segmento de línea propuesto en Geometría Analítica [24]. A partir de dos puntos de entrada el método traza un segmento de línea en la cual se intenta encontrar las coordenadas de un punto C que divide la línea en un determinado radio r. Si el par de puntos de entrada (A, B) en el operador BS son tomados para dibujar el segmento de línea y el tamaño de paso se convierte en el valor de r utilizado para generar el nuevo punto C, entonces, es posible alcanzar la frontera de la restricción utilizando menor número de evaluaciones con valores cortos o largos de r en lugar de solo generar puntos medios.

Las coordenadas del punto de interés C estarán dadas por las Ecuación (5.1) y (5.2).

$$\frac{\overline{AC}}{\overline{AB}} = \frac{x - x_1}{x_2 - x_1} = r \qquad \frac{\overline{AC}}{\overline{AB}} = \frac{y - y_1}{y_2 - y_1} = r \tag{5.1}$$

$$x = x_1 + r(x_2 - x_1) \qquad y = y_1 + r(y_2 - y_1) \tag{5.2}$$

donde, A, B son dos puntos extremos en la línea y $(x1, y1), (x2, y2)$ sus respectivas coordenadas; C es el punto no conocido en la línea y (x, y) sus coordenadas particulares; r es el radio en el cual el segmento de línea es dividido. Si r es 0 o 1, entonces, C sería el punto A o B, respectivamente; de otra manera, si r toma un valor de 0.5, entonces, C viene a ser el punto medio entre A and B.

El nuevo operador de frontera conocido como BS-r intenta acercarse a la frontera de las restricciones reduciendo el espacio de búsqueda al tamaño de la linea generada por los dos puntos de entrada A, B en cada iteración. En la Figura 5.2 se presenta de forma muy general el comportamiento del operador BS-r. Una vez que se tienen los puntos de entrada A y B donde A es un punto no factible y B es un punto factible, se procede a explorar el segmento de linea generado por dichos puntos de tal manera que los puntos generados se aproximen lo más posible a la región de interés, en este caso en el área delimitada por el circulo rojo punteado. Si un punto d se genera partiendo de un valor en r de 0.5 se estaría en el caso mostrado en la Figura 5.2. El siguiente paso es evaluar si el punto d es un punto factible o no factible, como en este caso el punto d es no factible pero con menor grado de violación que el punto A, entonces, automáticamente el punto A se convierte en el punto d quedando un espacio de búsqueda delimitado entre el punto d, B. Ahora parece más cercana la frontera de la restricción si se aproxima la búsqueda por el punto A en lugar de aproximarla por el punto B. En este caso puntos generados con valores de r menores a 0.5 se aproximarían mejor al objetivo del operador.

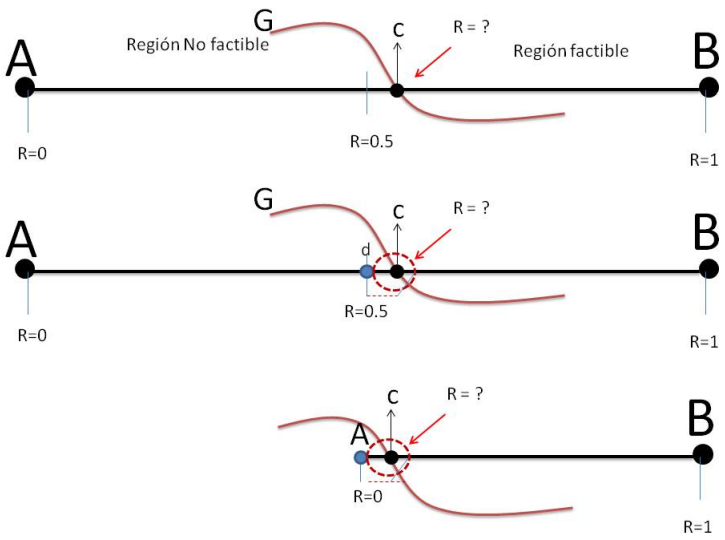

Figura 5.2: Operador de frontera BS-r.

5.3. Operador de frontera utilizando el gradiente de la restricción (BS-G)

Alcanzar la frontera de la restricción partiendo de puntos factibles no siempre es lo mas conveniente ya que en este tipo de problemas generar puntos factibles no siempre es posible. En este apartado se presenta una nueva forma de explorar la frontera de la restricción partiendo de puntos no factibles. El nuevo operador de frontera BS-G, intenta acercar un punto no factible a otro punto el cual se encuentre muy cerca de la frontera factible de la restricción tomando la idea de algoritmos de proyección.

El operador parte de identificar la restricción con mayor grado de violación para calcular su gradiente. Tomando en cuenta el gradiente de la restricción $\nabla g_i(x)$, su grado de violación $g_i(x)$ y la longitud del gradiente $\|\nabla g_i(x)\|$ se procede a construir un vector conocido como vector de factibilidad fv que será utilizado para alcanzar el punto más cercano a la frontera de la restricción. fv es construido a partir de la Ecuación (5.3).

$$fv = \frac{-g_i(x)\nabla g_i(x)}{\|\nabla g_i(x)\|^2} \tag{5.3}$$

Una vez que el vector de factibilidad es construido, este es usado para modificar el punto actual. El método es aplicado durante μ iteraciones o hasta que el nuevo punto sea factible o se encuentre cerca de la frontera de la restricción. El operador de frontera basado en gradiente se presenta en el Algoritmo (13).

Algoritmo 13 Operador de frontera basado en gradiente: BS-G

1: y, punto no factible;
2: **while** no se cumpla una condición de paro **do**
3: calcular fv
4: **if** $g(y) \neq 0$, o, dist-frontera$(y) > \delta$ **then**
5: regresa y
6: **else**
7: $y = y + fv$;
8: **end if**
9: **end while**

Los operadores de frontera BS, BS-r o BS-G, se convierten en el primer paso en el ciclo del Algoritmo 14, MBSO-BS. De esta manera, el algoritmo BSO no esta limitado a trabajar directamente con el operador, sino que, indirectamente MBSO utiliza las bondades del operador de frontera para alcanzar las mejores soluciones en los problemas de prueba. Antes de aplicar los operadores de frontera BS o BS-r, se debe cumplir la condición de soluciones factibles y no factibles dentro de la población actual, paso 3 el Algoritmo 14, si esta condición no se cumple los operadores no pueden aplicarse. Si en la población actual solo existen soluciones no factibles, entonces, se aplica el operador BS-G paso 7, con la finalidad de aumentar el número de soluciones factibles en la población. Una vez que las soluciones de entrada han sido procesados por el operador especial, éstas reemplazarán a las peores soluciones de la población actual, paso 9. A partir del paso 10, el algoritmo sigue su flujo normal partiendo de una población preprocesada por los operadores especiales.

Algoritmo 14 Algoritmo MBSO-BS

1: Aleatoriamente generar NP ideas y evaluarlas;
2: **while** No se cumpla la condición de paro **do**
3: Seleccionar P pares de puntos(factibles y no factibles) de N
4: **if** P es no vacío **then**
5: Aplicar BS o BS-r a P, mp = BS(P)
6: **else**
7: Aplicar BS-G sólo a los puntos P no factibles; mp = BS-G(P)
8: **end if**
9: Reemplazar los peores P puntos de la población actual por los mp puntos encontrados
10: *// operador de agrupamiento*
11: *// operador de reemplazo*
12: *// Operador de cruza*
13: **end while**

5.4. Diseño Experimental

Los valores de parámetros utilizados en este trabajo se presentan en la Tabla 5.1 donde P se refiere al máximo de puntos (factible, no factible) de entrada en el operador de frontera. El gradiente se calcula numéricamente como $\frac{g(x+\Delta)-g(x)}{\Delta}$ donde Δ es igual a 0.0001. Se aplicaron pruebas estadística de Wilcoxon a los resultados finales para hacer la comparación entre dos algoritmos. Cuando existe la posibilidad de comparar más de dos variantes o algoritmos, se utiliza la prueba post-hoc de Bonferroni. Los mejores resultados en cada tabla se presentan remarcadas en negritas.

Tabla 5.1: Configuración de parámetros.

MBSO	NP = 100, M =5, $p_{replace}$ = 0.2, p_{one} = 0.8, $p_{oneCenter}$ = 0.4, $p_{twoCenter}$ = 0.5, $Max_{iter} = \frac{Max_{fes}}{NP}$
ε−Constrained	cp=5, TC = 0.2*Max_{iter}, θ = 0.2NP
Max_{fes}	200,000 , 600,000
P	5
BS-G	μ = 10
BS-r	r=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]

El conjunto de treinta y seis funciones con 10 y 30 dimensiones propuesto en [40] es utilizado en este trabajo. Las características de cada función se pueden observar en la Tabla 5.2.

Para analizar el desempeño tanto del operador de frontera propuesto en [38], como los propuestos en este capítulo se realizaron tres experimentos, los cuales se definen a continuación.

1. **Experimento 1**. Desempeño del operador de frontera BS. Se analiza el rendimiento del operador de frontera BS [38] en combinación con el algoritmo MBSO [63] en el conjunto de funciones de prueba presentados.

2. **Experimento 2**. Desempeño del operador de frontera BS-r. En el segundo experimento, la versión del operador de frontera basado en la división del segmento de línea (BS-r) se combina con el algoritmo MBSO. El objetivo final es acercarse a la frontera de las restricciones utilizando un menor número de evaluaciones, por ello, se considera que variando el tamaño de paso a largo y/o corto podría acercarnos a este objetivo. En este operador el tamaño de paso estará dado por el valor dado al parámetro r, pero como el parámetro r puede tomar valores entre 0 y 1, no se conoce aún el valor de r que debe usarse para lograr este fin, es por ello que se probaron nueve versiones similares de BS-r donde el valor de r fue la variante en cada versión tomando valores entre 0.1 y 0.9.

Tabla 5.2: Conjunto de funciones de prueba CEC2010 con "$10D, 30D$" es el número de variables del problema, "E" es el número de restricciones de igualdad, "I" es el número de restricciones de desigualdad, "ρ" es el tamaño estimado de la región factible con respecto de todo el espacio de búsqueda. Tipo de función, "S" Separable, "NS" No Separable, "R" Rotada

F	**Rango de búsqueda**	**Tipo de función**	**Restricciones**		**Región factible ρ**	
			E	**I**	**10D**	**30D**
C01	$[0, 10]^D$	NS	0	2 NS	0.997689	1
C02	$[-5{,}12, 5{,}12]^D$	S	1S	2 NS	0	0
C03	$[-1000, 1000]^D$	NS	1S	0	0	0
C04	$[-50, 50]^D$	S	2 NS, 2 S	0	0	0
C05	$[-600, 600]^D$	S	2 S	0	0	0
C06	$[-600, 600]^D$	S	2 R	0	0	0
C07	$[-140, 140]^D$	NS	0	1 S	0.505123	0.503725
C08	$[-140, 140]^D$	NS	0	1 R	0.379512	0.375278
C09	$[-500, 500]^D$	NS	1 S	0	0	0
C10	$[-500, 500]^D$	NS	1 R	0	0	0
C11	$[-100, 100]^D$	R	1 NS	0	0	0
C12	$[-1000, 1000]^D$	S	1 NS	1 S	0	0
C13	$[-500, 500]^D$	S	0	2 S, 1 NS	0	0
C14	$[-1000, 1000]^D$	NS	0	3 S	0.003112	0.006123
C15	$[-1000, 1000]^D$	NS	0	3 R	0.00321	0.006023
C16	$[-10, 10]^D$	NS	2 S	1 S, 1 NS	0	0
C17	$[-10, 10]^D$	NS	1 S	2 NS	0	0
C18	$[-50, 50]$	NS	1 S	1 S	0.00001	0

3. **Experimento 3**. Desempeño del operador de frontera BS-G. Finalmente, la versión basada en gradiente es agregada al algoritmo MBSO de dos formas diferentes. En la primera versión, (MBSO-BS-CG), el método se aplica al punto medio obtenido después de aplicar el operador original BS durante μ iteraciones o mientras el punto generado por el gradiente sea mejorado. En la segunda versión (MBSO-BS-or-G), el método es aplicado sólo si los parámetros de entrada en el operador BS no están presentes en la población actual. En este caso, la condición de paro son μ iteraciones o cuando el punto generado se encuentra ya muy cerca de la frontera de la restricción. Cuando la condición de paro son μ iteraciones, entonces, de los μ puntos generados por BS-G, el mejor de éstos es el reportado como más cercano a la frontera factible de la restricción.

Si el problema tiene más de una restricción, el operador de frontera es aplicado a la restricción más difícil de satisfacer hasta el momento. Siendo esta restricción aquélla con menor cantidad de puntos factibles.

Los resultados finales reportados por Takahama en [56] son tomados como

base de comparación para medir el desempeño de la propuesta.

5.5. Resultados y Discusión

5.5.1. Experimento 1. Desempeño del operador de frontera BS

Los resultados obtenidos del mejor valor en f($\vec{x}$), la media y la desviación estándar después de combinar el algoritmo MBSO con el operador de frontera BS se presentan en la Tabla 5.3. Los resultados muestran que el uso del operador de frontera BS en el algoritmo MBSO permite mejorar los valores en la función objetivo tanto con 10D como en 30D, los cuales son resaltados en negritas en la misma tabla.

La Tabla 5.3 compara los resultados de la propuesta MBSO-BS con los obtenidos por el algoritmo MBSO sin el operador de frontera. Como se puede observar en 10D se pudo mejorar el valor obtenido por el algoritmo MBSO en 12 funciones (C01-C05, C08, C12, C14-C18). Además, aunque estadísticamente (ver columna ST de la misma tabla) sólo la función C16 muestra una mejora significativa de acuerdo a la prueba de suma de rango de Wilcoxon, la version del algoritmo con el operador especial es capaz de obtener soluciones factibles en todo el conjunto de funciones de prueba con 10D. Este comportamiento indica que colocar puntos clave en la frontera de las restricciones esta permitiendo al algoritmo MBSO concentrar la búsqueda en la región de interés, es decir, en la región factible de cada función de prueba. Un comportamiento similar se presentó en 30D mejorando el valor de 13 funciones en comparación con el algoritmo MBSO. Las funciones mejoradas son C01-C07, C10, C12, C14, C15, C17, C18.

Tabla 5.3: Experimento 1: Resultados obtenidos por la combinación del Algoritmo MBSO con el operador BS, 10D y 30D

F	CRITERIO	D10			D30		
		MBSO	MBSO-BS	ST	MBSO	MBSO-BS	ST
C01	MEJOR	-7.473103E-01	-7.473103E-01	=	-8.199336E-01	**-8.218535E-01**	=
	MEDIA	-7.426697E-01	**-7.453253E-01**		-7.940688E-01	**-8.002866E-01**	
	STD	1.061624E-02	**5.884700E-03**		1.839617E-02	**1.604693E-02**	
C02	MEJOR	-1.024535E+00	**-2.277705E+00**	=	-1.931765E+00	**-2.261698E+00**	+
	MEDIA	-1.024535E+00	**-2.126006E+00**		8.023309E-03	**-1.884621E+00**	
	STD	**0.000000E+00**	4.276689E-01		2.000646E+00	**4.181442E-01**	
C03	MEJOR	0.000000E+00	0.000000E+00	=	2.867347E+01	2.867347E+01	=
	MEDIA	1.966720E+06	**3.110000E-21**		5.300617E+01	**3.590278E+01**	
	STD	9.634921E+06	**1.210000E-20**		4.452446E+01	**2.883039E+01**	
C04	MEJOR		-1.000000E-05				
	MEDIA		3.885281E-03				
	STD		6.036209E-03				
C05	MEJOR		-4.834950E+02			-1.840970E+02	
	MEDIA		-1.793699E+02			3.366424E+02	
	STD		4.300943E+02			2.567318E+02	
C06	MEJOR	**-5.739239E+02**	-5.737048E+02	=	-1.698576E+02	**-2.475472E+02**	=
	MEDIA	**-3.806216E+02**	-3.761981E+02		3.345201E+02	**2.408127E+02**	
	STD	**3.487267E+02**	3.562107E+02		**3.001146E+02**	3.172443E+02	
C07	MEJOR	0.000000E+00	0.000000E+00	=	**9.900265E-10**	1.610000E-09	=
	MEDIA	**8.370000E-20**	1.450000E-05		1.594651E-01	**1.100000E-06**	
	STD	**3.540000E-19**	7.090000E-05		7.812155E-01	**3.370000E-06**	
C08	MEJOR	3.820000E-28	**0.000000E+00**	=	**2.993716E-12**	6.930000E-11	-
	MEDIA	8.797871E+00	**7.402312E+00**		**1.160860E+01**	8.119647E+01	
	STD	**4.492596E+00**	6.236581E+00		**5.687029E+01**	1.713378E+02	
C09	MEJOR	1.952055E+00	**0.000000E+00**	=	2.820875E-07	**9.760000E-11**	=
	MEDIA	**3.433254E+01**	4.880000E+10		**1.193549E+02**	3.076190E+02	
	STD	**2.876523E+01**	2.070000E+11		**9.847024E+01**	9.378332E+02	
C10	MEJOR	8.553841E-02	**2.590000E-24**	=	**2.294922E+01**	3.131214E+01	+
	MEDIA	**3.162346E+02**	1.270000E+11		1.704831E+12	**9.720000E+11**	
	STD	**3.387936E+02**	5.100000E+11		7.620273E+12	**4.350000E+12**	
C11	MEJOR	-1.522713E-03	-1.522713E-03	=	**-3.922282E-04**	-3.920000E-04	+
	MEDIA	-1.522711E-03	-1.522711E-03		-3.137408E-04	**-3.380000E-04**	
	STD	1.560000E-09	**1.280000E-09**		**6.331270E-05**	1.080000E-04	
C12	MEJOR	-1.992458E-01	**-5.700852E+02**	=	-1.992618E-01	**-1.992632E-01**	=
	MEDIA	-4.506579E-02	**-1.687469E+02**		-1.080196E-01	**-1.938905E-01**	
	STD	**7.553234E-01**	2.575207E+02		4.469290E-01	**2.269356E-02**	
C13	MEJOR	-6.842936E+01	-6.842936E+01	-	**-6.562204E+01**	-6.490825E+01	-
	MEDIA	**-6.738378E+01**	-6.379652E+01		**-6.344104E+01**	-5.991580E+01	
	STD	**1.761360E+00**	2.471246E+00		**1.439881E+00**	2.606253E+00	
C14	MEJOR	5.570000E-27	**0.000000E+00**	=	7.071555E-09	**5.010000E-12**	=
	MEDIA	5.100000E+08	**1.594632E-01**		1.074479E+01	**6.554351E+00**	
	STD	2.420000E+09	**7.812068E-01**		5.260264E+01	**2.266280E+01**	
C15	MEJOR	**0.000000E+00**	3.160000E-27	=	4.215002E+00	**1.470000E-11**	=
	MEDIA	1.060000E+10	**3.059192E+06**		3.331180E+12	**6.080000E+09**	
	STD	4.740000E+10	**1.500000E+07**		1.080224E+13	**2.980000E+10**	
C16	MEJOR	0.000000E+00	0.000000E+00	+	0.000000E+00	0.000000E+00	=
	MEDIA	5.994882E-01	**1.006303E-01**		**1.200782E-03**	1.015864E-02	
	STD	4.454175E-01	**1.968174E-01**		**4.083554E-03**	4.452883E-02	
C17	MEJOR	0.000000E+00	0.000000E+00	=	1.451447E-02	**1.279146E-03**	+
	MEDIA	7.366529E+00	**2.376878E-01**		2.135308E+02	**2.464582E+00**	
	STD	1.714114E+01	**5.408658E-01**		4.122819E+02	**8.412085E+00**	
C18	MEJOR	0.000000E+00	0.000000E+00	=	**1.526043E-04**	6.291048E-02	=
	MEDIA	5.830000E-06	**0.000000E+00**		5.756742E+03	**4.795064E+02**	
	STD	2.330000E-05	**0.000000E+00**		8.694744E+03	**1.911847E+03**	

5.5.2. Experimento 2: Desempeño del operador de frontera BS-r

Los resultados de este experimento se muestran en las Tablas 5.4 y 5.5 donde se observa que valores de r mayores a 0.6 hacen que en funciones como C04, C05, C06, C09 y C10 sea difícil obtener soluciones factibles y mayormente en 30D. Este desempeño muestra que valores altos en r requieren un mayor número de evaluaciones para alcanzar la frontera factible dejando pocas evaluaciones al algoritmo MBSO e impidiendo a éste explorar efectivamente el espacio de búsqueda y encontrar así la región factible. Este comportamiento se refleja en la Figura 5.3.

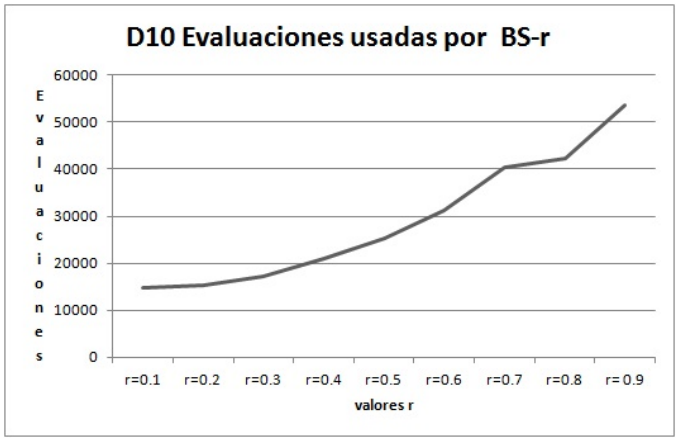

Figura 5.3: Experimento 2. Evaluaciones gastadas por el operador BS basado en segmento de línea variando el valor de r.

Tabla 5.4: Experimento 2: Resultados obtenidos por la combinación de MBSO y BS-r, con 10D

F	CRITERIO	r = 0.1	r = 0.2	r = 0.3	r = 0.4	r = 0.5	r = 0.6	r = 0.7	r = 0.8	r = 0.9
C01	MEJOR	-7.473102E-01	-7.473101E-01	-7.473102E-01	-7.473101E-01	-7.473102E-01	**-7.473103E-01**	-7.473101E-01	**-7.473103E-01**	-7.473102E-01
	MEDIA	**-7.461825E-01**	-7.437206E-01	-7.449399E-01	-7.434585E-01	-7.441979E-01	-7.436990E-01	-7.453251E-01	-7.461285E-01	-7.455954E-01
	STD	4.350295E-03	9.265315E-03	5.865574E-03	1.052856E-02	7.005693E-03	9.089079E-03	5.884685E-03	**2.741792E-03**	5.814181E-03
C02	MEJOR	**-2.277710E+00**	-2.277707E+00	-2.277709E+00	-2.277709E+00	-2.277708E+00	-2.277705E+00	-2.277705E+00	-2.277671E+00	-1.605581E-02
	MEDIA	-1.976785E+00	-2.087057E+00	-2.036947E+00	-2.227726E+00	-2.159588E+00	-1.996206E+00	-2.104111E+00	**-2.275630E+00**	-1.605581E-02
	STD	1.215701E+00	7.291500E-01	6.579564E-01	1.910640E-01	3.243558E-01	9.424912E-01	8.175919E-01	3.917655E-03	**0.000000E+00**
C03	MEJOR	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	MEDIA	**1.325175E-22**	1.627796E-21	3.259289E-21	2.794007E-18	1.908267E-22	1.630255E-19	4.923553E-22	3.550221E-01	3.550221E-01
	STD	**4.093993E-22**	6.280643E-21	1.276625E-20	1.362473E-17	7.689954E-22	7.983419E-19	1.302546E-21	1.739246E+00	1.739246E+00
C04	MEJOR	-9.999907E-06	**-1.000000E-05**	-9.99819SE-06	**-1.000000E-05**	-9.999999E-06	**-1.000000E-05**	-9.999547E-06	-9.999567E-06	-9.999999E-06
	MEDIA	1.379302E-03	1.009721E-03	1.102776E-03	2.814813E-02	1.000884E-03	1.284483E-03	2.852787E-02	**9.345202E-04**	1.393296E-03
	STD	1.035167E-03	1.155631E-03	**1.018408E-03**	1.331439E-01	1.259936E-03	1.235459E-03	1.327623E-01	1.044300E-03	1.251023E-03
C05	MEJOR	-4.836106E+02	-4.836106E+02	-4.836106E+02	-4.836106E+02	-4.836106E+02	-4.836106E+02	-4.836106E+02		
	MEDIA	-3.605830E+02	-3.921232E+02	**-4.139718E+02**	-3.956923E+02	-3.497517E+02	-3.588944E+02	-2.996809E+02		
	STD	1.815733E+02	1.346860E+02	1.320853E+02	**1.181879E+02**	1.586855E+02	1.737027E+02	2.500045E+02		
C06	MEJOR	-5.785718E+02	-5.785715E+02	-5.784922E+02	**-5.786124E+02**	-5.785726E+02	-5.786009E+02	-5.782661E+02		
	MEDIA	-4.936068E+02	-4.468503E+02	-5.531547E+02	-5.249821E+02	-5.421639E+02	-4.114484E+02	**-5.782661E+02**		
	STD	1.705314E+02	2.303468E+02	1.124846E+02	1.602301E+02	8.080462E+01	2.753887E+02	**0.000000E+00**		
C07	MEJOR	1.014317E-26	**0.000000E+00**	**0.000000E+00**	2.186722E-27	**0.000000E+00**	4.743026E-28	1.081218E-26	2.597127E-27	**0.000000E+00**
	MEDIA	7.435052E-19	2.199157E-11	1.725137E-19	4.372625E-18	**1.246266E-21**	4.175272E-19	1.785140E-18	7.981354E-18	6.195109E-20
	STD	2.381378E-18	1.077363E-10	7.859764E-19	1.552282E-17	**2.343687E-21**	1.973771E-18	7.369673E-18	2.650127E-17	2.105355E-19
C08	MEJOR	2.592246E-27	**0.000000E+00**	**0.000000E+00**	1.095536E-26	**0.000000E+00**	1.091251E-26	6.704718E-24	2.186722E-27	2.335028E-27
	MEDIA	1.298013E+01	6.223856E+00	7.573365E+00	6.780629E+00	7.904660E+00	**5.539270E+00**	8.620521E+00	8.401511E+00	7.247786E+00
	STD	2.080316E+01	6.199489E+00	5.269496E+00	4.990404E+00	4.636086E+00	5.160004E+00	**4.313381E+00**	4.433504E+00	4.875851E+00
C09	MEJOR	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	8.073533E-21	
	MEDIA	8.950714E+09	8.725552E+10	3.767878E+10	3.776241E+00	1.396600E+10	5.575103E+10	3.526522E-01	**3.410398E-01**	
	STD	4.125031E+10	3.980573E+11	1.833031E+11	1.631726E+01	6.664184E+10	2.730396E+11	1.195902E+00	**1.067855E+00**	4.875851E+00
C10	MEJOR	1.910937E-26	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	6.205403E-18	
	MEDIA	6.614592E+09	5.648985E+07	1.314171E+09	6.093800E+06	1.131545E+05	1.543070E+11	9.858769E+09	**8.902164E-01**	
	STD	3.048772E+10	2.767425E+08	5.569632E+09	2.985337E+07	5.347293E+05	7.559471E+11	3.431521E+10	**2.715322E+00**	
C11	MEJOR	-1.522713E-03	-1.522713E-03	-1.522713E-03	-1.522713E-03	-1.522713E-03	-1.522713E-03	-1.522713E-03	-1.522713E-03	-1.522713E-03
	MEDIA	-1.522712E-03	-1.522712E-03	-1.522712E-03	-1.522712E-03	-1.522712E-03	-1.522712E-03	-1.522712E-03	-1.522712E-03	-1.522712E-03
	STD	4.650307E-10	4.010151E-10	4.585497E-10	4.898977E-10	5.641840E-10	4.474467E-10	4.758918E-10	**3.586478E-10**	5.925952E-10
C12	MEJOR	-1.992457E-01	**-5.700899E+02**	-5.683509E+02	-1.992458E-01	-1.700346E+02	-2.218061E+02	-6.626281E+01	-5.609788E+02	-2.591629E+02
	MEDIA	-1.992456E-01	-2.299487E+01	-2.354132E+01	-1.978943E-01	-7.491374E+00	-9.308133E+00	1.574434E-01	**-2.952719E+01**	-1.041974E+01
	STD	**1.821135E-07**	1.116753E+02	1.112317E+02	6.620429E-03	3.326867E+01	4.431292E+01	2.017393E+01	1.136132E+02	5.193830E+01
C13	MEJOR	**-6.842936E+01**	**-6.842936E+01**	-6.842935E+01	**-6.842936E+01**	**-6.842936E+01**	-6.557847E+01	**-6.842936E+01**	-6.842935E+01	-6.842935E+01
	MEDIA	-6.358897E+01	-6.490225E+01	**-6.495900E+01**	-6.308126E+01	-6.447615E+01	-6.363481E+01	-6.348541E+01	-6.428896E+01	-6.408322E+01
	STD	2.642441E+00	2.320169E+00	2.267258E+00	3.063017E+00	2.517152E+00	**6.209168E-01**	2.226923E+00	1.778895E+00	1.699689E+00
C14	MEJOR	1.643986E-27	1.113294E-24	**0.000000E+00**	2.524670E-26	4.887782E-27	1.902417E-26	3.314794E-26	5.257461E-24	7.419540E-24
	MEDIA	2.186858E-12	1.594632E-01	3.560323E-16	1.980776E-14	**1.352319E-17**	1.905412E-16	3.189263E-01	2.433605E-15	1.073975E-16
	STD	1.035413E-11	7.812068E-01	1.644367E-15	9.692193E-14	**5.008146E-17**	5.910115E-16	1.081532E+00	1.151568E-14	4.855983E-16
C15	MEJOR	5.049521E-24	6.713478E-23	**3.728472E-26**	1.292880E-25	1.493566E-25	1.336614E-23	4.007413E-26	2.645303E-25	3.932314E-26
	MEDIA	6.490145E+10	8.305605E+09	**4.407887E-01**	1.100974E+11	1.871439E+09	3.321823E+10	1.731116E+09	6.510644E+09	4.827704E+11
	STD	3.078170E+11	3.959968E+10	**1.193662E+00**	5.393650E+11	9.168139E+09	1.595319E+11	8.480703E+09	1.791191E+10	2.363926E+12
C16	MEJOR	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	MEDIA	**4.541621E-02**	5.821864E-02	9.453968E-02	6.086684E-02	8.488502E-02	4.823963E-02	8.399412E-02	5.421474E-02	1.059572E-01
	STD	**6.447196E-02**	1.723607E-01	3.286865E-01	1.847363E-01	1.989817E-01	1.471178E-01	1.758703E-01	1.111290E-01	2.083254E-01
C17	MEJOR	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	1.007647E-29
	MEDIA	2.447278E+00	1.594486E-01	7.561638E+00	7.095383E+00	2.695766E+00	1.569484E-01	**7.863986E-02**	2.628179E+00	1.817824E-01
	STD	1.176571E+01	4.196204E-01	3.678638E+01	3.358761E+01	1.152329E+01	5.006872E-01	**1.931900E-01**	7.818167E+00	3.657755E-01
C18	MEJOR	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	3.375268E-23
	MEDIA	9.508763E-02	2.934680E-04	2.234675E+01	1.942042E+01	7.259870E+01	**2.107572E-25**	6.205433E-22	6.755889E+01	1.454929E+02
	STD	3.581179E-01	1.437694E-03	5.752416E+01	7.775645E+01	3.532991E+02	**1.032495E-24**	3.001765E-21	2.951496E+02	3.634213E+02

Tabla 5.5: Experimento 2: Resultados obtenidos por la combinación de MBSO y BS-r, con 30D

F	CRITERIO	r = 0.1	r = 0.2	r = 0.3	r = 0.4	r = 0.5	r = 0.6	r = 0.7	r = 0.8	r = 0.9
C01	MEJOR	-8.199331E-01	-8.180550E-01	-8.179462E-01	-8.218810E-01	-8.218813E-01	**-8.218831E-01**	-8.180536E-01	-8.21882 4E-01	-8.180549E-01
	MEDIA	-7.979353E-01	-7.874046E-01	-7.923190E-01	-7.938682E-01	-7.952162E-01	-7.927842E-01	-7.923642E-01	-7.960788E-01	**-7.980570E-01**
	STD	1.424188E-02	2.262632E-02	**1.349856E-02**	1.769703E-02	1.664078E-02	2.183830E-02	1.970792E-02	1.611510E-02	1.834485E-02
C02	MEJOR	-2.277263E+00	-2.276618E+00	-2.277442E+00	-2.277254E+00	-2.277251E+00	**-2.277585E+00**	-2.277392E+00	-2.277141E+00	-1.856680E+00
	MEDIA	-2.033479E+00	-1.984510E+00	-1.959209E+00	-1.957789E+00	-1.941693E+00	-2.113026E+00	**-2.144458E+00**	-2.010840E+00	-1.856680E+00
	STD	4.694480E-01	7.396981E-01	1.050914E+00	8.100867E-01	6.314631E-01	2.964855E-01	2.591067E-01	7.609689E-01	**0.000000E+00**
C03	MEJOR	**4.473669E-10**	2.367634E-09	5.536683E-01	2.867347E+01	2.867347E+01	3.716663E-07	2.867347E+01	1.094542E+01	5.628960E-09
	MEDIA	6.891859E+01	6.195357E+01	5.860637E+01	6.304622E+01	**5.636026E+01**	5.581981E+01	6.780548E+01	6.866013E+01	8.321868E+01
	STD	4.250408E+01	4.945514E+01	**2.891682E+01**	3.801309E+01	3.177682E+01	3.340765E+01	3.647214E+01	3.712498E+01	5.115658E+01
C04	MEJOR	-	-	-	2.039430E+01	-	-	-	-	**1.937036E-01**
	MEDIA				2.039430E+01					**1.937036E-01**
	STD				0.000000E+00					0.000000E+00
C05	MEJOR	-3.442846E+02	-3.251624E+02	-2.996327E+02	**-4.568796E+02**	-3.664991E+02	-3.066047E+02	3.119650E+02	-	-
	MEDIA	-2.002843E+02	-1.434077E+02	-1.818123E+02	**-2.502327E+02**	-2.131872E+02	-1.846591E+02	3.119650E+02		
	STD	8.885472E+01	1.772759E+02	1.137531E+02	1.084181E+02	8.288013E+01	1.330804E+02	**0.000000E+00**		
C06	MEJOR	-4.789778E+02	**-5.204431E+02**	-4.963454E+02	-5.112752E+02	-5.148038E+02	-5.099820E+02	2.682484E+02	-	-
	MEDIA	-2.156894E+02	**-3.116364E+02**	-2.574597E+02	-2.794808E+02	-2.236562E+02	-2.168404E+02	2.682484E+02		
	STD	1.599803E+02	1.401805E+02	2.220076E+02	1.921440E+02	2.262707E+02	1.866025E+02	**0.000000E+00**		
C07	MEJOR	6.581910E-10	1.586422E-11	5.274880E-11	1.599840E-10	2.516338E-09	2.127434E-10	1.042417E-11	**2.210947E-12**	1.415619E-10
	MEDIA	6.870430E-07	1.594651E-01	4.783950E-01	**5.205032E-07**	3.189647E-01	1.594652E-01	1.594661E-01	1.594652E-01	1.595346E-01
	STD	2.336111E-06	7.812155E-01	1.295499E+00	**1.166507E-06**	1.081534E+00	7.812157E-01	7.812153E-01	7.812155E-01	7.812014E-01
C08	MEJOR	1.912923E-08	2.934436E-09	**6.394597E-11**	5.579369E-10	8.259603E-10	9.494869E-09	1.008300E-09	3.680358E-08	5.576000E-10
	MEDIA	6.955829E+01	1.331174E+02	**2.132241E+01**	1.064030E+02	3.730418E+01	6.937903E+01	3.771022E+01	2.792996E+01	3.397398E+00
	STD	1.885293E+02	2.706616E+02	6.654177E+01	2.565145E+02	1.165107E+02	1.391876E+02	8.288941E+01	6.941408E+01	**1.664378E+01**
C09	MEJOR	4.141292E-16	8.162985E-16	**4.084236E-16**	6.769814E-15	8.179961E-14	5.714594E-13	1.916192E-13	6.920923E-11	-
	MEDIA	8.485500E+11	4.870462E+11	4.748640E+08	1.800728E+11	1.090090E+07	2.670258E+10	4.446901E+01	**3.580109E+01**	
	STD	2.873664E+12	2.328155E+12	2.326349E+09	8.723326E+11	5.340296E+07	1.264614E+11	**3.434831E+01**	4.938160E+01	
C10	MEJOR	3.174958E+01	**6.546980E+00**	3.731234E-10	6.571556E+00	1.271753E+01	1.703865E+01	6.547042E+00	4.381712E+01	-
	MEDIA	3.951907E+10	1.858856E+06	6.498226E+11	1.514851E+10	4.227503E+10	1.757220E+11	5.027711E+02	**2.992416E+02**	
	STD	1.936031E+11	9.105717E+06	2.266826E+12	6.771499E+10	1.565687E+11	7.970673E+11	2.258005E+03	**5.815667E+02**	
C11	MEJOR	-3.922710E-04	-3.923378E-04	-3.923137E-04	-3.922914E-04	**-3.923379E-04**	-3.922935E-04	-3.901353E-04	-3.922365E-04	-3.922874E-04
	MEDIA	-1.000646E-04	-1.907404E-04	-1.245599E-04	**-1.942464E-04**	-1.532695E-04	-9.126480E-05	-1.865165E-04	-1.415448E-04	-1.655191E-04
	STD	1.787449E-04	1.698832E-04	2.184984E-04	1.490302E-04	1.527757E-04	1.979945E-04	1.424683E-04	**1.415244E-04**	1.985155E-04
C12	MEJOR	-1.992632E-01	**-1.992634E-01**	-1.992633E-01	-1.992629E-01	**-1.992634E-01**	-1.992630E-01	**-1.992634E-01**	**-1.992634E-01**	**-1.992634E-01**
	MEDIA	9.428682E-01	-5.310139E-02	-1.965180E-01	2.558735E-01	**-1.992534E-01**	-1.991323E-01	-1.303664E-01	-8.042686E-02	-1.843827E-01
	STD	2.996101E+00	6.365173E-01	1.282674E-02	1.374369E+00	**7.403974E-06**	5.170184E-04	2.609009E-01	3.071844E-01	6.811633E-02
C13	MEJOR	-6.533405E+01	-6.533403E+01	-6.490834E+01	-6.364725E+01	-6.364722E+01	-6.432911E+01	**-6.561393E+01**	-6.364718E+01	-6.351741E+01
	MEDIA	-6.084882E+01	-6.133186E+01	-6.136846E+01	**-6.141815E+01**	-6.058031E+01	-6.105432E+01	-6.115205E+01	-6.123988E+01	-6.110351E+01
	STD	2.407293E+00	2.203691E+00	2.134950E+00	**1.395707E+00**	1.521414E+00	2.049446E+00	2.351658E+00	1.521373E+00	1.653716E+00
C14	MEJOR	1.454592E-09	2.979495E-09	5.299306E-10	8.231785E-11	1.101931E-10	**4.922296E-12**	1.466123E-09	1.168683E-11	1.525753E-07
	MEDIA	1.698073E+01	4.732631E+01	1.139208E+01	3.814118E+01	1.107291E+01	**8.967824E-07**	1.613004E+01	6.378626E+01	7.219705E+00
	STD	8.156727E+01	2.310382E+02	5.176379E+01	1.298066E+02	5.182074E+01	**2.232496E-06**	7.740073E+01	1.461520E+00	3.536917E+01
C15	MEJOR	2.024216E-09	5.967321E-10	8.009663E-11	**7.108706E-12**	1.373892E-09	5.167992E-10	3.849164E-10	6.135859E-10	4.042401E-09
	MEDIA	2.746736E+11	1.961806E+10	2.682906E+09	1.278536E+09	4.930577E+10	**2.057560E+01**	2.433714E+11	3.095748E+09	8.912060E+09
	STD	1.345620E+12	9.610425E+10	1.314350E+10	6.263521E+09	2.137990E+11	**2.632386E+01**	1.190850E+12	1.516601E+10	4.363276E+10
C16	MEJOR	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**
	MEDIA	7.286744E-04	2.600713E-03	9.745663E-03	5.135254E-04	1.024820E-02	4.105636E-02	**0.000000E+00**	7.047891E-04	2.746608E-03
	STD	3.569761E-03	7.427066E-03	4.367428E-02	2.515750E-03	3.301768E-02	1.946822E-01	**0.000000E+00**	3.452747E-03	7.471656E-03
C17	MEJOR	**9.006056E-16**	1.530907E-12	1.586286E-06	9.264679E-06	5.869639E-10	3.834716E-05	2.068439E-05	1.870053E-05	8.863433E-04
	MEDIA	**1.384125E+01**	2.783121E+01	4.278712E+01	2.883387E+01	4.596532E+01	5.260474E+01	4.371563E+01	1.140742E+02	4.103979E+01
	STD	**3.232634E+01**	1.324329E+02	1.720471E+02	1.084730E+02	1.060292E+02	1.393234E+02	1.794907E+02	4.691220E+02	7.695462E+01
C18	MEJOR	5.451333E-06	9.269846E-07	4.313439E-07	1.089667E-07	**1.874629E-09**	2.545633E-07	3.794157E-08	4.289967E-04	1.391887E-01
	MEDIA	1.608164E+02	6.605174E+01	7.091117E+00	**3.811537E+00**	2.104035E+02	1.187339E+01	6.977921E+01	3.043783E+02	4.303926E+02
	STD	5.431152E+02	2.703996E+02	2.170743E+01	**7.299359E+00**	8.863678E+02	4.464419E+01	3.317068E+02	1.383409E+03	1.471795E+03

Valores muy similares fueron obtenidos en las variantes del operador BS-r (ver Tablas 5.4 y 5.5), esta situación implica que no hay preferencia por algún valor del parámetro r como se muestra en los resultados de la prueba estadística post-hoc de Bonferroni en las Figuras 5.5 y 5.6. Lo que sí nos permite es reducir el rango de posibles valores que este parámetro pudiera tomar, ahora en lugar de tomar valores entre 0.1 y 0.9 este rango puede reducirse a 0.1 y 0.7 que son valores donde el operador obtiene soluciones factibles en todo el conjunto de prueba. Además, aunque el número de evaluaciones usado por la versión BS-r comparada con la versión original del operador BS no decrementó, como se muestra en la Figura 5.4, mejores valores en la función objetivo fueron reportados usando la version BS-r en las funciones C02, C05, C06 y C10 con 10D y en la mitad de las funciones con 30D (C02, C03, C05 - C10, C17 y C18).

De las nueve versiones presentadas en las Tablas 5.4 y 5.5 haciendo un conteo numérico la versión con valor de $r = 0.3$ con 10D y 0.4 con 30D son las que mejores valores reportan en la función objetivo, es por ello que se toman estas versiones para compararse con los resultados obtenidos por los operadores BS y BS-G.

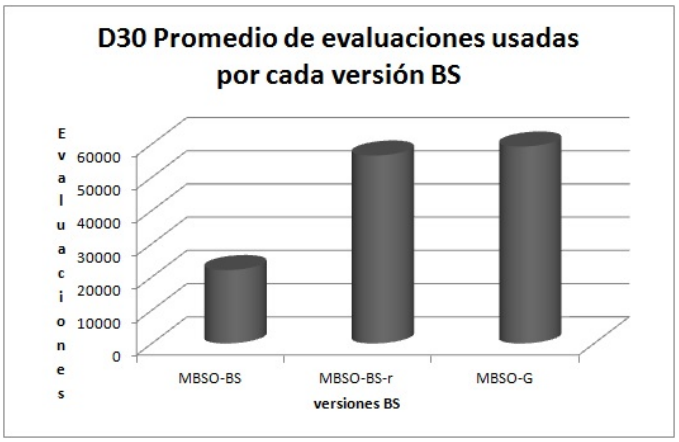

Figura 5.4: Evaluaciones utilizadas en las variantes propuestas 30D.

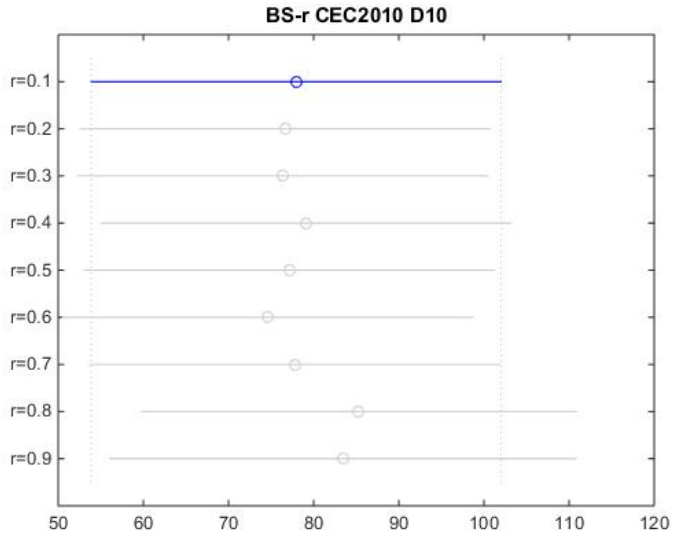

Figura 5.5: Prueba estadística post-hoc de Bonferroni para los resultados del operador BS-r D10

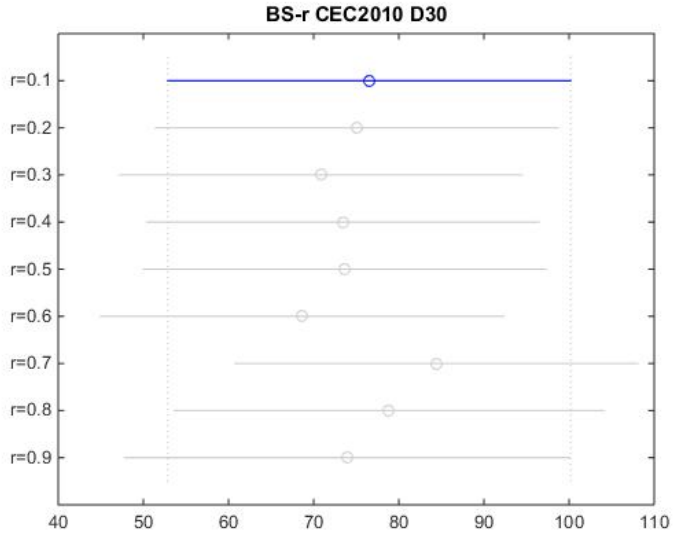

Figura 5.6: Prueba estadística post-hoc de Bonferroni para los resultados del operador BS-r D30

5.5.3. Experimento 3. Desempeño del operador BS-G

La condición necesaria para aplicar los operadores de frontera BS y BS-r es contar con puntos factibles y no factibles como puntos de entrada, si el algoritmo tiene problemas para encontrar soluciones factibles en alguna función como en la C04 con 30D, esta condición no se cumpliría y por lo tanto los operadores nunca serían aplicados. Como una solución a este problema, el método basado en gradiente, el cual no necesita de puntos factibles es ejecutado en este experimento obteniendo los resultados reportados en la Tabla 5.6.

Como se puede observar en la Tabla 5.6, agregando el operador de gradiente al algoritmo MBSO se lograron mejorar los resultados de las funciones tanto con 10D como en 30D. En 10D se mejoraron los resultados de siete funciones (C02, C04, C05, C06, c09, c10 y C14. Por otro lado, con 30D, además de encontrar soluciones factibles en todas las funciones de prueba, se mejoraron los resultados en once de las dieciocho funciones (C02-C06, C09-C11, C14, C15 y C18). Estadísticamente (ver columna ST en la misma tabla) la versión del algoritmo que utiliza el operador especial basado en el gradiente de la restricción más difícil de satisfacer muestra una mejora significativa a favor de 5 funciones (C03, C06, C10, C11, y C16) en D10 y D30 (C02, C03, C06, C15 y C18).

Aunque los resultados en la función objetivo fueron mejorados utilizando el operador de frontera basado en gradiente no necesariamente se redujo el número de evaluaciones como se esperaba. En la Figura 5.4 puede observarse un comparativo de las evaluaciones gastadas por cada operador de frontera visto en este capítulo.

Tabla 5.6: Resultados obtenidos por la combinación del Algoritmo MBSO y el método del gradiente como operador de frontera, 10D y 30D.

F	CRITERIO	D10			D30		
		MBSO	MBSO-BS-CG	ST	MBSO	MBSO-BS OR G	ST
C01	MEJOR	**-7.473103E-01**	-7.473102E-01	=	**-8.199336E-01**	-8.180541E-01	=
	MEDIA	-7.426697E-01	**-7.461825E-01**		**-7.940688E-01**	-7.934879E-01	
	STD	1.061624E-02	**4.350296E-03**		**1.839617E-02**	1.856451E-02	
C02	MEJOR	-1.024535E+00	**-2.277707E+00**	=	-1.931765E+00	**-2.256826E+00**	+
	MEDIA	-1.024535E+00	**-2.243236E+00**		8.023309E-03	**-1.685913E+00**	
	STD	**0.000000E+00**	4.043046E-02		2.000646E+00	**6.670892E-01**	
C03	MEJOR	0.000000E+00	0.000000E+00	+	2.867347E+01	**2.390000E-19**	+
	MEDIA	1.966720E+06	**6.106018E-21**		5.300617E+01	**1.605714E+01**	
	STD	9.634921E+06	**2.926590E-20**		4.452446E+01	**1.423313E+01**	
C04	MEJOR		-9.999998E-06			1.590548E-03	
	MEDIA		2.835219E-02			2.354341E-02	
	STD		1.327263E-01			2.771898E-02	
C05	MEJOR		-4.836106E+02			-4.629752E+02	
	MEDIA		-3.674250E+02			-2.003758E+02	
	STD		1.295727E+02			1.358803E+02	
C06	MEJOR	-5.739239E+02	**-5.785518E+02**	+	-1.698576E+02	**-5.287628E+02**	+
	MEDIA	-3.806216E+02	**-4.909016E+02**		3.345201E+02	**-2.746376E+02**	
	STD	3.487267E+02	**1.382472E+02**		3.001146E+02	**1.639472E+02**	
C07	MEJOR	0.000000E+00	0.000000E+00	=	**9.900265E-10**	4.470000E-09	=
	MEDIA	8.370000E-20	**5.994037E-20**		**1.594651E-01**	3.189305E-01	
	STD	3.540000E-19	**2.849815E-19**		**7.812155E-01**	1.081544E+00	
C08	MEJOR	**3.820000E-28**	2.335028E-27	=	**2.993716E-12**	4.990000E-10	-
	MEDIA	8.797871E+00	**5.949551E+00**		**1.160860E+01**	7.941454E+01	
	STD	**4.492596E+00**	5.182512E+00		**5.687029E+01**	2.161634E+02	
C09	MEJOR	1.952055E+00	**0.000000E+00**	-	2.820875E-07	**4.100000E-15**	-
	MEDIA	**3.433254E+01**	1.720570E+09		**1.193549E+02**	6.460000E+11	
	STD	**2.876523E+01**	7.584969E+09		**9.847024E+01**	2.150000E+12	
C10	MEJOR	8.553841E-02	**0.000000E+00**	+	2.294922E+01	**1.271753E+01**	=
	MEDIA	3.162346E+02	**5.007209E+00**		1.704831E+12	**2.360000E+11**	
	STD	3.387936E+02	**1.355959E+01**		7.620273E+12	**1.120000E+12**	
C11	MEJOR	-1.522713E-03	-1.522713E-03	+	**-3.922282E-04**	-3.920000E-04	=
	MEDIA	-1.522711E-03	-1.522712E-03		**-3.137408E-04**	-2.580000E-04	
	STD	1.560000E-09	**4.909212E-10**		**6.331270E-05**	2.270000E-04	
C12	MEJOR	-1.992458E-01	-1.992458E-01	=	**-1.992618E-01**	-1.992441E-01	=
	MEDIA	**-4.506579E-02**	2.724112E+01		-1.080196E-01	**-1.764895E-01**	
	STD	**7.553234E-01**	9.415815E+01		4.469290E-01	**2.275455E-02**	
C13	MEJOR	-6.842936E+01	-6.842936E+01	-	**-6.562204E+01**	-6.533404E+01	-
	MEDIA	**-6.738378E+01**	-6.427393E+01		**-6.344104E+01**	-5.953384E+01	
	STD	**1.761360E+00**	2.864785E+00		**1.439881E+00**	2.761740E+00	
C14	MEJOR	5.570000E-27	**0.000000E+00**	=	7.071555E-09	**1.630000E-09**	=
	MEDIA	5.100000E+08	**2.102895E+01**		**1.074479E+01**	1.207542E+03	
	STD	2.420000E+09	**1.030204E+02**		**5.260264E+01**	5.843198E+03	
C15	MEJOR	**0.000000E+00**	1.593491E-22	=	4.215002E+00	**2.450000E-10**	+
	MEDIA	1.060000E+10	**1.727940E+08**		3.331180E+12	**3.680995E+01**	
	STD	4.740000E+10	**6.100501E+08**		1.080224E+13	**8.581481E+01**	
C16	MEJOR	0.000000E+00	0.000000E+00	+	0.000000E+00	0.000000E+00	=
	MEDIA	5.994882E-01	**1.399923E-01**		**1.200782E-03**	8.758593E-03	
	STD	4.454175E-01	**1.988350E-01**		**4.083554E-03**	3.016258E-02	
C17	MEJOR	0.000000E+00	0.000000E+00	=	1.451447E-02	8.947486E-02	=
	MEDIA	**7.366529E+00**	7.701546E+00		2.135308E+02	**2.042799E+01**	
	STD	**1.714114E+01**	3.717040E+01		4.122819E+02	**3.409048E+01**	
C18	MEJOR	0.000000E+00	0.000000E+00	=	1.526043E-04	**2.450000E-09**	+
	MEDIA	**5.830000E-06**	1.435094E+01		5.756742E+03	**2.575481E+02**	
	STD	**2.330000E-05**	7.030496E+01		8.694744E+03	**1.102910E+03**	

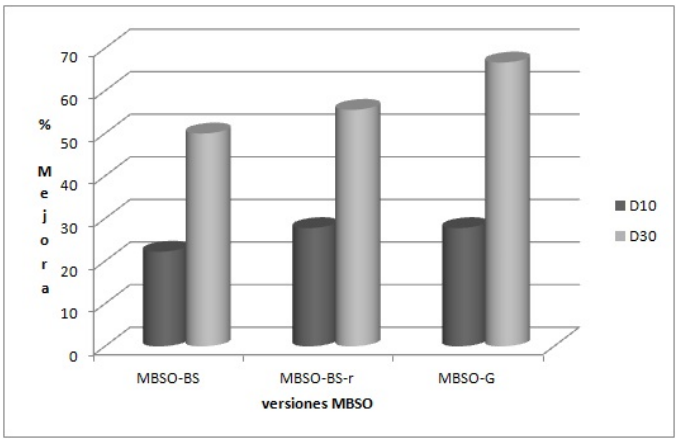

Figura 5.7: Mejor versión del operador de frontera y su porcentaje de mejora.

5.6. MBSO-BS comparado contra al algoritmo εDEag

En este apartado se muestra un comparativo entre los resultados obtenidos por el algoritmo MBSO sin operador de frontera y los resultados obtenidos por el mismo algoritmo combinado con los operadores de frontera vistos en este capítulo. El algoritmo εDEag es utilizado como base para comparar los resultados obtenidos. Los resultados muestran al mejor valor obtenido en la función objetivo, el promedio y la desviación estándar.

Tabla 5.7: Resultados obtenidos por la combinación del Algoritmo MBSO y las tres versiones del operador de frontera BS con 10D y 30D comparados con el algoritmo εDEag.

F	Criterio	D10					D30				
		εDEag	MBSO	MBSO-BS	MBSO-BS-r=0.3	MBSO-BS-CG	εDEag	MBSO	MBSO-BS	MBSO-BS-r=0.4	MBSO-BS OR G
C01	MEJOR	**-7.473104E-01**	-7.473103E-01	-7.473103E-01	-7.473102E-01	-7.473102E-01	**-8.218255E-01**	-8.199336E-01	-8.218355E-01	**-8.218810E-01**	-8.180641E-01
	MEDIA	**-7.470402E-01**	-7.426607E-01	-7.453253E-01	-7.449369E-01	-7.461825E-01	**-8.208687E-01**	-7.940088E-01	-8.002866E-01	-7.938682E-01	-7.934879E-01
	STD	**1.323339E-03**	1.061624E-02	5.884700E-03	5.865574E-03	4.350296E-03	**7.103893E-04**	1.839617E-02	1.604693E-02	1.769703E-02	1.856451E-02
C02	MEJOR	-2.277702E+00	-1.024535E+00	-2.277705E+00	**-2.277709E+00**	-2.277707E+00	-2.169248E+00	-1.831765E+00	-2.261698E+00	**-2.277254E+00**	-2.256826E+00
	MEDIA	**-2.258870E+00**	-1.024535E+00	-2.126006E+00	-2.036947E+00	-2.243296E+00	**-2.151424E+00**	8.023309E+03	-1.884621E+00	-1.957789E+00	-1.685913E+00
	STD	**2.389779E-02**	0.000000E+00	4.276089E-01	6.579564E-01	4.045046E-02	**1.197582E-02**	2.000646E+00	4.181442E-01	8.100867E-01	6.670892E-01
C03	MEJOR	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	2.867347E+01	2.867347E+01	2.867347E+01	2.867347E+01	**2.390000E-19**
	MEDIA	**0.000000E+00**	1.966720E+06	3.111000E-21	3.259289E-21	6.106018E-21	2.883785E+01	5.300617E+01	3.390278E+01	6.304622E+01	**1.605714E+01**
	STD	0.000000E+00	9.634921E+06	1.210000E-20	1.276625E-20	2.936590E-20	**8.047159E-01**	4.452446E+01	2.883039E+01	3.801309E+01	1.423313E+01
C04	MEJOR	**-9.922345E-06**		**-1.000000E-05**	-9.908198E-06	-9.909098E-06	4.608111E-03			2.039430E+01	**1.590548E-03**
	MEDIA	**-9.918452E-06**		3.885281E-03	1.102776E-03	2.835219E-02	**8.162973E-03**			2.039430E+01	2.354341E-02
	STD	**1.546730E-07**		6.036209E-03	1.018408E-03	1.327365E-01	3.067785E-03			**0.000000E+00**	2.771868E-02
C05	MEJOR	**-4.836106E+02**		-4.834950E+02	**-4.836106E+02**	**-4.836106E+02**	-4.531307E+02		-1.840970E+02	-4.363796E+02	**-4.629752E+02**
	MEDIA	**-4.836106E+02**		-1.793699E+02	-4.139718E+02	-3.674250E+02	**-4.495460E+02**		3.366424E+02	-2.502327E+02	-2.003758E+02
	STD	**3.890350E-13**		4.300943E+02	1.320853E+02	1.295727E+02	**2.899105E+00**		2.567318E+02	1.084181E+02	1.358803E+02
C06	MEJOR	**-5.786681E+02**	-5.789239E+02	-5.787048E+02	-5.784922E+02	-5.785581E+02	-5.285750E+02	-1.698576E+02	-2.475472E+02	-5.112752E+02	**-5.287628E+02**
	MEDIA	**-5.786528E+02**	-3.806216E+02	-3.761981E+02	-5.331547E+02	-4.906016E+02	**-5.279068E+02**	3.345201E+02	2.408127E+02	-2.794808E+02	-2.746376E+02
	STD	**3.627169E-03**	3.487267E+02	3.562107E+02	1.124846E+02	1.382472E+02	**4.748378E-01**	3.001146E+02	3.172443E+02	1.921440E+02	1.639472E+02
C07	MEJOR	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	**1.147112E-15**	9.900265E-10	1.610000E-09	1.598840E-10	4.470000E-09
	MEDIA	**0.000000E+00**	8.370000E-20	1.450000E-05	1.725137E-19	5.994037E-20	**2.603632E-15**	1.594651E-01	1.100000E-06	5.205032E-07	3.189305E-01
	STD	0.000000E+00	3.540000E-19	7.090000E-05	7.859764E-19	2.849813E-19	**1.233430E-15**	7.812155E-01	3.570000E-06	1.166507E-06	1.085144E+00
C08	MEJOR	**0.000000E+00**	3.820000E-28	**0.000000E+00**	**0.000000E+00**	2.335028E-27	**2.518693E-14**	2.993716E-12	6.930000E-11	5.573369E-10	4.990000E-10
	MEDIA	6.727528E+00	8.797871E+00	7.402312E+00	7.573865E+00	**5.949551E+00**	**7.831464E-14**	1.168660E+01	8.119647E+01	1.064030E+02	7.941454E+01
	STD	5.560648E+00	**4.492596E+00**	6.236881E+00	5.269496E+00	5.185212E+00	**4.855177E-14**	5.687029E+01	1.713878E+02	2.565145E+02	2.161634E+02
C09	MEJOR	**0.000000E+00**	1.952055E+00	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**2.770665E-16**	2.829875E-07	9.760000E-11	6.760814E-15	4.100000E-15
	MEDIA	**0.000000E+00**	3.433254E+01	4.880000E-10	3.767878E+10	1.720570E+09	**1.072140E+01**	1.193549E-02	3.076190E+02	1.800728E-11	6.460000E-11
	STD	**0.000000E+00**	2.876523E+01	2.070000E-11	1.853031E+11	7.584969E+09	**2.821923E+01**	9.847024E+01	9.578332E+02	8.723326E-11	2.150000E-11
C10	MEJOR	**0.000000E+00**	8.553841E-02	2.590000E-24	**0.000000E+00**	**0.000000E+00**	3.252002E+01	2.294922E+01	3.131214E+01	6.571556E+00	**1.271753E+01**
	MEDIA	**0.000000E+00**	3.162346E+02	1.270000E-11	1.31417E+09	5.007209E+00	**3.326175E+01**	1.704831E-12	9.720000E+11	1.514851E+10	2.360000E-11
	STD	**0.000000E+00**	3.387936E+02	5.100000E-11	5.569632E+09	1.335959E+01	**4.545577E-01**	7.620273E-12	4.350000E-12	6.771499E-10	1.120000E-12
C11	MEJOR	-1.522713E-03	-1.522713E-03	-1.522713E-03	-1.522713E-03	-1.522713E-03	-3.368462E-04	-3.922282E-04	-3.920000E-04	**-3.922914E-04**	-3.920000E-04
	MEDIA	**-1.522711E-03**	-1.522711E-03	-1.522711E-03	-1.522712E-03	-1.522712E-03	-2.863882E-04	-3.137403E-04	**-3.380000E-04**	-1.942464E-04	-2.580000E-04
	STD	**6.341035E-11**	1.560000E-09	1.280000E-09	4.585497E-10	4.909212E-10	**2.707605E-05**	6.331270E-05	1.080000E-04	1.493602E-04	2.270000E-04
C12	MEJOR	**-5.700899E+02**	-1.992458E+01	-5.700852E+02	-5.683500E+02	-1.992438E+01	-1.991453E+01	-1.992618E+01	**-1.992632E+01**	-1.992629E+01	-1.992441E+01
	MEDIA	**-3.367349E+02**	-4.306579E+02	-1.687469E+02	-2.354132E+01	2.724112E+01	3.562530E+02	-1.080196E+01	-1.938905E+01	**2.558735E+01**	-1.764895E+01
	STD	1.782166E+02	**7.553234E+01**	2.575207E+02	1.112317E+02	9.411851E+01	2.889253E+02	4.469290E+01	**2.269356E+02**	1.374969E+00	2.275455E+02
C13	MEJOR	**-6.842937E+01**	-6.842936E+01	-6.842936E+01	-6.842936E+01	-6.842936E+01	**-6.642473E+01**	-6.562204E+01	-6.490825E+01	-6.364725E+01	-6.533404E+01
	MEDIA	**-6.842936E+01**	-6.778787E+01	-6.376632E+01	-6.495900E+01	-6.427393E+01	**-6.535310E+01**	-6.344104E+01	-5.991589E+01	-6.141813E+01	-5.953884E+01
	STD	**1.025960E-06**	1.761360E+00	2.471246E+00	2.267258E+00	2.864785E+00	**5.733005E-01**	1.438881E+00	2.606253E+00	1.395707E+00	2.761740E+00
C14	MEJOR	**0.000000E+00**	5.570000E-27	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**5.015863E-14**	7.071555E-09	5.010000E-12	8.231785E-11	1.630000E-09
	MEDIA	**0.000000E+00**	5.100000E+08	1.594632E+01	3.560323E-16	2.102893E+01	**3.089407E-13**	1.074479E+01	6.554351E+00	3.841118E+01	1.207542E+03
	STD	**0.000000E+00**	2.420000E+09	7.812068E+01	1.644367E-15	1.030204E+02	**5.608409E-13**	5.206264E+01	2.266286E+01	1.298066E+02	5.843198E+03
C15	MEJOR	**0.000000E+00**	**0.000000E+00**	3.160000E-27	3.728472E-26	1.593491E-22	2.160345E+01	4.215002E+00	1.470000E-11	**7.108706E-12**	2.450000E-10
	MEDIA	**1.798978E-01**	1.060000E+10	3.059192E+06	4.407887E-01	1.727940E+08	**2.160576E+01**	3.331180E-12	6.080000E+09	1.278536E+09	3.680095E+01
	STD	**8.813156E-01**	4.740000E+10	1.500000E-07	1.193662E+00	6.100501E+08	**1.104834E-04**	1.089224E-13	2.980000E+10	6.263521E+09	8.381481E+01
C16	MEJOR	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	MEDIA	**3.702054E-01**	5.994882E-01	1.006303E-01	**9.453968E-02**	1.309922E-01	**2.168404E-21**	1.200782E-03	1.015864E-02	5.135254E-04	8.785803E-03
	STD	**3.710479E-01**	4.454175E-01	**1.968174E-01**	3.288865E-01	1.983350E-01	**1.062297E-20**	4.085354E-03	4.452883E-02	2.515750E-03	3.016258E-02
C17	MEJOR	**1.463180E-17**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	2.165719E-01	1.451447E-02	1.279146E-03	**9.264679E-06**	8.947486E-02
	MEDIA	**1.249561E-01**	7.306529E+00	2.376878E-01	7.561638E+00	7.701546E+00	**6.326487E+00**	2.135308E+02	2.464382E+00	2.883887E+01	2.042799E+01
	STD	**1.937197E-01**	1.714114E+01	5.408658E-01	3.677638E+01	3.717040E+01	**4.986691E+00**	4.122819E+02	8.412985E+00	1.084730E+02	3.400048E+01
C18	MEJOR	**3.731439E-20**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	1.223054E+00	1.526043E-04	6.291048E-02	1.089667E-07	**2.450000E-09**
	MEDIA	**9.678766E-19**	5.880000E-06	**0.000000E+00**	2.234673E-01	1.435694E-01	**8.754569E+01**	5.766742E-03	4.793664E+02	**3.811537E+00**	2.575481E-02
	STD	**1.811234E-18**	2.390000E-05	**0.000000E+00**	5.752416E+01	7.030496E+01	**1.664753E+02**	8.694744E-03	1.911847E+03	**7.299359E+00**	1.102910E-03

Los resultados de la Tabla 5.7 muestra que de las tres versiones presentadas, el método basado en el gradiente es la versión más robusta obteniendo soluciones factibles en todo el conjunto de funciones de prueba. Tomando en cuenta que son en total 18 funciones en 10D y 30D, el grado de mejora se encuentra en el rango del 22 al 27 % del total de funciones en 10D y del 50 al 66 % del total de funciones en 30D como se muestra en la Figura 5.7. Por ende, se tiene al algoritmo MBSO en un nivel de desempeño similar al algoritmo εDEag al resolver problemas de optimización numérica con restricciones. Esta similitud se puede observar en las Figuras 5.8, 5.9 en las cuales se presentan los resultados estadísticos obtenidos después de aplicar la prueba estadística post-hoc Bonferroni a los resultados finales. Los resultados estadísticos muestran que no existe diferencia significativa entre los algoritmos presentados.

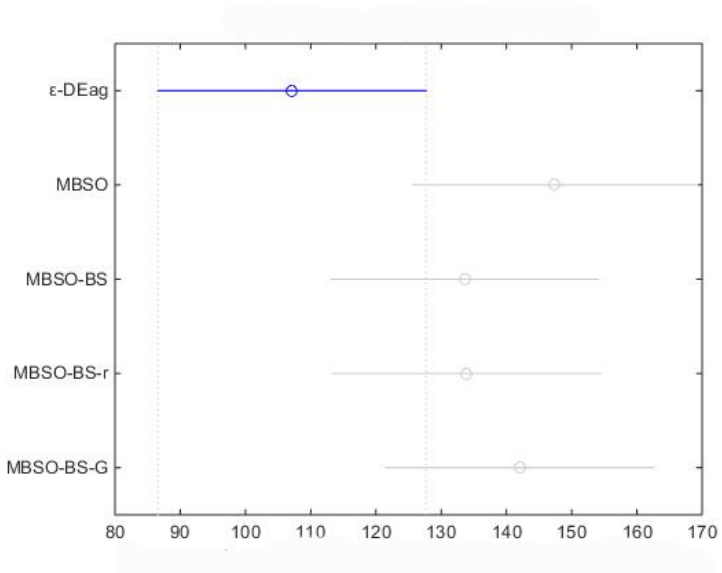

Figura 5.8: Prueba estadística post-hoc Bonferroni 10D.

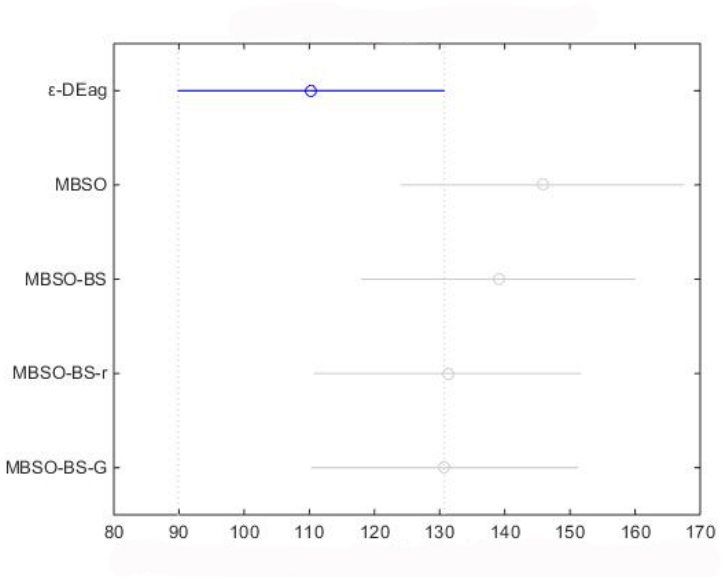

Figura 5.9: Prueba estadística post-hoc Bonferroni 30D.

5.7. Conclusiones

En este capítulo se analizó el desempeño del algoritmo MBSO [6] para resolver problemas de optimización restringida. Se mejoró el desempeño del algoritmo mediante el uso del operador especial de frontera BS y se propusieron dos nuevos operadores de frontera.

En el primer operador propuesto BS-r, se aplica la idea de la división de segmento de línea expuesto en Álgebra Lineal variando el factor r desde 0.1 hasta 0.9. La intención es explorar la frontera de la restricción utilizando tamaños de paso variables ya sea largos o cortos en lugar de solo obtener puntos medios como lo hace el operador BS y así poder disminuir el número de evaluaciones utilizados en este proceso. Los resultados mostraron que valores de r entre 0.1 y 0.7 permiten obtener los mejores resultados en la función objetivo encontrando soluciones factibles en la mayoría de las funciones con excepción de la función C04 en 30D debido a su dimensión y sus restricciones de igualdad. Aunque los resultados finales, en algunas funciones fueron mejorados, el número de evaluaciones no fueron disminuidas, esto es porque aunque se simulen tamaños de paso variables sigue sin conocerse la ubicación de la frontera de la restricción. Por lo tanto, el hecho de moverse dentro del espacio de búsqueda delimitado por la línea formada entre el punto factible y no factible mediante saltos largos o cortos fijos no proporciona información suficiente para alcanzar la frontera de la restricción ya que el método sigue moviéndose a ciegas dentro de ese espacio de búsqueda. En un trabajo futuro se debe buscar agregar mayor información al método para

aumentar la información disponible al momento de generar los nuevos puntos de tal manera que no se posicionen cerca del punto actual sino dentro del radio entre el punto actual y la frontera de la restricción. Por otro lado, los tamaños de paso fijos se deben transformar a tamaños de paso ajustables al radio de interés dentro del espacio de búsqueda.

El segundo operador propuesto BS-G, intenta explorar la frontera de las restricciones utilizando la información del gradiente de la restricción violada con el objetivo de alcanzar la frontera con mayor rapidez. El operador se aplicó en dos sentidos: En la primera opción (MBSO-BS-CG), se aplica en primer lugar el operador de frontera BS un cierto número de iteraciones, si el punto obtenido como resultado es no factible, entonces se aplica el operador BS-G a dicho punto con la finalidad de obtener un punto dentro o muy cerca de la frontera factible de la restricción. En la segunda opción (MBSO-BS-or-G), BS-G se utiliza como operador de frontera en lugar del operador BS, siempre y cuando no se cumpla la condición de entrada en el operador BS.

De las versiones presentadas (BS, BS-r y BS-G), la versión basada en gradiente se presenta como la más robusta obteniendo soluciones factibles en todas las funciones de prueba. Por lo anterior, si el problema permite cálculos de gradientes, un operador de frontera basado en gradiente representa la mejor opción para algoritmos como BSO y se evita el requerimiento de puntos factibles de entrada. Los resultados finales fueron comparados con el algoritmo εDEag mostrando un comportamiento similar en 10D y mejorando los resultados en mas del 50 % del total de funciones en 30D.

Además de poder explorar la frontera de la región factible con ayuda de los operadores especiales, es posible también acercarse con rapidez al área factible y poder alcanzar soluciones factibles disminuyendo el número de evaluaciones utilizadas en el proceso. En el Capítulo 6 se explorará un método de este tipo conocido como método de Consenso de Restricciones. Con este método se espera que el algoritmo MBSO alcance con mayor rapidez la zona factible, además, de obtener los mejores resultados del conjunto de prueba empleando un menor número de evaluaciones comparado con el algoritmo MBSO.

Capítulo 6

R+V: Una nueva variante del método de consenso de restricciones

Localizar la región factible y permanecer dentro de ella son dos grandes retos a los que se enfrenta cualquier método de búsqueda al tratar de resolver problemas restringidos, ya que en muchas ocasiones dentro del espacio de búsqueda la región factible suele ser de menor tamaño en comparación con la región no factible. Por lo tanto, dado que el objetivo de los métodos de búsqueda en este tipo de problemas es encontrar la mejor solución ubicada dentro de la región factible, encontrar esta región se convierte en una tarea no trivial para cualquiera de los método utilizados en este proceso [3], [45], [52].

En la literatura especializada existen tanto métodos que permiten mantener la factibilidad de una solución como métodos que tratan de obtener factibilidad a través de puntos no factibles reduciendo el número de evaluaciones requeridos para completar esta tarea. El inconveniente con los primeros métodos es el requerimiento de puntos factibles [11], [15], [48]. Dentro de la segunda clasificación, el método de consenso de restricciones (Constraint Consensus Method, CC) propuesto por John W. Chineck en 2004 [10] es de especial interés en este trabajo con el fin de obtener soluciones factibles competitivas en los problemas de prueba utilizados. El método de CC calcula un vector de factibilidad por cada restricción violada en el punto actual. Una vez que todos los vectores de factibilidad son calculados, el siguiente paso es combinarlos para formar un vector conocido como vector de consenso, el cual contiene tanto la dirección como la distancia necesaria para encontrar factibilidad. Algunos trabajos en donde el método de CC ha sido utilizado para resolver problemas de optimización restringida se presentan a continuación.

En 2004 Jhon W. Chineck [10] presentó el método de consenso de restricciones como un método que permite acercar un punto arbitrario no factible

a un punto relativamente cercano a la región factible requiriendo un bajo costo. Un conjunto con diferente tipo de restricciones, formas y complejidad fue utilizado para probar el método utilizando el software comercial MINOS [46]. En 2008 Walid Ibrahim y John W. Chinneck presentaron cinco nuevas variantes del método de consenso de restricciones [34]. Las variantes presentadas difieren únicamente en la forma de construcción del vector consenso. En las variantes FDnear y FDfar el vector consenso viene a ser el vector de factibilidad de menor y mayor longitud respectivamente. En las versiones restantes DBavg, DBmax, DBbnd, el vector consenso se conforma tomando en cuenta el signo (positivo o negativo) de las variables que conforman los vectores de factibilidad participantes. Para probar su propuesta resolvieron un conjunto de 231 funciones utilizando el software comercial CONOPT 3.13[19], SNOPT 6.1-1[26], MINOS 5.5[46], DONLP2[54], y KNITRO 4.0[58]. La variante DBmax resultó ser la más competitiva en este tipo de problemas. En 2013 Laurence Smith et al. [53] presentó una nueva versión al método de consenso de restricciones conocido como SUM, en la cual el vector consenso es el resultado de promediar los vectores de factibilidad obtenidos en el punto actual. Adicionalmente proponen la versión aumentada, donde la idea principal es buscar factibilidad utilizando la información del vector consenso y de la restricción violada en la generación anterior. En ese mismo año [52] los mismos autores presentaron el uso del método de consenso de restricciones para identificar regiones factibles disjuntas utilizando algoritmos con múltiples puntos de inicio.

En el párrafo anterior se puede ver como algoritmos clásicos de optimización han sido combinados con el método CC para dar solución a problemas sujetos a restricciones mostrando resultados competitivos. Recientemente, desde el 2011 hasta el 2016 [30], [31], [32] Hamza probó la combinación de este método con algoritmos inspirados en la naturaleza como son los Algoritmos Genéticos (GA) y la Evolución Diferencial (DE) en dos conjuntos de prueba, el primero cuenta con veinticuatro funciones y el segundo de treinta y seis funciones [39], [40]. El método de CC se utilizó como un método de preprocesa-miento, donde en cada iteración algunos puntos no factibles son seleccionados de la población actual y procesados por el método de CC. Los puntos obtenidos reemplazan a los peores en la población actual. En 2016 Hamza et al. [31], aplicaron el método dentro del operador de mutación utilizado en el algoritmo DE. La idea principal es generar un porcentaje de soluciones aplicando el nuevo operador para acercarse lo más rápido posible a la zona factible. La idea de combinar un algoritmo global con un operador especial logro superar tanto a la versión original de DE como a los algoritmos del estado del arte utilizados para comparar la propuesta.

Aunque los resultados obtenidos por la combinación del método CC con los algoritmos inspirados en la naturaleza muestran ser competitivos y alentadores, es necesario tener más evidencia que respalde este comportamiento.

Siguiendo la misma linea de investigación, en este capítulo se presenta una nueva variante del método de consenso de restricciones conocida como R+V en combinación con el algoritmo de inteligencia colectiva basado en el proceso de tormenta de ideas (MBSO). R+V construye el vector de consenso tomando en cuenta únicamente al vector de factibilidad generado por la restricción más violada en el punto actual. La variante propuesta es comparada con las versiones presentadas en [34] y [53] obteniendo resultados competitivos en el conjunto de treinta y seis funciones de prueba propuesto en [40]. Finalmente, la nueva variante en combinación con el algoritmo MBSO (MBSO-R+V) se compara con algoritmos del estado del arte mostrando un desempeño similar al resolver problemas de optimización numérica con restricciones, además de reportar un ahorro en el número de evaluaciones en comparación con el algoritmo MBSO.

6.1. Método de Consenso de Restricciones: Constraint Consensus Method (CC)

Consenso de restricciones es un método propuesto por John W. Chinneck en 2004 [10] basado en algoritmos de proyección. Partiendo de un punto no factible, el vector obtenido de la restricción violada es proyectado hacia el punto actual con el objetivo de disminuir su grado de violación y acercarlo a la región factible.

El vector obtenido a partir de la restricción violada, se conoce como vector de factibilidad fv, conformado por el grado de violación de la restricción $g_i(x)$, el gradiente obtenido de la restricción $\nabla g_i(x)$ y la longitud del gradiente $\|\nabla g_i(x)\|$. Al contemplar el gradiente de la restricción, el vector fv contiene tanto la distancia como la dirección necesarios para buscar factibilidad en el punto actual. Al existir una sola restricción en el problema, el vector de factibilidad automáticamente es usado para actualizar y obtener el nuevo punto, en caso contrario, se procede a obtener el vector de factibilidad por cada restricción violada.

Una vez generados los vectores de factibilidad, éstos se combinan de tal manera que se obtenga un solo vector donde se contemple la opinión de todas las restricciones y por supuesto pueda acercarnos lo más rápido posible hacia la región factible. El vector que resulta al combinar todos los vectores de factibilidad se denomina vector consenso y será utilizado para actualizar el punto no factible. Se espera que el vector consenso refleje la opinión de todos los vectores de factibilidad para cada restricción violada, así se buscará la factibilidad de tal manera que puedan satisfacerse o quedar representadas todas las restricciones a la vez. Por supuesto, se espera que el punto actualizado con el vector consenso resulte en un punto con un menor grado de violación de restricciones, aunque esto dependerá del método utilizado para formar el vector consenso.

El vector de factibilidad fv se genera a partir de la Ecuación (6.1).

$$fv = \frac{-g_i(x)\nabla g_i(x)}{\|\nabla g_i(x)\|^2} \tag{6.1}$$

Donde, $g_i(x)$ representa el valor de la violación en la restricción actual, $\nabla g_i(x)$ representa el vector obtenido al calcular el gradiente de la restricción actual y $\|\nabla g_i(x)\|$ indica la longitud del vector gradiente de la restricción actual.

El método de CC se creó con la idea de hacer frente a un conjunto de restricciones no convexas. Por lo tanto, resulta un cálculo exacto al tratarse de restricciones lineales, pero resulta sólo un estimado (basado en la proyección del gradiente de la restricción) en el caso de restricciones no lineales. Los pasos necesarios del método de CC se muestra en el Algoritmo 15.

Como puede observarse en el Algoritmo 15, el vector consenso se forma a nivel componente, es decir solo las variables participantes en cada restricción son tomadas en cuenta al momento de decidir el valor de cada variable que conformará el vector consenso. Lo anterior es porque no todas las variables de decisión que conforman la función objetivo participan en todas las restricciones asociadas al problema.

De acuerdo al método utilizado para generar el vector consenso existen diferentes variantes del método de CC. Cada una de las variantes se presentan en el siguiente apartado. En cada algoritmo que representa a cada variante $NINF$ es el número de restricciones violadas, s_j es la suma de los componentes del vector de factibilidad en la dimensión j, n_j es el número de restricciones violadas que contemplan la variable j, t es el vector consenso, z es el vector de factibilidad de mayor o menor longitud, s_j^+ y s_j^- son la suma de los componentes del vector de factibilidad en las direcciones positiva y negativa respectivamente a la variable j, n_j^+ y n_j^- es el número de restricciones violadas que votan en dirección positiva o negativa para la variable j, $n_j^{=+}$ y $n_j^{=-}$ representan el número de votos en dirección positiva o negativa para la variable j en las restricciones de igualdad, max_j^+ y max_j^- representan el componente del vector de factibilidad máximo positivo y negativo para la variable jth en las restricciones de desigualdad.

6.1.1. Variantes del método de CC

Con la finalidad de mejorar el desempeño del método de CC en la actualidad existen diferentes variantes las cuales difieren en el método utilizado para formar el vector consenso. A continuación se presentan las variantes mencionadas en la literatura [34].

Algoritmo 15 Algoritmo CC Básico

Un conjunto de I restricciones $c_1 ... c_I$;
Un punto inicial x;
Una tolerancia para la distancia de factibilidad α
Una tolerancia de movimiento β
Un número máximo de iteraciones μ
$contador = 0$
while $contador < \mu$ **do**
  $NINF$=0; for all $j : n_j = 0, s_j = 0$
  for Cada restricción c_i **do**
    if c_i es violada **then**
      Calcular el vector de factibilidad fv_i y su distancia $\|\nabla fv_i\|$
    end if
    if $\|fv_i\| > \alpha$ **then**
      $NINF = NINF + 1$
      for Cada variable x_j in c_i **do**
        $n_j = n_j + 1; s_j = s_j + fv_{ij}$
      end for
    end if
  end for
  Si $NINF = 0$, entonces, Terminar exitosamente
  for cada variable x_j **do**
    if $n_j \neq 0$ **then**
      $t_j = s_j / n_j$
    else
      $t_j = 0$
    end if
  end for
  if $\|t\| \leq \beta$ **then**
    Salida No satisfactoria
  end if
  $contador++$
end while
$x = x + t$; verificar que x_j este dentro de los límites de la variable

6.1.2. Basadas en distancia de factibilidad

Este tipo de algoritmos toma en cuenta la longitud del vector de factibilidad para formar el vector consenso.

- **FDNEAR**

  El vector de factibilidad de menor longitud se convierte automáticamente en el vector consenso. De acuerdo a los autores resulta conveniente satisfacer las restricciones con un menor grado de violación, dado que esto mantiene puntos en los que el gradiente obtiene una buena aproximación de la función [34].

- **FDFAR** Al contrario de FDNEAR, el vector de factibilidad de mayor longitud se convierte en el vector consenso en FDfar. Al seleccionar el vector de mayor longitud se generan movimientos más largos que permitirán moverse con mayor rapidez hacia la zona factible [34].

  El algoritmo tanto de FDNEAR como de FDFAR se presentan en el Algoritmo 16.

6.1.3. Basadas en dirección

Para cada variable que participa en el vector de factibilidad la variable del vector consenso se construye de la siguiente manera.

- **DBAVG**

  Para la variable en turno, en cada vector de factibilidad se verifica el signo (positivo o negativo) de la variable. De todos los vectores de factibilidad se tomará en cuenta únicamente el signo ganador. Ejemplo, ver Tabla 6.1: Si hay 4 vectores de factibilidad y para la variable en turno, en este caso $x2$, el vector $vf1$, $vf2$ y $vf4$ tienen signo positivo y solo el vector $vf3$ tiene signo negativo, entonces el signo ganador o dirección ganadora es el signo positivo y solo se tomará en cuenta los vectores $vf1$, $vf2$ y $vf4$ para conformar esta variable en el vector consenso. Si hubiera un empate de signos entonces todos los vectores de factibilidad participan para formar el vector consenso. La variable del vector consenso se construye promediando la variable $x2$, para este ejemplo, de los vectores participantes. Los pasos necesarios para implementar este método se presentan en el Algoritmo 17.

- **DBMAX** Al igual que en la variante DBAVG, se toman en cuenta los vectores con el signo ganador para formar el vector consenso. La diferencia es que en DBMAX de los vectores seleccionados la variable de máximo valor pasará automáticamente a conformar el vector consenso. Si hubiera un empate de signos entonces se toma la variable máxima

Algoritmo 16 Algoritmo FDnear, FDfar

Un conjunto de I restricciones $c_1 ... c_I$;
Un punto inicial x;
Una tolerancia para la distancia de factibilidad α
Una tolerancia de movimiento β
Un número máximo de iteraciones μ
$contador = 0$
while $contador < \mu$ **do**
  $NINF = 0; k = 0$ para toda j en $\vec{x} : n_j = 0, s_j = 0, z_j = 0$
  if $modo = near$ **then**
    $fd = \infty$
  else
    $fd = 0$
    for Cada restricción c_i **do**
      if c_i es violada **then**
        Calcular el vector de factibilidad fv_i y su distancia $\|\nabla fv_i\|$
        if $\|fv_i\| > \alpha$ **then**
          $NINF = NINF + 1$
          for Cada variable j in c_i **do**
            $n_j = n_j + 1; s_j = s_j + fv_{ij}$
            if ($modo = near$ and $\|\nabla fv_i\| < fd$) or ($modo = far$ and $\|\nabla fv_i\| > fd$) **then**
              $k = i$; $z = fv_i$
            end if
          end for
        end if
      end if
    end for
  end if
  if $NINF = 0$ **then**
    Terminar exitosamente
  end if
  for cada variable x_j **do**
    if xj aparece en c_k **then**
      $t_j = z_j$
    else
      if $n_j \neq 0$ **then**
        $t_j = s_j / n_j$
      else
        $t_j = 0$
      end if
    end if
  end for
  if $\|t\| < \beta$ **then**
    Salida No satisfactoria
  end if
  $x = x + t$; verificar que x_j este dentro de los límites de la variable
  $contador++$
end while

Algoritmo 17 Algoritmo DBAVG

Un conjunto de I restricciones $c_1 ... c_I$;
Un punto inicial x;
Una tolerancia para la distancia de factibilidad α
Una tolerancia de movimiento β
Un número máximo de iteraciones μ
$contador = 0$
while $contador < \mu$ **do**
  $NINF$=0; for all $j : n_j^+ = 0, n_j^- = 0, s_j^+ = 0, s_j^- = 0$
  for Cada restricción c_i **do**
    si c_i es violada, entonces, Calcular el vector de factibilidad fv_i y su distancia $\|\nabla fv_i\|$
    if $\|fv_i\| > \alpha$ **then**
      $NINF = NINF + 1$
      for Cada variable x_j in c_i **do**
        if $fv_{ij} > 0$ **then**
          $n_j^+ = n_j^+ + 1; s_j^+ = s_j^+ + fv_{ij}$
        end if
        if $fv_{ij} < 0$ **then**
          $n_j^- = n_j^- + 1; s_j^- = s_j^- + fv_{ij}$
        end if
      end for
    end if
  end for
  Si $NINF = 0$, entonces , terminar exitosamente
  for cada variable x_j **do**
    if $n_j^+ = n_j^-$ y $(n_j^+ + n_j^- > 0)$ **then**
      $t_j = (s_j^+ + s_j^-)/(n_j^+ + n_j^-)$
    else if $n_j^+ > n_j^-$ **then**
      $t_j = s_j^+/n_j^+$
    else
      $t_j = s_j^-/n_j^-$
    end if
  end for
  if $\|t\| \leq \beta$ **then**
    Salida No satisfactoria
  end if
  $x = x + t$; verificar que x_j este dentro de los límites de la variable
  $contador++$
end while

de cada dirección y los vectores dueños de estas variables son los participantes para formar el vector consenso y entonces la variable del vector consenso se construye promediando la variable en los vectores participantes, ver ejemplo en la Tabla 6.1. El Algoritmo 18 muestra los pasos necesarios para implementar esta variante.

- **DBBND** La dirección de movimiento toma en cuenta el número de votos en cada dirección y la longitud del movimiento depende del tipo de restricción ya sea de igualdad o desigualdad. Cada movimiento sugerido por las restricciones de igualdad es tomado en cuenta, pero solo movimientos largos son tomados en cuenta para restricciones de desigualdad ya que son estos movimientos los que las satisfacen. Los pasos necesarios para implementar este método se presentan en el Algoritmo 19.

- **SUM** En la variante SUM [53], el vector consenso se obtiene sumando los vectores de factibilidad. De esta manera todos los vectores de factibilidad participan en la formación del vector consenso, aunque al ser una suma de vectores, los vectores de factibilidad de mayor longitud tienen mayor influencia en el vector consenso. El algoritmo es similar al de Basic CC Algoritmo (15) sólo que en la variante SUM no se hace el promedio de vectores [53].

- **AUGMENTED**

  La versión aumentada de consenso de restricciones [53], usa un enfoque predictor-corrector para encontrar y ajustar la longitud del último vector consenso calculado en alguna generación anterior. El predictor lo constituye el vector consenso construido de igual forma que en la versión Basic CC. Por su parte, el corrector es un factor de relajación que se calcula independientemente por cada restricción violada, el cual es utilizado para ajustar la longitud del vector predictor sin cambiar su dirección. El promedio obtenido del factor de relajación es utilizado para formar el vector consenso [53]. El vector de factibilidad aumentado se calcula de acuerdo a la Ecuación (6.2).

$$fv = \frac{-g_i(x_t)}{g_i(x_t) - g_i(x_{t-1})}(vc) \tag{6.2}$$

  donde fv representa el vector de factibilidad aumentado calculado para la restricción violada actual; $g_i(x_t)$ es el valor de violación de la restricción actual, $g_i(x_{t-1})$ representa el valor de violación de la restricción actual en la generación anterior, vc representa el vector consenso obtenido previamente.

Algoritmo 18 Algoritmo DBMAX

Un conjunto de I restricciones $c_1 ... c_I$;
Un punto inicial x;
Una tolerancia para la distancia de factibilidad α
Una tolerancia de movimiento β
Un número máximo de iteraciones μ
$contador = 0$
while $contador < \mu$ **do**
  NINF=0; for all $j : n_j^+ = 0, n_j^- = 0, s_j^+ = 0, s_j^- = 0$
  for Cada restricción c_i **do**
    if c_i es violada **then**
      Calcular el vector de factibilidad fv_i y su distancia $\|\nabla fv_i\|$
    end if
    if $\|fv_i\| > \alpha$ **then**
      $NINF = NINF + 1$
      for Cada variable x_j in c_i **do**
        if $fv_{ij} > 0$ **then**
          $n_j^+ = n_j^+ + 1$
          if $fv_{ij} > s_j^+$ **then**
            $s_j^+ = fv_{ij}$
          end if
        else
          if $fv_{ij} < 0$ **then**
            $n_j^- = n_j^- + 1$
            if $fv_{ij} < s_j^-$ **then**
              $s_j^- = fv_{ij}$
            end if
          end if
        end if
      end for
    end if
  end for
  Si $NINF = 0$ entonces terminar exitosamente
  for cada variable x_j **do**
    if $n_j^+ = n_j^-$ y $(n_j^+ + n_j^- > 0)$ **then**
      $t_j = (s_j^+ + s_j^-)/(n_j^+ + n_j^-)$
    else if $n_j^+ > n_j^-$ **then**
      $t_j = s_j^+ / n_j^+$
    else
      $t_j = s_j^- / n_j^-$
    end if
  end for
  if $\|t\| \leq \beta$ **then**
    Salida No satisfactoria
  end if
  $x = x + t$; verificar que x_j este dentro de los límites de la variable
  $contador++$
end while

Algoritmo 19 Algoritmo DBBND

1: $contador = 0$
2: **while** $contador < \mu$ **do**
3: $NINF = 0$; for all $j : n_j^+ = 0, n_j^- = 0, n_j^{=+} = 0, n_j^{=-} = 0, s_j^+ = 0, s_j^- = 0, max_j^+ = 0, max_j^- = 0$
4: **for** Cada restricción c_i **do**
5: **if** c_i es violada **then**
6: Calcular el vector de factibilidad fv_i y su distancia $\|\nabla fv_i\|$
7: **if** $\|fv_i\| > \alpha$ **then**
8: $NINF = NINF + 1$
9: **for** Cada variable x_j in c_i **do**
10: **if** $fv_{ij} > 0$ **then**
11: $n_j^+ = n_j^+ + 1$
12: **if** es una restricción de igualdad **then**
13: $s_j^+ = s_j^+ + fv_{ij}$; $n_j^{=+} = n_j^{=+} + 1$
14: **else if** Si $fv_{ij} > max_j^+$ **then**
15: $max_j^+ = fv_{ij}$
16: **end if**
17: **else if** $fv_{ij} < 0$ **then**
18: $n_j^- = n_j^- + 1$
19: **if** es una restricción de desigualdad **then**
20: $s_j^- = s_j^- + fv_{ij}$; $n_j^{=-} = n_j^{=-} + 1$
21: **else if** $fv_{ij} < max_j^-$ **then**
22: $max_j^- = fv_{ij}$
23: **end if**
24: **end if**
25: **end for**
26: **end if**
27: **end if**
28: **end for**
29: $NINF = 0$ entonces terminar exitosamente
30: **for** cada variable x_j **do**
31: Si $max_j^+ \neq 0$, entonces $s_j^+ = s_j^+ + max_j^+$; $n_j^{=+} = n_j^{=+} + 1$
32: Si $max_j^- \neq 0$, entonces $s_j^- = s_j^- + max_j^-$; $n_j^{=-} = n_j^{=-} + 1$
33: **if** $n_j^+ = n_j^-$ **then**
34: $t_j = (s_j^+ + s_j^-)/(n_j^{=+} + n_j^{=-})$
35: **else if** $n_j^+ > n_j^-$ **then**
36: $t_j = s_j^+ / n_j^{=+}$
37: **else**
38: $t_j = s_j^- / n_j^{=-}$
39: **end if**
40: **end for**
41: Si $\|t\| \leq \beta$, Salida No satisfactoria
42: $x = x + t$; verificar que x_j este dentro de los límites de la variable
43: $contador++$
44: **end while**

Tabla 6.1: Ejemplo de construcción del vector consenso para los métodos DBAVG y DBMAX.

Vector de factibilidad	**Variables**				
	x1	**x2**	**x3**	**x4**	
vf1	-2	2	2	4	
vf2	2	1	-5	3	
vf3	-3	-1	2	-1	
vf4	0	5	-3	-4	
Vector Consenso					
DBavg	-2.5	2.66	-1	0.5	
DBmax	-3	5	-1.5	0	

6.2. R+V: una nueva variante del método de consenso de restricciones

Las variantes del método de consenso de restricciones presentadas en la sección anterior calculan el vector de factibilidad por cada restricción violada en el punto actual. Por lo tanto, el cálculo del gradiente es un paso obligado en este punto requiriendo de un alto costo computacional cuando el número de restricciones asociadas al problema aumenta. Para hacer frente a esta situación se propone la variante R+V la cual permite un ahorro computacional al calcular el gradiente únicamente para la restricción con mayor grado de violación sin importar el número de restricciones violadas en el punto actual. Por otro lado, dado que el vector consenso lo constituye el único vector de factibilidad construido no es necesario un método de combinación de vectores de factibilidad.

A diferencia de las variantes existentes, en la variante R+V existen sólo dos posibles condiciones de paro, esto se debe a que ya no es necesaria la construcción de un vector consenso dado que el vector de factibilidad directamente toma ese lugar. Las condiciones de paro son las siguientes:

- Cuando la distancia del vector de factibilidad es menor que una tolerancia α especificada.
- Cuando el número de iteraciones μ es alcanzado.

Los pasos necesarios para implementar la nueva variante R+V se presentan en el Algoritmo 20.

Algoritmo 20 Algoritmo R+V

1: Un conjunto de I restricciones $c_1 ... c_I$;
2: Un punto inicial no factible x;
3: Una tolerancia para la distancia de factibilidad α
4: Un número máximo de iteraciones μ
5: $contador = 0$
6: **while** $contador < \mu$ **do**
7: hd=RestriccionMasViolada(I)
8: fv = getVectorFactibilidad(hd)
9: **if** $fv > \alpha$ **then**
10: $x = x + fv$
11: verificar que x este dentro de los límites de las variables
12: **end if**
13: $contador++$
14: **end while**

6.3. Diseño experimental

En la Tabla 6.2 se presenta la configuración de parámetros utilizados en este trabajo, cabe destacar que los valores presentados son los propuestos por los autores en [31],[56], [63]. Se aplicaron las pruebas estadísticas no paramétricas, como rank sum de Wilcoxon o Kruskal-Wallis, debido a que no se asegura una distribución normal en los datos de prueba.

Con el objetivo de analizar el desempeño de la nueva variante se llevaron a cabo cuatro experimentos, los cuales se detallan a continuación.

- Experimento 1. Comparativo entre variantes de CC. En este apartado se presenta un comparativo entre las variantes de consenso de restricciones BASIC, FDFAR, DBMAX, DBBND, AUGMENTED y la nueva variante R+V propuesta en este capítulo. La finalidad del comparativo es conocer el desempeño de cada una de las variantes en el conjunto de funciones de prueba. Para cada función de prueba se generaron 100 puntos aleatorios, de los cuales sólo los no factibles serán considerados para ser procesados por cada una de las variantes.

- Experimento 2. Posición de la variante R+V dentro del algoritmo MBSO. La finalidad de este experimento es identificar la mejor posición de la variante R+V dentro del algoritmo MBSO la cual permita obtener resultados competitivos en el conjunto de funciones de prueba. De esta manera, analizando los tres operadores principales en los que se encuentra dividido el algoritmo MBSO se tomó la decisión de aplicar la variante R+V en estas tres posiciones y analizar su desempeño. Se realizaron por lo tanto tres experimentos aplicando la variante en cada operador, los cuales se mencionan a continuación.

    1. Experimento a), R+VE1: La variante R+V se colocó dentro del algoritmo MBSO antes de aplicar el operador de agrupamiento. De esta manera la función de la variante es preprocesar la población inicial la cual será utilizada por el algoritmo MBSO en etapas posteriores. Aleatoriamente se seleccionaron cinco puntos no factibles de la población inicial que se convertirán en puntos de entrada en la variante R+V. Los puntos obtenidos reemplazarán a los puntos seleccionados en la población inicial.
    2. Experimento b), R+VE2: La variante R+V es colocada dentro del operador de reemplazo. Si la nueva solución es no factible, entonces entra a la variante R+V con el objetivo de convertirse en una solución factible y reemplazar a la solución seleccionada en el operador.
    3. Experimento c): Tres situaciones se presentan en este experimento. Dado que el operador de cruza utiliza un vector base y un

vector diferencia para aplicar la cruza, tres posiciones son de interés para aplicar la variante R+V dentro del operador.

a) Experimento c1), R+VE3a: La variante R+V se aplica al vector base que será utilizado para generar las nuevas soluciones.

b) Experimento c2), R+VE3b: Siendo a, b los vectores utilizados en la operación diferencia del operador de cruza, la variante R+V se aplica al vector a el cual representa la dirección en dicha diferencia de vectores.

c) Experimento c3), R+VE3c: La variante R+V se aplica al vector base y al vector a utilizados en el operador de cruza.

- Experimento 3. Frecuencia de uso de la variante R+V dentro del algoritmo MBSO. La experiencia adquirida en algoritmos inspirados en la naturaleza indica que el número de soluciones no factibles disminuye en tanto que el número de generaciones se incrementa, por lo tanto, el mayor número de soluciones no factibles se concentra en las generaciones iniciales del proceso evolutivo (ver Figura 6.4). En este aspecto y siendo la fortaleza de los métodos basados en consenso de restricciones llegar rápidamente a la región factible, lo mas conveniente es utilizarlos cuando más soluciones no factibles existen en la población actual. En virtud de ello, la combinación de estos métodos con los algoritmos inspirados en la naturaleza se convierten en la oportunidad de obtener resultados competitivos en espacios restringidos ahorrando costo computacional medido por el número de evaluaciones de soluciones. Es así como se llega a la conclusión de aplicar la variante R+V únicamente en el 15 % inicial del total de las evaluaciones permitidas al algoritmo global. Además, con el propósito de reducir el número de evaluaciones, se decidió aplicar la variante en intervalos cada 5 generaciones. Los resultados obtenidos por cada una de las variantes se muestran en la Figura 6.5

- Experimento 4. Comparativo entre la mejor version obtenida por la combinación R+V y MBSO (MBSO:R+V) y algunos algoritmos del estado del arte.

Tabla 6.2: Configuración de parámetros.

MBSO	NP = 100, M =5, $p_{replace} = 0.2$, $p_{one} = 0.8$, $p_{oneCenter} = 0.4$, $p_{twoCenter} = 0.5$, $Max_{iter} = \frac{Max_{fes}}{NP}$
$\varepsilon-$Constrained	cp=5, TC = 0.2*Max_{iter}, θ = 0.2NP
Max_{fes}	200,000 , 600,000
R+V	$\mu = 5$, $\alpha = 0.000001$, $\beta = 0.0001$

Tabla 6.3: Resultados obtenidos por las variantes de CC en el conjunto de funciones del CEC2010 10D.

F	P.I. No Factibles	BASIC	FDFAR	DBMAX	AUGMENTED	DBBND	R+V
C01	1	1	1	1	1	1	1
C02	100	72	78	72	64	72	76
C03	100	100	100	100	98	100	100
C04	100	61	37	32	68	37	100
C05	100	67	65	71	64	67	59
C06	100	53	62	68	64	53	59
C07	65	65	65	65	65	65	65
C08	59	56	56	56	53	56	56
C09	100	46	46	46	100	46	46
C10	100	70	70	70	88	70	70
C11	100	100	100	100	100	100	100
C12	100	100	100	100	94	100	100
C13	100	100	100	100	98	100	100
C14	100	100	100	100	97	100	100
C15	100	57	50	55	63	55	49
C16	100	95	62	79	42	90	83
C17	100	53	52	52	50	50	49
C18	100	57	100	57	57	57	100
TOTAL	1625	1253	1244	1224	1266	1219	**1313**

6.4. Resultados y discusión

6.4.1. Experimento 1. Comparativo entre R+V y otras variantes de consenso de restricciones

El número de puntos no factibles obtenidos por cada problema se presentan en la columna 2 de la Tabla 6.3. Las otras columnas muestran el éxito obtenido por cada variante, el cual se mide por el número de puntos no factibles que fueron mejorados por cada variante. Debido a que resultados similares fueron obtenidos con 10D y 30D, únicamente los resultados en 10D son presentados. Aunque los resultados muestran un comportamiento muy similar entre las variantes presentadas, se puede observar que la variante AUGMENTED y la variante R+V se presentan como las más competitivas en la Tabla 6.3, especialmente en las funciones C04, C09 y C18. Si se comparan estas dos versiones por separado sobre el total de puntos mejorados por cada versión se tiene la gráfica de la Figura 6.1, donde se ve de forma más clara que la variante R+V muestra un mejor rendimiento por una diferencia de 44 puntos a favor en comparación con la variante AUGMENTED.

Los resultados de la Tabla 6.3 y la Figura 6.1 muestran que el tomar en cuenta la información de la restricción con mayor grado de violación en lugar del consenso de todas las restricciones violadas permite obtener mejores resultados en las funciones de prueba.

6.4.2. Experimento 2. R+V dentro del algoritmo MBSO

La prueba estadística de Wilcoxon fue aplicada a los resultados finales en este experimento. Los resultados se muestran en las Tablas 6.4 y 6.5 donde la variante R+VE1 se toma como base para realizar la prueba estadística.

Los resultados presentados en las Tablas 6.4 y 6.5 no muestran diferencia

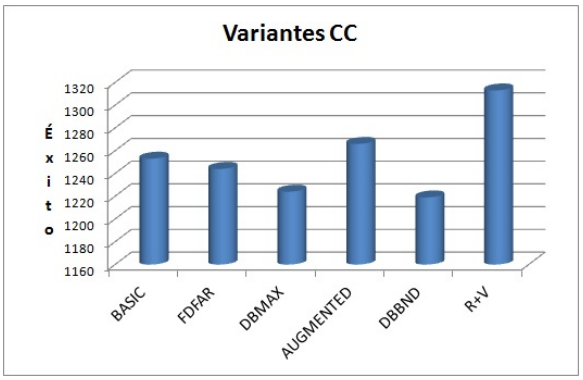

Figura 6.1: Comparativo entre las variantes de CC.

Tabla 6.4: Comparativo entre las variantes del Experimento 2, 10D.

Versiones	Criterio	Mejor	Igual	Peor	Decisión	p-value
R+VE1 vs R+VE2	**Mejor**	3	15	0	=	0.974277525
R+VE1 Vs R+VE3a		4	14	0	=	0.66585532
R+VE1 Vs R+VE3b		2	15	0	=	0.961219496
R+VE1 Vs R+VE3c		4	14	0	=	1
R+V E1 vs R+VE2	**Media**	2	15	1	=	0.987376927
R+VE1 Vs R+VE3a		3	14	1	=	0.874296698
R+VE1 Vs R+VE3b		2	16	0	=	0.824715242
R+VE1 Vs R+VE3c		4	14	0	=	0.447628106

Tabla 6.5: Comparativo entre las variantes del Experimento 2, 30D.

Versiones	Criterio	Mejor	Igual	Peor	Decisión	p_value
R+VE1 vs R+VE2	**Mejor**	3	15	0	=	0.843011125
R+VE1 Vs R+VE3a		4	13	1	=	0.679885581
R+VE1 Vs R+VE3b		5	12	1	=	0.800171553
R+VE1 Vs R+VE3c		4	12	2	=	0.861838193
R+V E1 vs R+VE2	**Media**	3	15	0	=	0.65591049
R+VE1 Vs R+VE3a		5	13	0	=	0.65591049
R+VE1 Vs R+VE3b		4	12	2	=	1
R+VE1 Vs R+VE3c		6	12	0	=	0.987378551

significativa a favor de ninguna de las variantes presentadas. En contraste, el análisis numérico de la Figura 6.2 y 6.3 muestra que aplicar la variante R+V antes del operador de agrupamiento permite al algoritmo MBSO obtener un mejor desempeño principalmente en el conjunto de funciones de dimensión 30.

Aunque el desempeño entre las variantes presentadas en este experimento parece muy similar, aplicar la variante en el operador de cruza tiene un grado de complejidad mayor ya que este operador se aplica por cada solución

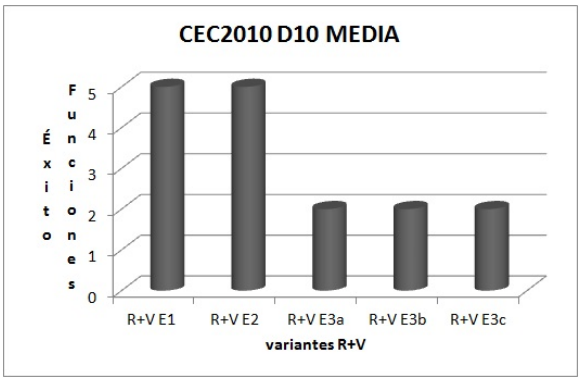

Figura 6.2: Mejor posición de la variante R+V dentro de MBSO en el conjunto de prueba CEC2010 con 10D.

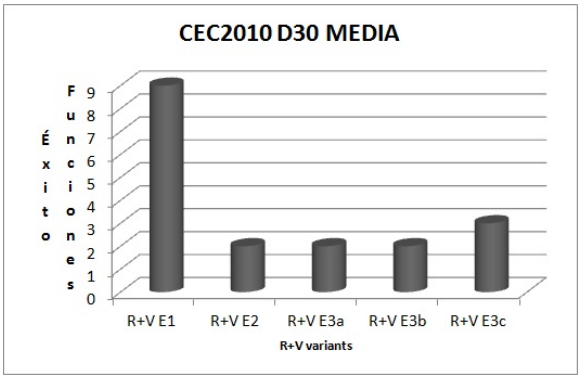

Figura 6.3: Mejor posición de la variante R+V dentro de MBSO en CEC2010 con 30D.

dentro de la población, por lo tanto, se tiene que seleccionar un parámetro probabilístico pf que indique la frecuencia de aplicación de la variante R+V. De otra forma su uso gastaría un total de NP X Max_{iter} evaluaciones afectando el desempeño del algoritmo. En este trabajo el parámetro pf tiene un valor de 0.1 el cual fue calculado empíricamente. Por otro lado, el desempeño del algoritmo al aplicar la variante R+V en el operador de reemplazo se ve afectado por el cumplimiento del parámetro $p_{replace}$, de no cumplirse esta condición la variante no se ejecuta en el algoritmo. Todas estas condiciones afectan el comportamiento global del algoritmo MBSO en combinación con la variante R+V a diferencia de aplicar la variante R+V antes del operador de agrupamiento, en la cual el algoritmo MBSO sigue con su flujo normal de procesamiento. Esta situación y los resultados mostrados hacen ver a la variante R+VE1 como la mejor opción para utilizarse dentro del algoritmo MBSO.

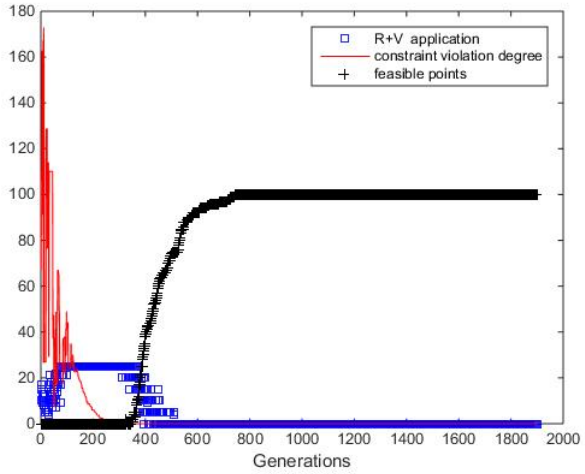

Figura 6.4: R+V aplicado durante todo el proceso evolutivo de la función 05 10D del CEC2010.

6.4.3. Experimento 3. R+V: Frecuencia de uso dentro del algoritmo MBSO

Los resultados de la Figura 6.5 y 6.6 indican que ejecutando la variante R+V esporádicamente en el intervalo sugerido, del 15 % de las primeras generaciones, permite al algoritmo MBSO ahorrar evaluaciones. En la Figura 6.5 y 6.6 se muestra que los mejores resultados se obtuvieron ejecutando la variante R+V cada 35 generaciones tanto con 10D como con 30D. Los mismos resultados muestran que el uso de la variante R+V en cada generación no necesariamente ayuda a decrementar el número de evaluaciones. Al contrario, el uso esporádico dentro del intervalo especificado es más conveniente. Este comportamiento es el esperado puesto que los operadores especiales son sólo métodos que proporcionan un sesgo particular al algoritmo global para alcanzar áreas de interés dentro del espacio de búsqueda, siendo el esfuerzo principal producido por el algoritmo global.

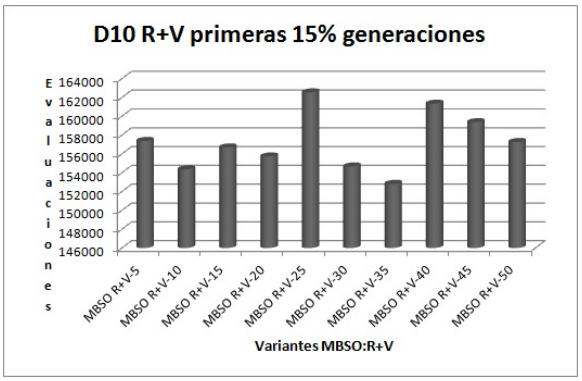

Figura 6.5: MBSO:R+V aplicado cada (5,10,15,20,25,30,35,40,45,50) generaciones durante el primer 15 % de las generaciones permitidas al algoritmo global en CEC2010 10D.

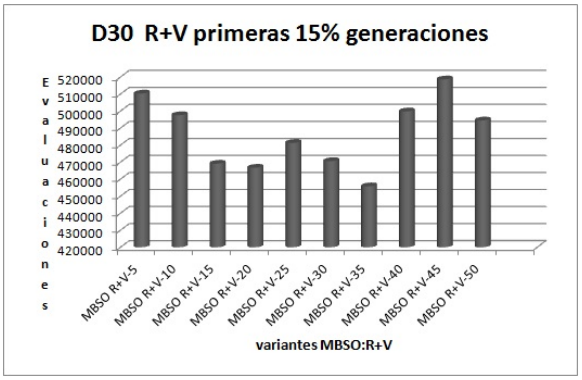

Figura 6.6: MBSO:R+V aplicado cada (5,10,15,20,25,30,35,40,45,50) generaciones durante las primeras 15 % de las generaciones permitidas al algoritmo global en CEC2010 30D.

6.4.4. Experimento 4. Comparativo entre el algoritmo MBSO-R+V y los algoritmos del estado del arte

En este apartado se presenta el comparativo entre el algoritmo MBSO:R+V y algoritmos representativos del estado del arte. El algoritmo de Evolución Diferencial (DE), el cual se muestra como un algoritmo competitivo tanto en espacios sin restricciones como en espacios restringidos, es el algoritmo global utilizado por los algoritmos del estado del arte. Los resultados del comparativo se muestran en las Tablas 6.6 y 6.7. Los algoritmos del estado del arte que se utilizaron en el comparativo son CoBe-MmDE [17], EMODE [20], SAMO-DE [21], ECHT-ARMOR-DE [28], DEbavDBmax [31], DE-DPS [50], ε-DEag [56].

Se aplicó la prueba estadística de Kruskal-Wallis y la prueba post-hoc de

Tabla 6.6: Comparativo entre MBSO-R+V y Algoritmos representativos del Estado del Arte. Parte 1

F	Algoritmo	10D			30D		
		Mejor	Promedio	DE	Mejor	Promedio	DE
C01	MBSO-R+V	**-7.47310E-01**	-7.43611E-01	7.85043E-03	-8.21883E-01	-7.86077E-01	1.98774E-02
	CoBe-MDE	**-7.47310E-01**	-7.34552E-01	2.71701E-02	**-8.21884E-01**	-7.37349E-01	2.56417E+00
	HAMZA-2016	**-7.47310E-01**	-7.46399E-01	2.55289E-03	**-8.21884E-01**	-8.13563E-01	7.85460E-03
	E-MODE	**-7.47310E-01**	**-7.47310E-01**	2.45131E-16	**-8.21884E-01**	-8.19898E-01	2.86297E-03
	SAMO-DE	**-7.47310E-01**	-7.47040E-01	1.35064E-03	**-8.21884E-01**	-8.14367E-01	4.76600E-03
	εDEag	**-7.47310E-01**	-7.47040E-01	1.32334E-03	-8.21826E-01	-8.20869E-01	**7.10389E-04**
	DE-DPS	**-7.47310E-01**	**-7.47310E-01**	**2.26623E-16**	**-8.21884E-01**	**-8.21204E-01**	1.79648E-03
	ECHT-ARMOR-DE	-7.47300E-01	-7.47000E-01	1.40000E-03	-8.18060E-01	-7.89920E-01	2.51000E-02
C02	MBSO-R+V	-2.19851E+00	-1.94946E+00	3.76873E-01	-2.23316E+00	-1.91178E+00	6.76282E-01
	CoBe-MDE	-2.23826E+00	-2.06665E+00	8.67312E-02	-2.21795E+00	-1.99525E+00	1.42773E-01
	HAMZA-2016	**-2.27771E+00**	-2.19837E+00	8.38973E-02	**-2.28097E+00**	**-2.27665E+00**	**3.15042E-03**
	E-MODE	**-2.27771E+00**	**-2.27771E+00**	**2.54021E-10**	**-2.28097E+00**	-2.27623E+00	4.89723E-03
	SAMO-DE	**-2.27771E+00**	-2.27684E+00	1.15496E-03	-2.28096E+00	-2.27611E+00	3.70600E-03
	εDEag	-2.27770E+00	-2.25887E+00	2.38978E-02	-2.16925E+00	-2.15142E+00	1.19758E-02
	DE-DPS	**-2.27771E+00**	-2.27751E+00	2.54035E-04	-2.28067E+00	-2.24463E+00	5.20548E-02
	ECHT-ARMOR-DE	-2.27770E+00	-2.27700E+00	3.30000E-03	-2.26070E+00	-2.17060E+00	7.36000E-02
C03	MBSO-R+V	0.00000E+00	2.43000E-17	1.16000E-16	3.61000E-11	5.98409E+01	4.35583E+01
	CoBe-MDE	0.00000E+00	**0.00000E+00**	**0.00000E+00**	2.11786E-06	1.24536E-05	1.02138E-05
	HAMZA-2016	0.00000E+00	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	6.38217E-25	7.13348E-25
	E-MODE	0.00000E+00	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**3.12360E-25**	**5.72975E-25**
	SAMO-DE	0.00000E+00	4.17300E-23	1.64051E-22	4.59960E-24	4.82600E-22	1.14600E-21
	εDEag	0.00000E+00	**0.00000E+00**	**0.00000E+00**	2.86735E+01	2.88379E+01	8.04716E-01
	DE-DPS	0.00000E+00	**0.00000E+00**	**0.00000E+00**	1.62000E-19	1.84790E-13	4.17994E-13
	ECHT-ARMOR-DE	0.00000E+00	0.00000E+00	0.00000E+00	2.58010E-24	2.63800E+01	7.94000E+00
C04	MBSO-R+V	**-1.00000E-05**	8.18634E-02	2.20385E-01	1.93682E+01	1.93682E+01	**0.00000E+00**
	CoBe-MDE	-9.99939E-06	-9.99499E-06	4.59939E-09	4.09359E-04	5.47581E-02	1.83033E-01
	HAMZA-2016	**-1.00000E-05**	**-1.00000E-05**	**0.00000E+00**	-3.33296E-06	-3.33087E-06	2.77215E-09
	E-MODE	**-1.00000E-05**	**-1.00000E-05**	**0.00000E+00**	**-3.33333E-06**	**-3.33333E-06**	2.45925E-16
	SAMO-DE	**-1.00000E-05**	**-1.00000E-05**	1.44608E-11	-3.24800E-06	-2.41130E-06	4.49234E-07
	εDEag	-9.99235E-06	-9.91845E-06	1.54673E-07	4.69811E-03	8.16297E-03	3.06779E-03
	DE-DPS	**-1.00000E-05**	**-1.00000E-05**	9.09633E-15	-3.33180E-06	-3.31230E-06	2.03926E-08
	ECHT-ARMOR-DE	**-1.00000E-05**	**-1.00000E-05**	**0.00000E+00**	-3.33260E-06	8.37130E-02	2.89000E-01
C05	MBSO-R+V	**-4.83611E+02**	-3.88419E+02	2.40866E+02	-4.83487E+02	-4.00802E+02	1.19541E+02
	CoBe-MDE	**-4.83611E+02**	**-4.83611E+02**	2.42094E-10	-3.66104E+02	-1.89424E+02	7.81926E+01
	HAMZA-2016	**-4.83611E+02**	**-4.83611E+02**	4.94808E-06	**-4.83611E+02**	**-4.83611E+02**	7.16023E-09
	E-MODE	**-4.83611E+02**	**-4.83611E+02**	3.48093E-13	**-4.83611E+02**	**-4.83611E+02**	**2.03634E-13**
	SAMO-DE	**-4.83611E+02**	**-4.83611E+02**	4.14428E-06	**-4.83611E+02**	**-4.83611E+02**	5.38991E-06
	εDEag	**-4.83611E+02**	**-4.83611E+02**	3.89035E-13	-4.53131E+02	-4.49546E+02	2.89911E+00
	DE-DPS	**-4.83611E+02**	**-4.83611E+02**	1.25826E-10	**-4.83611E+02**	**-4.83611E+02**	4.42074E-06
	ECHT-ARMOR-DE	-4.83610E+02	-4.83610E+02	**0.00000E+00**	-4.81220E+02	-4.33350E+02	1.46000E+02
C06	MBSO-R+V	-5.78308E+02	-4.84418E+02	2.75726E+02	-5.29945E+02	-3.30283E+02	3.67816E+02
	CoBe-MDE	**-5.78662E+02**	**-5.78662E+02**	2.17474E-07	-5.22832E+02	-4.79459E+02	7.41970E+01
	HAMZA-2016	-5.75505E+02	-5.75505E+02	7.67776E-07	**-5.30638E+02**	-5.30566E+02	2.34649E-01
	E-MODE	**-5.78662E+02**	**-5.78662E+02**	**3.24887E-13**	**-5.30638E+02**	**-5.30638E+02**	**4.66102E-10**
	SAMO-DE	**-5.78662E+02**	-5.78656E+02	9.35219E-03	-5.30637E+02	-5.30616E+02	1.28805E-02
	εDEag	-5.78658E+02	-5.78653E+02	3.62717E-03	-5.28575E+02	-5.27907E+02	4.74838E-01
	DE-DPS	**-5.78662E+02**	**-5.78662E+02**	8.05379E-04	**-5.30638E+02**	-5.30633E+02	5.89364E-03
	ECHT-ARMOR-DE	-5.78660E+02	-5.78660E+02	4.00000E-13	-5.24650E+02	-4.89310E+02	1.32000E+02
C07	MBSO-R+V	1.48000E-26	1.59463E-01	7.81207E-01	1.30000E-09	3.18933E-01	1.08154E+00
	CoBe-MDE	**0.00000E+00**	1.66108E-01	8.13757E-01	4.25873E+00	7.69379E+01	8.04727E+01
	HAMZA-2016	**0.00000E+00**	1.25029E-28	2.27457E-28	**0.00000E+00**	1.59465E-01	7.97325E-01
	E-MODE	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**1.60513E-27**	**4.13547E-27**
	SAMO-DE	**0.00000E+00**	7.76275E-23	3.88080E-22	9.49525E-23	1.78279E-13	3.62804E-13
	εDEag	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.14711E-15	2.60363E-15	1.23343E-15
	DE-DPS	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	5.48786E-20	1.03359E-13	2.20540E-13
	ECHT-ARMOR-DE	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.07890E-25	2.20000E-25
C08	MBSO-R+V	1.95000E-22	9.89005E+00	6.14239E+00	5.61000E-08	3.19457E+01	8.63495E+01
	CoBe-MDE	**0.00000E+00**	6.48219E+00	5.72333E+00	2.16719E-01	1.16733E+03	1.37638E+03
	HAMZA-2016	**0.00000E+00**	9.21687E+00	3.59585E+00	1.01271E-26	4.20083E-26	4.91772E-26
	E-MODE	**0.00000E+00**	1.00662E+01	3.02958E+00	**0.00000E+00**	**1.37398E-27**	**4.09125E-27**
	SAMO-DE	**0.00000E+00**	**2.52009E-25**	**1.26005E-24**	6.42581E-21	1.03293E-09	2.37330E-09
	εDEag	**0.00000E+00**	6.72753E+00	5.56065E+00	2.51869E-14	7.83146E-14	4.85518E-14
	DE-DPS	**0.00000E+00**	3.95039E+00	5.01904E+00	4.57572E-13	3.44757E-09	8.86366E-09
	ECHT-ARMOR-DE	**0.00000E+00**	7.52620E+00	5.00000E+00	**0.00000E+00**	2.01010E+01	4.70000E+01
C09	MBSO-R+V	**0.00000E+00**	1.24000E+10	6.07000E+10	2.54000E-15	9.00000E+10	4.39000E+11
	CoBe-MDE	**0.00000E+00**	3.92029E+00	1.64557E+01	4.69352E+00	4.24395E+05	1.30292E+06
	HAMZA-2016	**0.00000E+00**	4.53818E-26	1.83954E-25	**0.00000E+00**	4.30250E-26	4.47479E-26
	E-MODE	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**9.27762E-27**	**1.81923E-26**
	SAMO-DE	**0.00000E+00**	5.08975E+00	2.41083E+01	1.18617E-20	6.07967E+00	1.43260E+01
	εDEag	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	2.77067E-16	1.07214E+01	2.82192E+01
	DE-DPS	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	3.81580E-23	5.29059E-14	1.27939E-13
	ECHT-ARMOR-DE	**0.00000E+00**	1.76330E-01	8.80000E-01	**0.00000E+00**	4.61100E+00	2.31000E+01

Tabla 6.7: Continuación: Comparativo entre MBSO-R+V y Algoritmos representativos del Estado del Arte. Parte 2

F	Algoritmo	10D			30D		
		Mejor	Promedio	DE	Mejor	Promedio	DE
C10	MBSO-R+V	1.91000E-26	1.65350E+01	2.05163E+01	6.54698E+00	9.92844E+01	2.55637E+02
	CoBe-MDE	**0.00000E+00**	3.47716E+00	1.17805E+01	5.13956E+00	1.27313E+03	3.55121E+03
	HAMZA-2016	**0.00000E+00**	1.33352E-26	3.33891E-26	1.50649E-26	4.41228E-20	1.21180E-19
	E-MODE	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**6.98791E-27**	**1.42640E-26**
	SAMO-DE	**0.00000E+00**	4.46766E-01	1.54630E+00	9.76900E-21	1.96078E+01	2.12375E+01
	εDEag	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	3.25200E+01	3.32618E+01	4.54558E-01
	DE-DPS	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.63778E-21	2.23928E-13	6.24058E-13
	ECHT-ARMOR-DE	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	6.02090E-13	6.55360E+01	1.07000E+02
C11	MBSO-R+V	**-1.52271E-03**	**-1.52271E-03**	3.36000E-10	-3.92000E-04	-1.48000E-04	2.19000E-04
	CoBe-MDE	**-1.52271E-03**	**-1.52271E-03**	2.27150E-10	-3.81894E-04	-3.81894E-04	-3.81894E-04
	HAMZA-2016	**-1.52271E-03**	**-1.52271E-03**	1.46160E-14	**-3.92344E-04**	**-3.92344E-04**	3.98468E-10
	E-MODE	**-1.52271E-03**	**-1.52271E-03**	**8.90202E-17**	**-3.92344E-04**	**-3.92344E-04**	**1.07480E-10**
	SAMO-DE	**-1.52271E-03**	**-1.52271E-03**	3.66761E-09	-3.92300E-04	-3.86900E-04	6.14966E-06
	εDEag	**-1.52271E-03**	**-1.52271E-03**	6.34104E-11	-3.26846E-04	-2.86388E-04	2.70761E-05
	DE-DPS	**-1.52271E-03**	**-1.52271E-03**	3.14275E-12	**-3.92344E-04**	-3.92342E-04	8.77688E-10
	ECHT-ARMOR-DE	-1.52270E-03	-1.52270E-03	4.40000E-02	-3.92340E-04	-3.92340E-04	5.28000E-03
C12	MBSO-R+V	-5.70051E+02	-3.01914E+01	1.27246E+02	-1.99259E-01	4.18288E-01	1.94925E+00
	CoBe-MDE	-3.03784E+02	-2.31934E+01	6.79731E+01	-1.99117E-01	-1.49136E-01	7.10500E-02
	HAMZA-2016	-2.20251E-01	-2.20250E-01	6.62040E-07	**-1.99264E-01**	**-1.99264E-01**	**1.63099E-09**
	E-MODE	-4.23133E+02	-9.46103E+01	1.47632E+02	-1.99263E-01	-1.99263E-01	2.90292E-09
	SAMO-DE	**-5.70090E+02**	-1.16613E+02	1.83001E+02	-1.99260E-01	-1.99257E-01	1.32189E-06
	εDEag	**-5.70090E+02**	**-3.36735E+02**	1.78217E+02	-1.99145E-01	3.56233E+02	2.88925E+02
	DE-DPS	-1.99250E-01	-1.99250E-01	4.19599E-10	**-1.99264E-01**	-1.99260E-01	2.60287E-08
	ECHT-ARMOR-DE	-1.99250E-01	-1.99250E-01	**1.60000E-13**	-1.99260E-01	-1.60760E-01	1.93000E-01
C13	MBSO-R+V	**-6.84294E+01**	-6.46314E+01	2.12383E+00	-6.43291E+01	-6.05827E+01	2.38479E+00
	CoBe-MDE	-6.35253E+01	-5.72131E+01	8.52298E+00	-6.28397E+01	-5.76379E+01	2.56417E+00
	HAMZA-2016	**-6.84294E+01**	-6.75089E+01	1.41478E+00	-6.58321E+01	-6.01806E+01	5.06689E+00
	E-MODE	**-6.84294E+01**	-6.84293E+01	2.96573E-04	-6.84286E+01	-6.51674E+01	1.91800E+00
	SAMO-DE	**-6.84294E+01**	**-6.84294E+01**	1.54288E-07	**-6.84294E+01**	**-6.81918E+01**	**3.89164E-01**
	εDEag	**-6.84294E+01**	**-6.84294E+01**	1.02596E-06	-6.64247E+01	-6.53531E+01	5.73301E-01
	DE-DPS	**-6.84294E+01**	**-6.84294E+01**	**0.00000E+00**	**-6.84294E+01**	-6.63314E+01	2.08493E+00
	ECHT-ARMOR-DE	-6.84290E+01	-6.71690E+01	2.10000E+00	-6.74160E+01	-6.46460E+01	1.97000E+00
C14	MBSO-R+V	3.73000E-26	4.69000E-05	2.29000E-04	1.57000E-12	1.23559E+02	5.61397E+02
	CoBe-MDE	**0.00000E+00**	6.64430E-01	1.51766E+00	1.97420E+01	1.93110E+06	4.98750E+06
	HAMZA-2016	**0.00000E+00**	3.30388E-27	4.71892E-27	**0.00000E+00**	1.01059E-25	8.05286E-26
	E-MODE	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**2.07587E-27**	**5.75660E-27**
	SAMO-DE	**0.00000E+00**	1.20638E-21	2.43593E-21	1.74700E-22	1.19691E-08	2.56942E-08
	εDEag	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	5.01586E-14	3.08941E-13	5.60841E-13
	DE-DPS	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	8.92580E-21	5.13741E-14	1.31791E-13
	ECHT-ARMOR-DE	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.58090E-27	6.61350E+02	2.47000E+03
C15	MBSO-R+V	4.06000E-22	8.74000E+11	3.22000E+12	1.38000E-09	1.11000E+12	5.42000E+12
	CoBe-MDE	**0.00000E+00**	1.87394E-01	9.18037E-01	1.46048E+06	3.35372E+09	6.84313E+09
	HAMZA-2016	**0.00000E+00**	2.95843E-25	1.08632E-24	5.02646E-26	1.86828E-22	2.96870E-22
	E-MODE	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**2.53682E-27**	**6.80738E-27**
	SAMO-DE	**0.00000E+00**	7.05380E-04	2.44138E-03	5.84240E-18	2.11281E+00	4.51067E+00
	εDEag	**0.00000E+00**	1.79898E-01	8.81316E-01	2.16035E+01	2.16038E+01	1.10483E-04
	DE-DPS	**0.00000E+00**	5.43891E-26	2.19329E-25	6.33626E-20	1.95781E-13	5.22687E-13
	ECHT-ARMOR-DE	**0.00000E+00**	2.82460E+00	1.60000E+00	1.17160E-04	3.13160E+08	1.20000E+09
C16	MBSO-R+V	**0.00000E+00**	2.75811E-01	3.66072E-01	**0.00000E+00**	4.79015E-02	2.01555E-01
	CoBe-MDE	**0.00000E+00**	4.53657E-01	5.12495E-01	1.93179E-14	3.98805E-03	1.28401E-02
	HAMZA-2016	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**
	E-MODE	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**
	SAMO-DE	**0.00000E+00**	6.46960E-03	1.08704E-02	**0.00000E+00**	4.16070E-03	7.67790E-03
	εDEag	**0.00000E+00**	3.70205E-01	3.71048E-01	**0.00000E+00**	2.16840E-21	1.06230E-20
	DE-DPS	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**
	ECHT-ARMOR-DE	**0.00000E+00**	2.84780E-02	5.00000E-02	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**
C17	MBSO-R+V	**0.00000E+00**	4.10073E+01	1.17585E+02	6.93579E-03	1.88444E+02	3.46971E+02
	CoBe-MDE	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	4.11832E-10	6.49249E-04	3.17945E-03
	HAMZA-2016	1.34661E-29	7.05345E-19	1.07920E-18	1.06123E-18	1.81193E-12	5.28194E-12
	E-MODE	**0.00000E+00**	3.41749E-30	1.70721E-29	1.95482E-32	**2.77147E-21**	**8.43914E-21**
	SAMO-DE	**0.00000E+00**	1.26550E-23	3.21799E-23	**0.00000E+00**	1.02263E-10	1.45515E-10
	εDEag	1.46318E-17	1.24956E-01	1.93720E-01	2.16572E-01	6.32649E+00	4.98669E+00
	DE-DPS	**0.00000E+00**	1.06127E-24	3.88726E-24	3.23601E-12	8.76619E-02	1.55966E-01
	ECHT-ARMOR-DE	**0.00000E+00**	3.69780E-33	3.10000E-33	3.35640E-16	4.03360E-01	3.51000E-01
C18	MBSO-R+V	**0.00000E+00**	1.28565E+02	4.18874E+02	3.78000E-12	2.08031E+02	9.55766E+02
	CoBe-MDE	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	3.27286E-13	6.47806E+01	1.45721E+02
	HAMZA-2016	4.23953E-25	3.88548E-24	4.33667E-24	1.03010E-18	2.82943E-01	1.35637E+00
	E-MODE	**0.00000E+00**	1.52842E-32	2.19609E-32	**1.27764E-28**	**1.34036E-20**	**6.55039E-20**
	SAMO-DE	1.17400E-23	4.44799E-19	6.63671E-19	9.19280E-18	2.56810E-09	6.98480E-09
	εDEag	3.73144E-20	9.67877E-19	1.81123E-18	1.22605E+00	8.75457E+01	1.66475E+02
	DE-DPS	**0.00000E+00**	3.20908E-23	7.12457E-23	6.74774E-13	9.45991E-08	1.91224E-07
	ECHT-ARMOR-DE	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**

Bonferroni. Los resultados se presentan en la Figura 6.7 en los cuales no se observa diferencia significativa entre la media y el mejor valor encontrado en las funciones objetivo. Únicamente se reporta diferencia significativa en los valores obtenidos por la desviación estándar con excepción de los algoritmos CoBe-MDE y SAMO-DE. Con base en los resultados se puede concluir que el algoritmo MBSO-R+V es un algoritmo competitivo con los algoritmos del estado del arte para resolver problemas de optimización con restricciones.

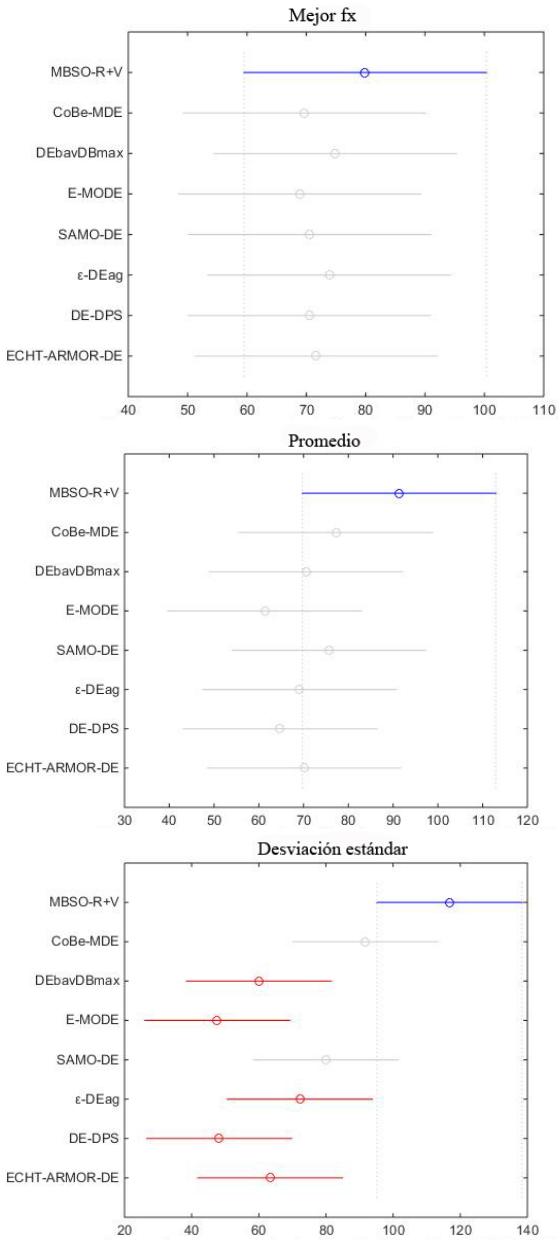

Figura 6.7: Resultados de la prueba Post-Hoc de Bonferroni. Mejor valor, mejor media y mejor desviación estándar obtenidos en el conjunto de funciones de prueba. Comparativo entre MBSO-R+V y los algoritmos representativos del estado del arte.

6.5. Conclusiones

En este capítulo se presentó una nueva variante del método de consenso de restricciones como operador especial por el algoritmo MBSO. En la nueva variante (R+V) el vector consenso se construye a partir del vector de factibilidad calculado únicamente para la restricción con mayor grado de violación. La variante R+V fue comparada con las variantes presentadas en [34] y [53] resolviendo el conjunto de treinta y seis funciones del CEC2010. Entre las variantes presentadas se tiene a R+V como la variante mas competitiva en este tipo de problemas ya que fue la que reportó mayor número de soluciones factibles y redujo la violación de las restricciones en un mayor número de puntos no factibles. Demostrando que al obtener información de la restricción con mayor grado de violación se generan resultados similares que el consultar la información de las diferentes restricciones violadas en el punto actual. Por otra parte, permite reducir costo computacional, medido por el número de evaluaciones, al evitar el cálculo del vector de factibilidad por cada restricción violada antes de decidir quién se convertirá en el vector consenso. La nueva variante sólo calcula un vector de factibilidad sin importar el número de restricciones violadas por el punto actual, lo cual lo hace atractivo respecto de las variantes inspiradas en el mismo método. Este vector de factibilidad es el que se convertirá automáticamente en el vector consenso. Agregar la variante R+V al algoritmo MBSO [6] hizo que se mejorara su desempeño resaltando sus habilidades para resolver problemas de optimización con restricciones. Los resultados indicaron que posicionando la variante R+V antes del operador de agrupamiento dentro del ciclo principal del algoritmo MBSO, y aplicando la variante durante las primeras 15 % del total de generaciones permitidas al algoritmo global, hacen del nuevo algoritmo MBSO:R+V un algoritmo competitivo al obtener resultados similares a los obtenidos con algoritmos representativos del estado del arte. Además, requirió un menor número de evaluaciones comparado con el algoritmo MBSO original al necesitar un promedio de 535,968.75 evaluaciones por el algoritmo MBSO y un promedio de 455,821.27 evaluaciones para el algoritmo MBSO:R+V en 30D, sin mostrar diferencia en 10D. Esta situación muestra que la variante R+V permite al algoritmo MBSO:R+V alcanzar la zona factible con mayor rapidez en el conjunto de prueba utilizado en este trabajo.

Capítulo 7

Conclusiones y Trabajo Futuro

7.1. Conclusiones

En este trabajo de tesis se utilizaron dos operadores especiales en combinación con el algoritmo de inteligencia colectiva basado en tormenta de ideas (BSO) para resolver problemas de optimización con restricciones. En primer lugar, dado que BSO es un algoritmo exitoso en espacios sin restricciones pero no analizado en espacios restringidos, el primer paso fue asegurar su buen desempeño en problemas de optimización con restricciones. Para esto, se hizo un comparativo de tres versiones del algoritmo BSO con tres manejadores de restricciones para definir el desempeño de cada versión del algoritmo con cada manejador de restricciones. El éxito del algoritmo fue probado en el conjunto de 24 funciones de prueba del CEC2006 (ver Apéndice A). Las conclusiones de este experimento se mencionan a continuación.

- Tanto para el algoritmo BSO como para las variantes utilizadas en este trabajo, especialmente la versión MBSO, se requiere de manejadores de restricciones como ε-constrained donde se maneje una tolerancia que permite soluciones ligeramente no factibles como factibles, esto hace que esta clase de algoritmos pueda obtener los mejores resultados en el tipo de problemas utilizados.

Una vez comprobado el éxito del algoritmo BSO en espacios restringidos, el siguiente paso fue tratar de mejorar su desempeño guiando su búsqueda hacia la frontera de la región factible a través del operador especial Boundary Search (BS) y dos nuevos operadores de frontera propuestos en este trabajo. El primero de ellos (BS-r) basado en la división de segmento de línea de Álgebra Lineal, en el cual se busca alcanzar la frontera de las restricciones a través de tamaños de paso variable definidos por los valores del parámetro r los cuales van desde 0.1 hasta 0.9. El segundo operador propuesto (BS-G) intenta cubrir las limitantes observadas en el operador BS. BS-G no requiere de puntos factibles para iniciar el proceso de búsqueda sino que

busca explorar la frontera de las restricciones a través del gradiente de la restricción violada. A partir del cálculo de vectores de factibilidad se intentó acercar un punto no factible a un punto que se encuentre dentro o muy cerca de la frontera de la restricción.

Dado que esta clase de operadores necesita definir una política para decidir a qué restricción se aplicará el método, cuando existe más de una restricción asociada al problema, se decidió explorar la frontera de la restricción más difícil de satisfacer en el punto actual utilizando el conjunto de treinta y seis funciones de prueba del CEC2010 (ver Apéndice B. Los resultados de este trabajo se presentan en el Capítulo 5 y se puede concluir lo siguiente.

- Si el tipo de problema lo permite, utilizar un operador de frontera que haga uso del gradiente además de mejorar los resultados finales del algoritmo MBSO, se elimina el requerimiento de puntos factibles de entrada, lo cual es una gran ventaja en espacios restringidos.

Otra área de interés en espacios restringidos es el aproximarse a la región factible. El objetivo de los métodos de búsqueda en este tipo de problemas es acercarse lo más rápido posible a esta región de interés y poder mantenerse dentro de ella. Para lograr la meta, en esta tesis se propuso explorar el método de consenso de restricciones basado en la proyección del gradiente de las restricciones. Se propuso además, una variante al método conocida como R+V, la cual ahorra costo computacional al calcular un único vector de factibilidad y por lo tanto, un único gradiente para la restricción con mayor grado de violación. La propuesta se comparó con las variantes existentes en el conjunto de treinta y seis funciones de prueba del CEC2010 (ver Apéndice B). Una vez que se demostró su éxito en este tipo de problemas, se combinó con el algoritmo MBSO y los mejores resultados se compararon con los algoritmos del estado del arte. Los resultados de este trabajo se pueden apreciar en el Capítulo 6 y del experimento se aporta el siguiente conocimiento:

- Si bien se pueden obtener resultados competitivos utilizando las versiones existentes del método de consenso de restricciones, en este trabajo se demostró que con la variante R+V se puede simplificar el proceso para obtener resultados muy similares utilizando información de una sola restricción, sin necesidad de considerar todas las restricciones violadas en el punto actual.

Observaciones generales respecto al algoritmo BSO:

- El algoritmo BSO mostró la capacidad de generar resultados finales muy similares a los obtenidos por algoritmos representativos del estado del arte en problemas de optimización numérica con restricciones. Los conjuntos de prueba a los que se sometió el algoritmo son los presentados en el CEC2006 y CEC2010 (ver Apéndices A y B).

- La mayor dificultad en el algoritmo BSO es la cantidad de parámetros a calibrar. Aunque en esta tesis los valores de los parámetros utilizados son los propuestos por los autores, es evidente que los resultados mostrados pueden variar si estos valores de parámetros son reconfigurados.

- El desempeño general del algoritmo BSO en espacios restringidos es mejorado con el uso de operadores especiales.

Observaciones generales respecto a los operadores de frontera BS, BS-r y BS-G:

- En los tres operadores de frontera estudiados, aunque la idea de aplicarlo a la restricción con mayor grado de violación resultó conveniente, existe la desventaja de no tomar en cuenta la información que pudieran presentar otras restricciones que son violadas en el punto actual. Ello provoca que el método esté continuamente generando puntos que exploran sólo un límite de la región factible dejando de lado áreas que pudieran ser prometedoras.

- Tanto en el operador BS como en el operador BS-r la necesidad de puntos factibles constituyen una desventaja cuando se trata con problemas difíciles de resolver. Por ejemplo, espacios de búsqueda muy grandes con regiones factibles muy pequeñas.

- La simplicidad de la búsqueda ofrecida por los métodos BS y BS-r para alcanzar la frontera de las restricciones junto con los resultados ofrecidos son su principal fortaleza. Además de poder ser aplicados a la gran mayoría de problemas sin importar sus características, siempre y cuando en este tipo de problemas exista la posibilidad de generar puntos factibles.

- En el operador BS-G, el no requerir puntos factibles de entrada se convierte en una ventaja para el método que le permite ser aplicado en generaciones tempranas dentro del ciclo de evolución. Ello permite al algoritmo de búsqueda global alcanzar con mayor rapidez la región factible.

- La mayor desventaja del método BS-G es que al ser un método basado en el gradiente necesita de espacios continuos y funciones diferenciales. Si éstas condiciones no se cumplen, el método presenta dificultades de aplicación y si no se dan las condiciones de paro apropiadas, el método sólo utilizaría evaluaciones innecesarias en el proceso.

Observaciones generales respecto a la variante R+V:

- R+V es una variante del método de consenso de restricciones, por lo tanto su objetivo principal es alcanzar la región factible con rapidez. Lo que la distingue de otras variantes del mismo método es que la búsqueda de la región factible lo hace a partir de la restricción con mayor grado de violación en lugar de tomar consenso de todas las restricciones violadas en el punto actual. Esta diferencia respecto de las otras variantes permite un ahorro computacional al momento de calcular el gradiente sólo para una restricción y no por cada restricción violada. Aunque esta es una ventaja para la variante, la desventaja es que el método aproxima la región factible sólo considerando información parcial de las restricciones.

- La variante R+V muestra un desempeño similar al de las otras variantes basadas en el mismo método. Esto la convierte en una opción atractiva en el área ya que se pueden obtener resultados similares a un bajo costo computacional.

- Al igual que sus antecesoras, la variante R+V padece las mismas carencias al tratarse de un método basado en el gradiente y enfrentarse a espacios discontinuos y funciones no diferenciales.

Los resultados muestran que se validó la hipótesis propuesta y se cumplió con los objetivos especificados en el Capítulo 1. Todo el trabajo presentado permitió obtener una versión del algoritmo BSO, reforzada con operadores especiales, competitiva en espacios restringidos y contra algoritmos del estados del arte. Además, de contribuir a la literatura con dos operadores especiales de frontera y un operador de factibilidad que pueden ser utilizados en distintos problemas de prueba y por cualquier algoritmo sin cambios adicionales.

7.2. Trabajo Futuro

Como posible continuación del trabajo presentado en esta tesis se encuentra los siguientes:

- Enfrentar el algoritmo MBSO reforzado con operadores especiales (MBSO-BS, MBSO-R+V) al conjunto de funciones propuesto en el CEC2017.

- Aplicar los operadores especiales propuestos, tanto de frontera como de factibilidad, a otros algoritmos del estado del arte como lo es la Evolución diferencial y PSO.

- Mejorar el operador de frontera BS-r de tal manera que el parámetro r tome valores adaptativos de acuerdo al grado de violación de las restricciones.

- Mejorar el operador de frontera BS de tal manera que pueda integrar información de todas las restricciones violadas al mismo tiempo antes de generar el nuevo punto. Esto con la finalidad de generar puntos cerca de la frontera de todas las restricciones a la vez y poder rodear la frontera de la región factible en el espacio de búsqueda.
- Combinar el uso de los dos operadores (BS-G y R+V) en el mismo algoritmo MBSO usando un mecanismo de aprendizaje para su correcta aplicación.

Parte I

Apéndices

Apéndice A

Funciones de Prueba CEC 2006

g01

Minimizar:

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

sujeto a

$$\begin{aligned}
g_1(\vec{x}) &= 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0 \\
g_2(\vec{x}) &= 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0 \\
g_3(\vec{x}) &= 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0 \\
g_4(\vec{x}) &= -8x_1 + x_{10} \leq 0 \\
g_5(\vec{x}) &= -8x_2 + x_{11} \leq 0 \\
g_6(\vec{x}) &= -8x_3 + x_{12} \leq 0 \\
g_7(\vec{x}) &= -2x_4 - x_5 + x_{10} \leq 0 \\
g_8(\vec{x}) &= -2x_6 - x_7 + x_{11} \leq 0 \\
g_9(\vec{x}) &= -2x_8 - x_9 + x_{12} \leq 0
\end{aligned}$$

Donde los límites son $0 \leq x_i \leq 1 (i = 1, \ldots, 9), 0 \leq x_i \leq 100 (i = 10, 11, 12)y0 \leq x13 \leq 1$

g02

Minimizar:

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

Tabla A.1: Detalle de las 24 funciones de prueba. **D** dimensión del problema; **LI** restricciones de desigualdad lineal; **NI** restricciones de desigualdad no lineal; **LE** restricciones de igualdad lineal; **NE** restricciones de igualdad no lineal. ρ radio estimado entre la region factible y todo el espacio de bÁºsqueda

Tipo	D	Función	ρ	LI	NI	LE	NE
g01	13	Cuadrática	0.0111 %	9	0	0	0
g02	20	No lineal	99.9971 %	0	2	0	0
g03	10	Polinomial	0.0000 %	0	0	0	1
g04	5	Cuadrática	52.1230 %	0	6	0	0
g05	4	cubica	0.0000 %	2	0	0	3
g06	2	cubica	0.0066 %	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 %	3	3	0	0
g11	2	Cuadrática	0.0000 %	0	0	0	1
g12	3	Cuadrática	4.7713 %	0	1	0	0
g13	5	No lineal	0.0000 %	0	0	0	3
g14	10	nonlinear	0.0000 %	0	0	3	0
g15	3	Cuadrática	0.0000 %	0	0	1	1
g16	5	No lineal	0.0204 %	4	34	0	0
g17	6	No lineal	0.0000 %	0	0	0	4
g18	9	Cuadrática	0.0000 %	0	13	0	0
g19	15	No lineal	33.4761 %	0	5	0	0
g20	24	Lineal	0.0000 %	0	6	2	12
g21	7	Lineal	0.0000 %	0	1	0	5
g22	22	Lineal	0.0000 %	0	1	8	11
g23	9	Lineal	0.0000 %	0	2	3	1
g24	2	Lineal	79.6556 %	0	2	0	0

Sujeto a:

$$g_1(\vec{x}) = 0.75 - \prod_{i=1}^{n} x_i \leq 0$$

$$g_2(\vec{x}) = \sum_{i=1}^{n} x_i - 7.5n \leq 0$$

donde $n = 20$ y $0 < x_i \leq 10 (i = 1, \ldots, n)$

g03

Minimizar:

$$f(\vec{x}) = -(\sqrt{n})^n \prod_{i=1}^{n} x_i$$

Sujeto a:

$$h(\vec{x}) = \sum_{i=1}^{n} x_i^2 - 1 = 0$$

donde $n = 10$ y $0 \leq x_i \leq 1 (i = 1, \ldots, n)$.

g04

Minimizar:

$$f(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$

Sujeto a:

$$g_1(\vec{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0$$
$$g_2(\vec{x}) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0$$
$$g_3(\vec{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0$$
$$g_4(\vec{x}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0$$
$$g_5(\vec{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0$$
$$g_6(\vec{x}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0$$

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

g05

Minimizar:

$$f(\vec{x}) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3$$

Sujeto a:

$$g_1(\vec{x}) = -x_4 + x_3 - 0.55 \leq 0$$
$$g_2(\vec{x}) = -x_3 + x_4 - 0.55 \leq 0$$
$$h_3(\vec{x}) = 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0$$
$$h_4(\vec{x}) = 1000\sin(x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0$$
$$h_5(\vec{x}) = 1000\sin(x_4 - 0.25) + 1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0$$

donde $0 \leq x_1 \leq 1200$, $0 \leq x_2 \leq 1200$, $-0{,}55 \leq x_3 \leq 0{,}55$ y $-0{,}55 \leq x_4 \leq 0{,}55$.

g06

Minimizar:

$$f(\vec{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$$

Sujeto a:

$$g_1(\vec{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$$
$$g_2(\vec{x}) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0$$

donde $13 \leq x_1 \leq 100$ y $0 \leq x_2 \leq 100$

g07

Minimizar:

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

Sujeto a:

$$g_1(\vec{x}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0$$
$$g_2(\vec{x}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0$$
$$g_3(\vec{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0$$
$$g_4(\vec{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0$$
$$g_5(\vec{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0$$
$$g_6(\vec{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1 x_2 + 14x_5 - 6x_6 \leq 0$$
$$g_7(\vec{x}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0$$
$$g_8(\vec{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0$$

donde $-10 \leq x_i \leq 10 (i = 1, \ldots, 10)$

g08

Minimizar:

$$f(\vec{x}) = -\frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$$

Sujeto a:

$$g_1(\vec{x}) = x_1^2 - x_2 + 1 \leq 0$$
$$g_2(\vec{x}) = 1 - x_1 + (x_2 - 4)^2 \leq 0$$

donde $0 \leq x_1 \leq 10$ y $0 \leq x_2 \leq 10$

g09

Minimizar:

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

Sujeto a:

$$g_1(\vec{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0$$
$$g_2(\vec{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0$$
$$g_3(\vec{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0$$
$$g_4(\vec{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0$$

donde $-10 \leq x_i \leq 10 for (i = 1, \ldots, 7)$

g10

Minimizar:

$$f(\vec{x}) = x_1 + x_2 + x_3$$

Sujeto a:

$$g_1(\vec{x}) = -1 + 0.0025(x_4 + x_6) \leq 0$$
$$g_2(\vec{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \leq 0$$
$$g_3(\vec{x}) = -1 + 0.01(x_8 - x_5) \leq 0$$
$$g_4(\vec{x}) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leq 0$$
$$g_5(\vec{x}) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0$$
$$g_6(\vec{x}) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0$$

donde $100 \leq x_1 \leq 10000$, $1000 \leq x_i \leq 10000 (i = 2, 3)$ y $10 \leq x_i \leq 1000 (i = 4, \ldots, 8)$

g11

Minimizar:

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

Sujeto a:

$$h(\vec{x}) = x_2 - x_1^2 = 0$$

donde $-1 \leq x_1 \leq 1$ y $-1 \leq x_2 \leq 1$

g12

Minimizar:

$$f(\vec{x}) = -(100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2)/100$$

Sujeto a:

$$g(\vec{x}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leq 0$$

donde $0 \leq x_i \leq 10(i = 1, 2, 3)$ y $p, q, r = 1, 2, \ldots, 9$

g13

Minimizar:

$$f(\vec{x}) = e^{x_1 x_2 x_3 x_4 x_5}$$

Sujeto a:

$$h_1(\vec{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$$
$$h_2(\vec{x}) = x_2 x_3 - 5x_4 x_5 = 0$$
$$h_3(\vec{x}) = x_1^3 + x_2^3 + 1 = 0$$

donde $-2{,}3 \leq x_i \leq 2{,}3(i = 1, 2)$ y $-3{,}2 \leq x_i \leq 3{,}2(i = 3, 4, 5)$

g14

Minimizar:

$$f(\vec{x}) = \sum_{i=1}^{10} x_i \left(c_i + ln \frac{x_i}{\sum_{j=1}^{10} x_j} \right)$$

Sujeto a:

$$h_1(\vec{x}) = x_1 + 2x_2 + 2x_3 + x_6 + x_{10} - 2 = 0$$
$$h_2(\vec{x}) = x_4 + 2x_5 + x_6 + x_7 - 1 = 0$$
$$h_3(\vec{x}) = x_3 + x_7 + x_8 + 2x_9 + x_{10} = 0$$

Donde los límites son $0 < x_i \leq 10(i = 1, \ldots, 10)$, y $c_1 = -6{,}089$, $c_2 = -17{,}164$, $c_3 = -34{,}054$, $c_4 = -5{,}914$, $c_5 = -24{,}721$, $c_6 = -14{,}986$, $c_7 = -24{,}1$, $c_8 = -10{,}708$, $c_9 = -26{,}662$, $c_{10} = -22{,}179$

g15

Minimizar:

$$f(\vec{x}) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1 x_2 - x_1 x_3$$

Sujeto a:

$$h_1(\vec{x}) = x_1^2 + x_2^2 + x_3^2 - 25 = 0$$
$$h_2(\vec{x}) = 8x_1 + 14x_2 + 7x_3 - 56 = 0$$

Donde los límites son $0 \leq x_i \leq 10(i = 1, 2, 3)$

g16

Minimizar:

$$f(\vec{x}) = 0{,}000117y_{14} + 0{,}1365 + 0{,}0002358y_{13} + 0{,}000001502y_{16} + 0{,}0321y_{12} + 0{,}004324y_5 + 0{,}0001\frac{c_{15}}{c_{16}} + 37{,}48\frac{y_2}{c_{12}} - 0{,}0000005843y_{17}$$

Sujeto a:

$$
\begin{aligned}
g_1(\vec{x}) &= \frac{0{,}28}{0{,}72} y_5 - y_4 \leq 0 \\
g_2(\vec{x}) &= x_3 - 1{,}5 x_2 \leq 0 \\
g_3(\vec{x}) &= 3496 \frac{y_2}{c_{12}} - 21 \leq 0 \\
g_4(\vec{x}) &= 110{,}6 + y_1 - \frac{62212}{c_{17}} \leq 0 \\
g_5(\vec{x}) &= 213{,}1 - y_1 \leq 0 \\
g_6(\vec{x}) &= y_1 - 405{,}23 \leq 0 \\
g_7(\vec{x}) &= 17{,}505 - y_2 \leq 0 \\
g_8(\vec{x}) &= y_2 - 1053{,}6667 \leq 0 \\
g_9(\vec{x}) &= 11{,}275 - y_3 \leq 0 \\
g_{10}(\vec{x}) &= y_3 - 35{,}03 \leq 0 \\
g_{11}(\vec{x}) &= 214{,}228 - y_4 \leq 0 \\
g_{12}(\vec{x}) &= y_4 - 665{,}585 \leq 0 \\
g_{13}(\vec{x}) &= 7{,}458 - y_5 \leq 0 \\
g_{14}(\vec{x}) &= y_5 - 584{,}463 \leq 0 \\
g_{15}(\vec{x}) &= 0{,}961 - y_6 \leq 0 \\
g_{16}(\vec{x}) &= y_6 - 265{,}916 \leq 0 \\
g_{17}(\vec{x}) &= 1{,}612 - y_7 \leq 0 \\
g_{18}(\vec{x}) &= y_7 - 7{,}046 \leq 0 \\
g_{19}(\vec{x}) &= 0{,}146 - y_8 \leq 0 \\
g_{20}(\vec{x}) &= y_8 - 0{,}222 \leq 0 \\
g_{21}(\vec{x}) &= 107{,}99 - y_9 \leq 0 \\
g_{22}(\vec{x}) &= y_9 - 273{,}366 \leq 0 \\
g_{23}(\vec{x}) &= 922{,}693 - y_{10} \leq 0 \\
g_{24}(\vec{x}) &= y_{10} - 1286{,}105 \leq 0 \\
g_{25}(\vec{x}) &= 926{,}832 - y_{11} \leq 0 \\
g_{26}(\vec{x}) &= y_{11} - 1444{,}046 \leq 0 \\
g_{27}(\vec{x}) &= 18{,}766 - y_{12} \leq 0 \\
g_{28}(\vec{x}) &= y_{12} - 537{,}141 \leq 0 \\
g_{29}(\vec{x}) &= 1072{,}163 - y_{13} \leq 0 \\
g_{30}(\vec{x}) &= y_{13} - 3247{,}039 \leq 0 \\
g_{31}(\vec{x}) &= 8961{,}448 - y_{14} \leq 0 \\
g_{32}(\vec{x}) &= y_{14} - 26844{,}086 \leq 0 \\
g_{33}(\vec{x}) &= 0{,}063 - y_{15} \leq 0 \\
g_{34}(\vec{x}) &= y_{15} - 0{,}386 \leq 0 \\
g_{35}(\vec{x}) &= 71084{,}33 - y_{16} \leq 0 \\
g_{36}(\vec{x}) &= -140000 + y_{16} \leq 0 \\
g_{37}(\vec{x}) &= 2802713 - y_{17} \leq 0 \\
g_{38}(\vec{x}) &= y_{17} - 12146108 \leq 0
\end{aligned}
$$

donde:

$$y_1 = x_2 + x_3 + 41{,}6$$
$$c_1 = 0{,}024x_4 - 4{,}62$$
$$y_2 = \frac{12{,}5}{c_1} + 12$$
$$c_2 = 0{,}0003535x_1^2 + 0{,}5311x_1 + 0{,}08705y_2x_1$$
$$c_3 = 0{,}052x_1 + 78 + 0{,}002377y_2x_1$$
$$y_3 = \frac{c_2}{c_3}$$
$$y_4 = 19y_3$$
$$c_4 = 0{,}04782(x_1 - y_3) + \frac{0{,}1956(x_1 - y_3)^2}{x_2} + 0{,}6376y_4 + 1{,}594y_3$$
$$c_5 = 100x_2$$
$$c_6 = x_1 - y_3 - y_4$$
$$c_7 = 0{,}950 - \frac{c_4}{c_5}$$
$$y_5 = c_6c_7$$
$$y_6 = x_1 - y_5 - y_4 - y_3$$
$$c_8 = (y_5 + y_4)0{,}995$$
$$y_7 = \frac{c_8}{y_1}$$
$$y_8 = \frac{c_8}{3798}$$
$$c_9 = y_7 - \frac{0{,}0663y_7}{y_8} - 0{,}3153$$
$$y_9 = \frac{96{,}82}{c_9} + 0{,}321y_1$$
$$y_{10} = 1{,}29y_5 + 1{,}258y_4 + 2{,}29y_3 + 1{,}71y_6$$
$$y_{11} = 1{,}71x_1 - 0{,}452y_4 + 0{,}580y_3$$
$$c_{10} = \frac{12{,}3}{752{,}3}$$
$$c_{11} = (1{,}75y_2)(0{,}995x_1)$$
$$c_{12} = 0{,}995y_{10} + 1998$$
$$y_{12} = c_{10}x_1 + \frac{c_{11}}{c_{12}}$$
$$y_{13} = c_{12} - 1{,}75y_2$$
$$y_{14} = 3623 + 64{,}4x_2 + 58{,}4x_3 + \frac{146312}{y_9 + x_5}$$
$$c_{13} = 0{,}995y_{10} + 60{,}8x_2 + 48x_4 - 0{,}1121y_{14} - 5095$$
$$y_{15} = \frac{y_{13}}{c_{13}}$$
$$y_{16} = 148000 - 331000y_{15} + 40y_{13} - 61y_{15}y_{13}$$
$$c_{14} = 2324y_{10} - 28740000y_2$$
$$y_{17} = 14130000 - 1328y_{10} - 531y_{11} + \frac{c_{14}}{c_{12}}$$
$$c_{15} = \frac{y_{13}}{y_{15}} - \frac{y_13}{0{,}52}$$
$$c_{16} = 1{,}104 - 0{,}72y_{15}$$

y Donde los límites son $704{,}4148 \leq x_1 \leq 906{,}3855$, $68{,}6 \leq x_2 \leq 288{,}88$, $0 \leq x_3 \leq 134{,}75$, $193 \leq x_4 \leq 287{,}0966$ y $25 \leq x_5 \leq 84{,}1988$.

g17

Minimizar:

$$f(\vec{x}) = f(x_1) + f(x_2)$$

donde:

$$f_1(x_1) = \begin{cases} 30x_1 & 0 \leq x_1 < 300 \\ 31x_1 & 300 \leq x_1 < 400 \end{cases}$$

$$f_2(x_2) = \begin{cases} 28x_2 & 0 \leq x_2 < 100 \\ 29x_2 & 100 \leq x_2 < 200 \\ 30x_2 & 200 \leq x_2 < 1000 \end{cases}$$

Sujeto a:

$$h_1(\vec{x}) = -x_1 + 300 - \frac{x_3 x_4}{131{,}078} cos(1{,}48477 - x_6) + \frac{0{,}90798 x_3^2}{131{,}078} cos(1{,}47588)$$

$$h_2(\vec{x}) = -x_2 - \frac{x_3 x_4}{131{,}078} cos(1{,}48477 + x_6) + \frac{0{,}90798 x_4^2}{131{,}078} cos(1{,}47588)$$

$$h_3(\vec{x}) = -x_5 - \frac{x_3 x_4}{131{,}078} sin(1{,}48477 + x_6) + \frac{0{,}90798 x_4^2}{131{,}078} sin(1{,}47588)$$

$$h_4(\vec{x}) = 200 - \frac{x_3 x_4}{131{,}078} sin(1{,}48477 - x_6) + \frac{0{,}90798 x_3^2}{131{,}078} sin(1{,}47588)$$

Donde los límites son $0 \leq x_1 \leq 400$, $0 \leq x_2 \leq 1000$, $340 \leq x_3 \leq 420$, $340 \leq x_4 \leq 420$, $-1000 \leq x_5 \leq 1000$ y $0 \leq x_6 \leq 0{,}5236$

g18

Minimizar:

$$f(\vec{x}) = -0{,}5(x_1 x_4 - x_2 x_3 + x_3 x_9 - x_5 x_9 + x_5 x_8 - x_6 x_7)$$

Sujeto a:

$$\begin{aligned}
g_1(\vec{x}) &= x_3^2 + x_4^2 - 1 \leq 0 \\
g_2(\vec{x}) &= x_9^2 - 1 \leq 0 \\
g_3(\vec{x}) &= x_5^2 + x_6^2 - 1 \leq 0 \\
g_4(\vec{x}) &= x_1^2 + (x_2 - x_9)^2 - 1 \leq 0 \\
g_5(\vec{x}) &= (x_1 - x_5)^2 + (x_2 - x_6)^2 - 1 \leq 0 \\
g_6(\vec{x}) &= (x_1 - x_7)^2 + (x_2 - x_8)^2 - 1 \leq 0 \\
g_7(\vec{x}) &= (x_3 - x_5)^2 + (x_2 - x_6)^2 - 1 \leq 0 \\
g_8(\vec{x}) &= (x_3 - x_7)^2 + (x_2 - x_8)^2 - 1 \leq 0 \\
g_9(\vec{x}) &= x_7^2 + (x_8 - x_9)^2 - 1 \leq 0 \\
g_{10}(\vec{x}) &= x_2 x_3 - x_1 x_4 \leq 0 \\
g_{11}(\vec{x}) &= -x_3 x_9 \leq 0 \\
g_{12}(\vec{x}) &= x_5 x_9 \leq 0 \\
g_{13}(\vec{x}) &= x_6 x_7 - x_5 x_8 \leq 0
\end{aligned}$$

Donde los límites son $-10 \leq x_i \leq 10 (i = 1, \ldots, 8)$ y $0 \leq x_9 \leq 20$

g19

Minimizar:

$$f(\vec{x}) = \sum_{j=1}^{5} \sum_{i=1}^{5} c_{i,j} x_{(10+j)} + 2 \sum_{j=1}^{5} d_j x_{(10+j)}^3 - \sum_{i=1}^{10} b_i x_i$$

Sujeto a:

$$g(\vec{x}) = -2 \sum_{i=1}^{5} c_{i,j} x_{(10+j)} - e_j + \sum_{i=1}^{10} a_{i,j} x_i \leq 0 \quad j = 1, \ldots, 5$$

donde $b = [-40, -2, -0{,}25, -4, -4, -1, -40, -60, 5, 1]$. los límites are $0 \leq x_i \leq 10 (i = 1, \ldots, 15)$

g20

Minimizar:

$$f(\vec{x}) = \sum_{i=1}^{24} a_i x_i$$

Sujeto a:

$$
\begin{aligned}
g_i(\vec{x}) &= \frac{(x_i + x_{(i+12)})}{\sum_{j=1}^{24} x_j + e_i} \leq 0 \quad i = 1, 2, 3 \\
g_i(\vec{x}) &= \frac{(x_{(i+3)} + x_{(i+15)})}{\sum_{j=1}^{24} x_j + e_i} \leq 0 \quad i = 4, 5, 6 \\
h_i(\vec{x}) &= \frac{x_{(i+12)}}{b_{(i+12)} \sum_{j=13}^{24} \frac{x_j}{b_j}} - \frac{c_i x_i}{40 b_i \sum_{j=1}^{12} \frac{x_j}{b_j}} = 0 \quad i = 1, \ldots, 12 \\
h_{13}(\vec{x}) &= \sum_{i=1}^{24} x_i - 1 = 0 \\
h_{14}(\vec{x}) &= \sum_{i=1}^{12} \frac{x_i}{d_i} + k \sum_{i=13}^{24} \frac{x_i}{b_i} - 1{,}671 = 0
\end{aligned}
$$

donde $k = (0{,}7302)(530)(\frac{14{,}7}{40})$. los límites are $0 \leq x_i \leq 10 (i = 1, \ldots, 24)$
g21

Minimizar:

$$f(\vec{x}) = x_1$$

Sujeto a:

$$
\begin{aligned}
& g_1(\vec{x}) = -x_1 + 35 x_2^{0{,}62} + 35 x_3^{0{,}6} \leq 0 \\
& h_1(\vec{x}) = -300 x_3 + 7500 x_5 - 7500 x_6 - 25 x_4 x_5 + 25 x_4 x_6 + x_3 x_4 = 0 \\
& h_2(\vec{x}) = 100 x_2 + 155{,}365 x_4 + 2500 x_7 - x_2 x_4 - 25 x_4 x_7 - 15536{,}5 = 0 \\
& h_3(\vec{x}) = -x_5 + ln(-x_4 + 900) = 0 \\
& h_4(\vec{x}) = -x_6 + ln(x_4 + 300) = 0 \\
& h_5(\vec{x}) = -x_7 + ln(-2 x_4 + 700) = 0
\end{aligned}
$$

Donde los límites son $0 \leq x_1 \leq 1000$, $0 \leq x_2$, $x_3 \leq 40$, $100 \leq x_4 \leq 300$, $6{,}3 \leq x_5 \leq 6{,}7$, $5{,}9 \leq x_6 \leq 6{,}4$ y $4{,}5 \leq x_7 \leq 6{,}25$
g22

Minimizar:

$$f(\vec{x}) = x_1$$

Sujeto a:

$$g_1(\vec{x}) = -x_1 + x_2^{0,62} + x_3^{0,6} + x_4^{0,6} \leq 0$$
$$h_1(\vec{x}) = x_5 - 100000x_8 + 1 \times 10^7 = 0$$
$$h_2(\vec{x}) = x_6 + 100000x_8 - 100000x9 = 0$$
$$h_3(\vec{x}) = x_7 + 100000x_9 - 5 \times 10^7 = 0$$
$$h_4(\vec{x}) = x_5 + 100000x_{10} - 3{,}3 \times 10^7 = 0$$
$$h_5(\vec{x}) = x_6 + 100000x_{11} - 4{,}4 \times 10^7 = 0$$
$$h_6(\vec{x}) = x_7 + 100000x_{12} - 6{,}6 \times 10^7 = 0$$
$$h_7(\vec{x}) = x_5 - 120x_2x_{13} = 0$$
$$h_8(\vec{x}) = x_6 - 80x_3x_{14} = 0$$
$$h_9(\vec{x}) = x_7 - 40x_4x_{15} = 0$$
$$h_{10}(\vec{x}) = x_8 - x_{11} + x_{16} = 0$$
$$h_{11}(\vec{x}) = x_9 - x_{12} + x_{17} = 0$$
$$h_{12}(\vec{x}) = -x_{18} + ln(x_{10} - 100) = 0$$
$$h_{13}(\vec{x}) = -x_{19} + ln(-x_8 + 300) = 0$$
$$h_{14}(\vec{x}) = -x_{20} + ln(x_{16}) = 0$$
$$h_{15}(\vec{x}) = -x_{21} + ln(-x_9 + 400) = 0$$
$$h_{16}(\vec{x}) = -x_{22} + ln(x_{17}) = 0$$
$$h_{17}(\vec{x}) = -x_8 - x_{10} + x_{13}x_{18} - x_{13}x_{19} + 400 = 0$$
$$h_{18}(\vec{x}) = x_8 - x_9 - x_{11} + x_{14}x_{20} - x_{14}x_{21} + 400 = 0$$
$$h_{19}(\vec{x}) = x_9 - x_{12} - 4{,}60517x_{15} + x_{15}x_{22} + 100 = 0$$

Donde los límites son $0 \leq x_1 \leq 20000$, $0 \leq x_2, x_3, x_4 \leq 1 \times 10^6$, $0 \leq x_5, x_6, x_7 \leq 4 \times 10^7$, $100 \leq x_8 \leq 299{,}99$, $100 \leq x_9 \leq 399{,}99$, $100{,}01 \leq x_{10} \leq 300$, $100 \leq x_{11} \leq 400$, $100 \leq x_{12} \leq 600$, $0 \leq x_{13}, x_{14}, x_{15} \leq 500$, $0{,}01 \leq x_{16} \leq 300$, $0{,}01 \leq x_{17} \leq 400$, $-4{,}7 \leq x_{18}, x_{19}, x_{20}, x_{21}, x_{22} \leq 6{,}25$

g23

Minimizar:

$$f(\vec{x}) = -9x_5 - 15x_8 + 6x_1 + 16x_2 + 10(x_6 + x_7)$$

Sujeto a:

$$g_1(\vec{x}) = x_9x_3 + 0{,}02x_6 - 0{,}025x_5 \leq 0$$
$$g_2(\vec{x}) = x_9x_4 + 0{,}02x_7 - 0{,}015x_8 \leq 0$$
$$h_1(\vec{x}) = x_1 + x_2 - x_3 - x_4 = 0$$
$$h_2(\vec{x}) = 0{,}03x_1 + 0{,}01x_2 - x_9(x_3 + x_4) = 0$$
$$h_3(\vec{x}) = x_3 + x_6 - x_5 = 0$$
$$h_4(\vec{x}) = x_4 + x_7 - x_8 = 0$$

Donde los límites son $0 \leq x_1, x_2, x_6 \leq 300$, $0 \leq x_3, x_5, x_7 \leq 100$, $0 \leq x_4, x_8 \leq 200$ y $0{,}01 \leq x_9 \leq 0{,}03$

g24

Minimizar:

$$f(\vec{x}) = -x_1 - x_2$$

Sujeto a:

$$g_1(\vec{x}) = -2x_1^4 + 8x_1^3 - 8x_1^2 + x_2 - 2 \leq 0$$
$$g_2(\vec{x}) = -4x_1^4 + 32x_1^3 - 88x_1^2 + 96x_1 + x_2 - 36 \leq 0$$

donde los límites $are 0 \leq x_1 \leq 3$ y $0 \leq x_2 \leq 4$

Apéndice B

Funciones de Prueba CEC 2010

Tabla B.1: Detalles de las 18 funciones de prueba. D número de variables. S, NS and R son Separable, No Separable and Rotada, respectivamente.

Problema	Rango	Tipo de objetivo	Nnúmero de restricciones	
			Igualdad	Desigualdad
C01	$[0, 10]^D$	No Separable	-	2-NS
C02	$[-5.12, 5.12]^D$	Separable	1-S	2-S
C03	$[-1000, 1000]^D$	No Separable	1-NS	-
C04	$[-50, 50]^D$	Separable	2-S / 2-NS	-
C05	$[-600, 600]^D$	Separable	2-S	-
C06	$[-600, 600]^D$	Separable	2-R	-
C07	$[-140, 140]^D$	No Separable	-	1-S
C08	$[-140, 140]^D$	No Separable	-	1-R
C09	$[-500, 500]^D$	No Separable	1-S	-
C10	$[-500, 500]^D$	No Separable	1-R	-
C11	$[-100, 100]^D$	Rotada	1-NS	-
C12	$[-1000, 1000]^D$	Separable	1-NS	1-S
C13	$[-500, 500]^D$	Separable	-	2-S / 1-NS
C14	$[-1000, 1000]^D$	No Separable	-	3-S
C15	$[-1000, 1000]^D$	No Separable	-	3-R
C16	$[-10, 10]^D$	No Separable	2-S	1-S / 1-NS
C17	$[-10, 10]^D$	No Separable	1-S	2-NS
C18	$[-50, 50]^D$	No Separable	1-S	1-S

C01

Minimize:

$$f(x) = -\left|\sum_{i=1}^{D} cos^4(z_i) - 2\prod_{i=1}^{D} cos^2(z_i)\right|$$
$$z = x - o$$

sujeto a:

$$g_1(x) = 0{,}75 - \prod_{i=1}^{D} z_i \leq 0$$
$$g_2(x) = \sum_{i=1}^{D} z_i - 7{,}5D \leq 0$$
$$x \in [0, 10]^D$$

C02

Minimizar:

$$f(x) = max(z)$$
$$z = x - o$$
$$y = z - 0{,}5$$

sujeto a:

$$g_1(x) = 10 - \frac{1}{D}\sum_{i=1}^{D}\left[z_i^2 - 10cos(2\Pi z_i) + 10\right] \leq 0$$
$$g_2(x) = \frac{1}{D}\sum_{i=1}^{D}\left[z_i^2 - 10cos(2\Pi z_i) + 10\right] - 15 \leq 0$$
$$h(x) = \frac{1}{D}\sum_{i=1}^{D}\left[y_i^2 - 10cos(2\Pi y_i) + 10\right] - 20 = 0$$
$$x \in [-5{,}12, 5{,}12]^D$$

C03

Minimizar:

$$f(x) = \sum_{i=1}^{D-1}(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)$$
$$z = x - o$$

sujeto a:

$$h(x) = \sum_{i=1}^{D-1}(z_i - z_{i+1})^2 = 0$$
$$x \in [-1000, 1000]^D$$

C04

Minimizar:

$$f(x) = max(z)$$
$$z = x - o$$

sujeto a:

$$h_1(x) = \frac{1}{D}\sum_{i=1}^{D}(z_i cos(\sqrt{|z_i|})) = 0$$
$$h_2(x) = \sum_{i=1}^{D/2-1}(z_i - z_{i+1})^2 = 0$$
$$h_3(x) = \sum_{i=1}^{D-1}(z_i^2 - z_{i+1})^2 = 0$$
$$h_4(x) = \sum_{i=1}^{D} z = 0$$
$$x \in [-50, 50]^D$$

C05

Minimizar:

$$f(x) = max(z)$$
$$z = x - o$$

sujeto a:

$$h_1(x) = \frac{1}{D}\sum_{i=1}^{D}(-z_i sin(\sqrt{|z_i|})) = 0$$
$$h_2(x) = \frac{1}{D}\sum_{i=1}^{D}(-z_i cos(0{,}5\sqrt{|z_i|})) = 0$$
$$x \in [-600, 600]^D$$

C06

Minimizar:

$$f(x) = max(z)$$
$$z = x - o$$
$$y = (x + 483{,}6106156535 - o)M - 483{,}6106156535$$

sujeto a:

$$h_1(x) = \frac{1}{D}\sum_{i=1}^{D}(-y_i sin(\sqrt{|y_i|})) = 0$$
$$h_2(x) = \frac{1}{D}\sum_{i=1}^{D}(-y_i cos(0{,}5\sqrt{|y_i|})) = 0$$
$$x \in [-600, 600]^D$$

C07

Minimizar:

$$f(x) = \sum_{i=1}^{D-1}(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)$$
$$z = x + 1 - o$$
$$y = x - o$$

sujeto a:

$$g(x) = 0{,}5 - \exp\left(-0{,}1\sqrt{\frac{1}{D}\sum_{i=1}^{D} y_i^2}\right) - 3\exp\left(\frac{1}{D}\sum_{i=1}^{D} cos(0{,}1y)\right) + \exp(1) \leq 0$$
$$x \in [-140, 140]^D$$

C08

Minimizar:

$$f(x) = \sum_{i=1}^{D-1}(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)$$
$$z = x + 1 - o$$
$$y = (x - o)M$$

sujeto a:

$$g(x) = 0{,}5 - \exp\left(-0{,}1\sqrt{\frac{1}{D}\sum_{i=1}^{D} y_i^2}\right) - 3\exp\left(\frac{1}{D}\sum_{i=1}^{D} cos(0{,}1y)\right) + \exp(1) \leq 0$$
$$x \in [-140, 140]^D$$

C09

Minimizar:

$$f(x) = \sum_{i=1}^{D-1}(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)$$
$$z = x + 1 - o$$
$$y = x - o$$

sujeto a:

$$h(x) = \sum_{i=1}^{D}(ysin(\sqrt{|y_i|})) = 0$$
$$x \in [-500, 500]^D$$

C10

Minimizar:

$$f(x) = \sum_{i=1}^{D-1}(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)$$
$$z = x + 1 - o$$
$$y = (x - o)M$$

sujeto a:

$$h(x) = \sum_{i=1}^{D}(ysin(\sqrt{|y_i|})) = 0$$
$$x \in [-500, 500]^D$$

C11

Minimizar:

$$f(x) = \frac{1}{D}\sum_{i=1}^{D}(-z_i cos(2\sqrt{|z_i|}))$$
$$z = (x - o)M$$
$$y = x + 1 - o$$

sujeto a:

$$h(x) = \sum_{i=1}^{D-1}(100(y_i^2 - y_{i+1})^2 + (y_i - 1)^2) = 0$$
$$x \in [-100, 100]^D$$

C12

Minimizar:

$$f(x) = \frac{1}{D}\sum_{i=1}^{D}(z_i sin(\sqrt{|z_i|}))$$
$$z = x - o$$

sujeto a:

$$h(x) = \sum i = 1^{D-1}(z_i^2 - z_{i+1})^2 = 0$$
$$g(x) = \sum i = 1^{D}(z - 100cos(0{,}1z) + 10) \leq 0$$
$$x \in [-1000, 1000]^D$$

C13

Minimizar:

$$f(x) = \frac{1}{D}\sum_{i=1}^{D}(-z_i sin(\sqrt{|z_i|}))$$
$$z = x - o$$

sujeto a:

$$g_1(x) = -50 + \frac{1}{100D}\sum_{i}^{D} z_i^2 \leq 0$$
$$g_2(x) = \frac{50}{D}\sum_{i=1}^{D} sin\left(\frac{1}{50}\Pi z\right) \leq 0$$
$$g_3(x) = 75 - 50\left(\sum_{i=1}^{D}\frac{z_i^2}{4000} - \prod_{i=1}^{D} cos\left(\frac{z_i}{\sqrt{i}}\right) + 1\right) \leq 0$$
$$x \in [-500, 500]^D$$

C14

Minimizar:

$$f(x) = \sum_{i=1}^{D-1}(100(z_i^2 - z_{i+1})^2 + (z-1)^2)$$
$$z = x + 1 - o$$
$$y = x - o$$

sujeto a:

$$g_1(x) = \sum_{i=1}^{D}(-y_i cos(\sqrt{y_i})) - D \leq 0$$
$$g_2(x) = \sum_{i=1}^{D}(y_i cos(\sqrt{y_i})) - D \leq 0$$
$$g_3(x) = \sum_{i=1}^{D}(y_i sin(\sqrt{y_i})) - 10D \leq 0$$
$$x \in [-1000, 1000]^D$$

C15

Minimizar:

$$f(x) = \sum_{i=1}^{D-1}(100(z_i^2 - z_{i+1})^2 + (z-1)^2)$$
$$z = x + 1 - o$$
$$y = (x - o)M$$

sujeto a:

$$g_1(x) = \sum_{i=1}^{D}(-y_i cos(\sqrt{y_i})) - D \leq 0$$
$$g_2(x) = \sum_{i=1}^{D}(y_i cos(\sqrt{y_i})) - D \leq 0$$
$$g_3(x) = \sum_{i=1}^{D}(y_i sin(\sqrt{y_i})) - 10D \leq 0$$
$$x \in [-1000, 1000]^D$$

C16

Minimizar:

$$f(x) = \sum_{i=1}^{D}\frac{z_i^2}{4000} - \prod_{i=1}^{D} cos\left(\frac{z_i}{\sqrt{i}}\right) + 1$$
$$z = x - o$$

sujeto a:

$$g_1(x) = \sum_{i=1}^{D}[z_i^2 - 100cos(\Pi z_i) + 10] \leq 0$$
$$g_2(x) = \prod_{i=1}^{D} z_i \leq 0$$
$$h_1(x) = \sum_{i=1}^{D}(z_i sin(\sqrt{|z_i|})) = 0$$
$$h_2(x) = \sum_{i=1}^{D}(-z_i sin(\sqrt{|z_i|})) = 0$$
$$x \in [-10, 10]^D$$

C17

Minimizar:

$$f(x) = \sum_{i=1}^{D-1}(z_i - z_{i+1})^2$$
$$z = x - o$$

sujeto a:

$$g_1(x) = \prod_{i=1}^{D} z_i \leq 0$$
$$g_2(x) = \sum_{i=1}^{D} z_i \leq 0$$
$$h_1(x) = \sum_{i=1}^{D} (z_i sin(4\sqrt{z_i})) = 0$$
$$x \in [-10, 10]^D$$

C18

Minimizar:

$$f(x) = \sum_{i=1}^{D-1} (z_i - z_{i+1})^2$$
$$z = x - o$$

sujeto a:

$$g(x) = \frac{1}{D} \sum_{i=1}^{D} (-z_i sin(\sqrt{z_i})) \leq 0$$
$$h(x) = \frac{1}{D} \sum_{i=1}^{D} (z_i sin(\sqrt{z_i})) = 0$$
$$x \in [-50, 50]^D$$

Referencias

El fin de todo el discurso oído es éste: Teme a Dios, y guarda sus mandamientos; porque esto es el todo del hombre.

Eclesiastés 12:13

[1] A. Andreas and L. Wu-Sheng. *Practical Optimization: Algorithms and Engineering Applications*. Springer Publishing Company, Incorporated, 1st edition, 2007.

[2] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, Oxford, UK, 1996.

[3] M. R. Bonyadi and Z. Michalewicz. On the edge of feasibility: a case study of the particle swarm optimizer. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2014.

[4] M. J. Box. A new method of constrained optimization and a comparison with other methods. Computer Journal. 8, 42-52, 1965.

[5] A. Cervantes-Castillo, G. Leguizamon, and E. Mezura-Montes. Exploring boundary constraints using the modified brain storm optimization algorithm and a boundary search operator: Mbso-bs. In *2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, pages 1–6, Nov 2017.

[6] A. Cervantes-Castillo and E. Mezura-Montes. A study of constraint-handling techniques in brain storm optimization. *Congress on Evolutionary Computation (CEC), 2016 IEEE*, 2016.

[7] S. Cheng, Q. Qin, J. Chen, and Y. Shi. Brain storm optimization algorithm: a review. *Artificial Intelligence Review*, 46(4):445–458, Dec 2016.

[8] S. Cheng, Y. Shi, Q. Qin, and S. Gao. Solution clustering analysis in brain storm optimization algorithm. In *2013 IEEE Symposium on Swarm Intelligence (SIS)*, pages 111–118, April 2013.

[9] S. Cheng, Y. Sun, J. Chen, Q. Qin, X. Chu, X. Lei, and Y. Shi. A comprehensive survey of brain storm optimization algorithms. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 1637–1644, June 2017.

[10] J. W. Chinneck. The constraint consensus method for finding approximately feasible points in nonlinear programs. *INFORMS Journal on Computing*, 16(3):255–265, 2004.

[11] J. W. Chinneck. *Feasibility and Infeasibility in Optimization:Algorithms and Computational Methods*. Springer Science + Business Media, LLC, 2008.

[12] C. A. Coello-Coello. *Introducción a la Computación Evolutiva*. CINVESTAV-IPN.

[13] C. A. Coello-Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11):1245 – 1287, 2002.

[14] P. Cortez. *Modern Optimization with R*. Springer, September 2014.

[15] R. Datta and K. Deb. *Evolutionary Constrained Optimization*. Springer Publishing Company, Incorporated, 2014.

[16] K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2-4):311–338, 2000.

[17] S. Domínguez-Isidro and E. Mezura-Montes. A cost-benefit local search coordination in multimeme differential evolution for constrained numerical optimization problems. *Swarm and Evolutionary Computation*, 39:249–266, 2018.

[18] M. Dorigo. *Optimization, learning and natural algorithms. Ph.D. thesis*. PhD thesis, Politecnico di Milano, 1992.

[19] A. S. Drud. Conopt a large-scale grg code. *ORSA Journal on Computing*, 6(2):207–216, 1994.

[20] S. Elsayed, R. Sarker, and C. C. Coello. Enhanced multi-operator differential evolution for constrained optimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 4191–4198, July 2016.

[21] S. M. Elsayed, R. A. Sarker, and D. L. Essam. Ga with a new multi-parent crossover for solving ieee-cec2011 competition problems. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 1034–1040, June 2011.

[22] A. P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, 2006.

[23] L. J. Fogel. *Artificial Intelligence through Simulated Evolution*. John-Wiley, New York, 1966., 1966.

[24] G. Fuller and D. Tarwater. *Analityc Geometry, Seventh Edition*. Addison-Wesley Publishing Company, Inc., Reading Massachusetts, E.U.A, 1986.

[25] C. García-Martínez, M. Lozano, F. Herrera, D. Molina, and A.M. Sánchez. Global and local real-coded genetic algorithms based on parent-centric crossover operators. *European Journal of Operational Research*, 185(3):1088 – 1113, 2008.

[26] PE. Gill, W. Murray, and MA. Saunders. Snopt: an sqp algorithm for large-scale constrained optimization. report sol 97-3. Technical report, Systems Optimization Laboratory, Stanford University, 1997.

[27] D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. volume 1 of *Foundations of Genetic Algorithms*, pages 69 – 93. Elsevier, 1991.

[28] W. Gong, Z. Cai, and D. Liang. Adaptive ranking mutation operator based differential evolution for constrained optimization. *IEEE Transactions on Cybernetics*, 45(4):716–727, April 2015.

[29] Z. h. Zhan, W. n. Chen, Y. Lin, Y. j. Gong, Y. l. Li, and J. Zhang. Parameter investigation in brain storm optimization. In *2013 IEEE Symposium on Swarm Intelligence (SIS)*, pages 103–110, April 2013.

[30] N. M. Hamza, S. M. Elsayed, D. L. Essam, and R. A. Sarker. Differential evolution combined with constraint consensus for constrained optimization. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 865–872, June 2011.

[31] N. M. Hamza, D. L. Essam, and R. A. Sarker. Constraint consensus mutation-based differential evolution for constrained optimization. *IEEE Transactions on Evolutionary Computation*, 20(3):447–459, June 2016.

[32] N. M. Hamza, R. A. Sarker, and D. L. Essam. Differential evolution with multi-constraint consensus methods for constrained optimization. *Journal of Global Optimization*, 57(2):583–611, Oct 2013.

[33] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992.

[34] W. Ibrahim and J. W. Chinneck. Improving solver success in reaching feasibility for sets of nonlinear constraints. *Computers & Operations Research*, 35(5):1394 – 1411, 2008. Part Special Issue: Algorithms and Computational Methods in Feasibility and Infeasibility.

[35] R. Eberhart J. Kennedy. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks, 4, 1942-1948.*, 1995.

[36] T. Jansen. *Analyzing Evolutionary Algorithms: The Computer Science Perspective*. Springer Publishing Company, Incorporated, 2013.

[37] Dervis Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report, 2005.

[38] G. Leguizamón and C. A. Coello-Coello. Boundary search for constrained numerical optimization problems with an algorithm inspired by the ant colony metaphor. *Trans. Evol. Comp*, 13(2):350–368, April 2009.

[39] J. J. Liang, T. P. Runarsson, E. Mezura-Montes., M. Clerc, P. N. Suganthan, and C. A. Coello-Coello.and K. Deb. Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization. *Technical Report*, September 18, 2006.

[40] R. Mallipeddi and P.N. Suganthan. Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization. *Technical Report, Nanyang Technological University, Singapore*, April, 2010.

[41] E. Mezura-Montes and C. A. Coello Coello. Identifying on-line behavior and some sources of difficulty in two competitive approaches for constrained optimization. In *2005 IEEE Congress on Evolutionary Computation*, volume 2, pages 1477–1484 Vol. 2, Sept 2005.

[42] E. Mezura-Montes and C.A. Coello-Coello. Constraint-handling in nature-inspired numerical optimization: Past, present and future. In *Swarm and Evolutionary Computation 1 (2011) 173-194*. Elsevier B.V., 2011.

[43] Efrén Mezura-Montes and Jorge Isacc Flores-Mendoza. *Improved Particle Swarm Optimization in Constrained Numerical Search Spaces*, pages 299–332. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[44] Z. Michalewicz and G. Nazhiyath. Genocop iii: a co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. In *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, volume 2, pages 647–651 vol.2, Nov 1995.

[45] Z. Michalewicz and M. Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evol. Comput.*, 4(1):1–32, March 1996.

[46] M. A. Saunders. 1993 Murtagh, B. A. Minos 5.4 users guide (preliminary). technical report sol 83-20r. Technical report, Stanford University, 1993.

[47] A. F. Osborn and L. H. Bristol. *Applied imagination : principles and procedures of creative problem-solving.* New York : Scribners, 3rd. rev. ed edition, 1979. Includes index.

[48] S. S. Rao. *Engineering Optimization: Theory and Practice.* John Wiley & Sons, 2009.

[49] T.P. Runarsson and Yao X. Stochastic ranking for constrained evolutionary optimization. *Evolutionary Computation, IEEE Transactions on*, 4(3):284–294, Sep 2000.

[50] R. A. Sarker, S. M. Elsayed, and T. Ray. Differential evolution with dynamic parameters selection for optimization problems. *IEEE Transactions on Evolutionary Computation*, 18(5):689–707, Oct 2014.

[51] M. Schoenauer and Z. Michalewicz. Evolutionary computation at the edge of feasibility. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature — PPSN IV*, pages 245–254, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[52] L. Smith, J. W. Chinneck, and V. Aitken. Constraint consensus concentration for identifying disjoint feasible regions in nonlinear programmes. *Optimization Methods Software*, 28(2):339–363, April 2013.

[53] L. Smith, J. W. Chinneck, and V. Aitken. Improved constraint consensus methods for seeking feasibility in nonlinear programs. *Computational Optimization and Applications*, 54(3):555–578, Apr 2013.

[54] P. Spellucci. An sqp method for general nonlinear programs using only equality constrained subproblems. *Mathematical Programming*, 82(3):413–448, Aug 1998.

[55] T. Takahama and S. Sakai. Constrained optimization by the epsilon constrained differential evolution with gradient-based mutation and feasible elites. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1–8, July 2006.

[56] T. Takahama and S. Sakai. Constrained optimization by the epsilon constrained differential evolution with an archive and gradient-based mutation. *WCCI 2010 IEEE World Congress on Computational Intelligence July, 18-23, 2010 - CCIB, Barcelona, Spain*, 2010.

[57] T. Takahama, S. Sakai, and N. Iwane. Solving nonlinear constrained optimization problems by the epsilon constrained differential evolution. In *2006 IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2322–2327, Oct 2006.

[58] RA. Waltz and J. Nocedal. *KNITRO Users manual. Technical report OTC 2003/05.* Optimization Technology Center, Northwestern University,Evanston, IL, USA, 2003.

[59] Z. Y. Wu and A. R. Simpson. A self-adaptive boundary search genetic algorithm and its application to water distribution systems. *Journal of Hydraulic Research, 2002; 40(2):191-203*, 2002.

[60] XS. Yang. *Nature-Inspired Optimization Algorithms.* Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 1st edition, 2014.

[61] X. Yu and M. Gen. *Introduction to Evolutionary Algorithms.* Springer Publishing Company, Incorporated, 2012.

[62] S. Yuhui. Brain storm optimization algorithm. In Ying Tan, Yuhui Shi, Yi Chai, and Guoyin Wang, editors, *Advances in Swarm Intelligence*, volume 6728 of *Lecture Notes in Computer Science*, pages 303–309. Springer Berlin Heidelberg, 2011.

[63] Z. H. Zhan, J. Zhang, Y. H. Shi, and Hai lin Liu. A modified brain storm optimization. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8, June 2012.

[64] H. Zhu and Y. Shi. Brain storm optimization algorithms with k-medians clustering algorithms. In *2015 Seventh International Conference on Advanced Computational Intelligence (ICACI)*, pages 107–110, March 2015.