

---

USO DE TOLERANCIAS DINÁMICAS PARA EL MANEJO DE RESTRICCIONES EN PROBLEMAS DE  
OPTIMIZACIÓN USANDO LA COLONIA ARTIFICIAL DE ABEJAS

---

Tesis

Que presenta:  
**Mauricio Damián Araoz**

Para obtener el grado de:  
**Licenciado en Ingeniería en Sistemas Computacionales**

Director de Tesis:  
**Dr. Efrén Mezura Montes**  
**Dr. Giner Alor Hernández**

**Instituto Tecnológico de Orizaba**  
Orizaba, Veracruz.

**Laboratorio Nacional de Informática**  
**Avanzada**  
Xalapa, Veracruz.

Enero 2013

## Resumen

La colonia Artificial de Abejas (ABC por Artificial Bee Colony) es un algoritmo de búsqueda inspirado en el comportamiento de forrajeo de las abejas melíferas. En este trabajo de tesis se presenta la incorporación de la técnica  $\epsilon$ -*constraint* al algoritmo conocido como Modified Artificial Bee Colony SM-ABC para resolver problemas de optimización numérica con restricciones. SM-ABC utiliza una tolerancia dinámica para manejar restricciones de igualdad así como un operador para generar soluciones en el vecindario de la mejor solución obtenida hasta el momento.  $\epsilon$ -*constraint* utiliza una tolerancia que permite a soluciones no factibles considerarse como factibles y así favorecer un acercamiento promisorio a la zona factible del espacio de búsqueda.  $\epsilon$ -*constraint* sustituye al conjunto de reglas de factibilidad que originalmente usa SM-ABC para manejar las restricciones del problema. El algoritmo propuesto es comparado contra SM-ABC y contra el algoritmo original ABC usando medidas de desempeño encontradas en la literatura especializada para finalmente ser comparado contra algoritmos del estado del arte en optimización con restricciones. Los resultados obtenidos sugieren que el uso de  $\epsilon$ -*constraint* mejora el desempeño general del algoritmo.

# Índice general

<b>1. Introducción</b>	<b>6</b>
1.1. Motivación . . . . .	6
1.2. Objetivo de este trabajo de tesis . . . . .	6
1.3. Vista rápida . . . . .	7
<b>2. Optimización</b>	<b>8</b>
2.1. Definición . . . . .	8
2.2. Optimización en el mundo real . . . . .	8
2.2.1. Problemas de optimización numérica sin restricciones . . . . .	9
2.2.2. Problemas de optimización numérica con restricciones . . . . .	9
2.2.3. Restricciones activas . . . . .	10
2.3. Técnicas de Optimización . . . . .	11
2.3.1. Optimización mediante algoritmos metaheurísticos . . . . .	12
<b>3. Colonia Artificial de Abejas (ABC)</b>	<b>13</b>
3.1. Inteligencia Colectiva . . . . .	13
3.2. Motivación biológica . . . . .	15
3.3. Algoritmo . . . . .	16
<b>4. Algoritmo propuesto</b>	<b>18</b>
4.1. El algoritmo ABC y los problemas de optimización con restricciones . . . . .	18
4.2. Manejo de restricciones de igualdad . . . . .	18
4.3. Generación de la fuente candidata . . . . .	19
4.4. Operador de la abeja exploradora . . . . .	19
4.5. Manejo de límites . . . . .	20
4.6. Mecanismo de selección para manejo de restricciones . . . . .	21
4.7. Límite de una solución . . . . .	22
4.8. El algoritmo Static Phi Epsilon Constraint Artificial Bee Colony (SPEC-ABC) . . . . .	23
<b>5. Experimentos y resultados</b>	<b>25</b>
5.1. Diseño Experimental . . . . .	25
5.1.1. Parámetros de nuestro algoritmo . . . . .	27
5.1.2. Medidas de Desempeño . . . . .	27
5.2. Comparativo indirecto . . . . .	29
5.3. Comparativo directo . . . . .	32
5.4. Comparativo con medidas de desempeño . . . . .	35
5.4.1. Análisis de AFES, P, SP y FP . . . . .	35
5.4.2. EVALS . . . . .	38
5.4.3. PROGRESS RATIO . . . . .	41
5.5. Comparativo con algoritmos del estado del arte . . . . .	44

5.5.1. Comparativo entre SAPF-GA y nuestro algoritmo . . . . .	44
5.5.2. Comparativo entre HCOEA y nuestro algoritmo . . . . .	44
5.5.3. Comparativo entre ATMES y nuestro algoritmo . . . . .	44
<b>6. Conclusiones y trabajo futuro</b>	<b>47</b>
<b>A. Funciones de benchmark</b>	<b>48</b>

# Índice de figuras

2.1. Mínimo de $f(\vec{x})$ . . . . .	9
2.2. Ejemplo de la región factible $F$ en un espacio de búsqueda $P$ . . . . .	10
2.3. Ejemplo de la región factible $F$ formada por una restricción de igualdad y una restricción de desigualdad un espacio de búsqueda $P$ . . . . .	11
3.1. Paradigmas del Cómputo Inteligente [9]. (FS)Fuzzy Systems, (NN) Artificial Neural Networks, (AIS) Artificial Immune Systems, (EC) Evolutionary Computation, (SI) Swarm Intelligence. . . . .	14
3.2. Proceso de exploración de las abejas melíferas en el sistema visualizador de la Colonia Artificial de Abejas . . . . .	15
3.3. Elementos básicos del forrajeo de las abejas . . . . .	16
4.1. Gráfica que ejemplifica la tolerancia de una restricción de igualdad convertida a una restricción de desigualdad . . . . .	19
4.2. La figura <b>a</b> muestra el vector resultante de $x_k$ y $x_j$ . La figura <b>b</b> muestra las posibles soluciones candidatas obtenidas con 4.3 para cuando $\phi$ adquiere un valor positivo o negativo . . . . .	20
4.3. Figura que muestra un ejemplo del mecanismo de la abeja exploradora . . . . .	21
4.4. Figura que muestra el espacio de tolerancia en regiones no factibles establecidas por las restricciones del problema a resolver. . . . .	22
4.5. Pseudocódigo del Static Phi Epsilon Constraint ABC (SPEC-ABC) . . . . .	24

# Índice de cuadros

3.1. Parámetros del ABC . . . . .	16
3.2. Algoritmo de la Colonia Artificial de Abejas (ABC) . . . . .	17
4.1. Parámetros del SPEC-ABC . . . . .	23
5.1. Resumen con las principales características de los problemas a resolver . . . . .	25
5.2. Parámetros del algoritmo propuesto . . . . .	27
5.3. Comparación indirecta de los primeros 13 problemas entre el algoritmo ABC de Dervis Karaboga, SM-ABC de Omar Cetina y SPEC-ABC. Los mejores resultados se enmarcan en negritas. . . . .	30
5.4. Comparación indirecta de los últimos 11 problemas entre el algoritmo ABC de Dervis Karaboga, SM-ABC de Omar Cetina y SPEC-ABC. Los mejores resultados se enmarcan en negritas. "Significa que no fueron encontradas soluciones factibles en ninguna ejecución independiente. . . . .	31
5.5. Comparación directa ABC, SM-ABC and SPEC-ABC en las primeras trece funciones de prueba. Los mejores resultados se remarcan en negritas. . . . .	33
5.6. Comparación directa entre ABC, SM-ABC and SPEC-ABC en las últimas once funciones de prueba. Los mejores resultados se remarcan en negritas. "Significa que no fueron encontradas soluciones factibles en ninguna ejecución independiente. . . . .	34
5.7. Comparativo de las primeras 4 medidas de desempeño en las primeras trece funciones de prueba. Los mejores resultados se remarcan con negritas. "Significa que la medida de desempeño no se puede calcular dada la ausencia de ejecuciones exitosas y/o factibles. . . . .	36
5.8. Comparativo de las primeras cuatro medidas de rendimiento en las últimas once funciones de prueba. Los mejores resultados se remarcan con negritas. "Significa que la medida de desempeño no se puede calcular dada la ausencia de ejecuciones exitosas y/o factibles. . . . .	37
5.9. Comparación de resultados para la medida de rendimiento EVALS en las primeras trece funciones de prueba. Los mejores resultados se remarcan en negritas . . . . .	39
5.10. Comparación de resultados para la medida de rendimiento EVALS en las últimas once funciones de prueba. Los mejores resultados se remarcan en negritas. "Significa que no fueron encontradas soluciones factibles en ninguna ejecución independiente. . . . .	40
5.11. Comparación de resultados para la medida de rendimiento PROGRESS RATIO en las primeras trece funciones de prueba. Los mejores resultados se remarcan en negritas. . . . .	42
5.12. Comparación de resultados para la medida de rendimiento PROGRESS RATIO en las últimas once funciones de prueba. Los mejores resultados se remarcan en negritas. "Significa que no fueron encontradas soluciones factibles en ninguna ejecución independiente. . . . .	43
5.13. Comparación de nuestro algoritmo (titulado SPEC-ABC) con respecto a los algoritmos del estado del arte (SAPF-GA [26], HCOEA [27] and ATMES [28]) en las 13 funciones de benchmark (g01-g13). El resultado marcado en negritas indica el mejor resultado u óptimo global (mejor solución conocida) alcanzado. . . . .	46

A.1. Conjunto de datos del problema g19. . . . .	56
A.2. Conjunto de datos del problema g20. . . . .	57

# Capítulo 1

## Introducción

El presente trabajo se encuentra enfocado en la solución de problemas de optimización numérica con restricciones. Dado que los métodos de optimización tradicional pueden llegar ser insuficientes para la búsqueda de soluciones se ha propuesto implementar metaheurísticas para la solución a dichos problemas. En este caso en particular la propuesta se basa en la mejora de un algoritmo novedoso de inteligencia colectiva llamado SM-ABC (Scout Modified Artificial Bee Colony [4]).

### 1.1. Motivación

La motivación de este trabajo se numera de la siguiente manera:

- Los problemas de optimización, que se ejemplifican en las secciones siguientes son problemas abiertos.
- El número de algoritmos de inteligencia colectiva basados en abejas melíferas propuestos es muy reducido [1].
- Sólo algunos de los algoritmos basados en el comportamiento de las abejas han sido usados en problemas de optimización numérica, la mayoría de ellos sólo han sido usados en optimización combinatoria.
- Las implementaciones del ABC (ABC[12] y SM-ABC[4]) convergen rápidamente y son sensibles a caer en óptimos locales.

### 1.2. Objetivo de este trabajo de tesis

Mejorar el rendimiento que implica encontrar mejores resultados en los problemas de benchmark sin incrementar el número de evaluaciones y evitar la convergencia prematura del algoritmo, estos resultados serán comparados contra el ABC[12] y SM-AB[4] que son las 2 implementaciones del ABC conocidas hasta el momento. Este algoritmo ha sido seleccionado dado que es un algoritmo de fácil implementación con un número de parámetros relativamente bajo. Cabe destacar que el ABC es un algoritmo prometedor al mostrar resultados competitivos en problemas de benchmark.

Para alcanzar el objetivo anteriormente planteado se propone el *Static Phi Epsilon Constraint Artificial Bee Colony (SPEC-ABC)* basado en el SM-ABC[4]. En el algoritmo propuesto los cambios son localizados en el mecanismo de selección de las soluciones con la técnica llamada  $\epsilon$ -constraint[25] evitando la discriminación rápida soluciones ligeramente no factibles; además, se modifica el operador de las abejas para la generación de una nueva solución con la técnica propuesta de *static* -  $\phi$  la cual permite el muestreo de soluciones en zonas más prometedoras según el valor generado de  $\phi$ .



## 1.3. Vista rápida

Para tener una perspectiva más amplia sobre de los contenidos de este trabajo, a continuación se muestra un breve resumen de cada uno de los capítulos:

- **Capítulo 2. Optimización:** Se muestran conceptos básicos sobre optimización; algunas aplicaciones en el mundo real, los problemas de optimización sin restricciones así como los problemas de optimización con restricciones (problemas sobre los cuales se basa este trabajo de tesis).
- **Capítulo 3. Colonia Artificial de Abejas:** Se describen los conceptos básicos de Inteligencia Colectiva, así también se explica el algoritmo original del ABC (Artificial Bee Colony), su motivación biológica y la adaptación de ésta a su modelo algorítmico.
- **Capítulo 4. Descripción del algoritmo:** Se describe a detalle cada parámetro del algoritmo propuesto, técnicas para el manejo de restricciones, manejo del operador así como el pseudocódigo del algoritmo SPEC-ABC.
- **Capítulo 5. Experimentos y resultados:** Se hacen comparaciones directas e indirectas del algoritmo propuesto con sus predecesores así como comparaciones indirectas con algoritmos del Estado del Arte. Aquí también se describe la comparación de resultados de los algoritmos, las medidas de rendimientos a usar.
- **Capítulo 6. Conclusiones y trabajo futuro:** Se presenta una conclusión basada en los resultados obtenidos así como las áreas de oportunidad que no se encuentran dentro del alcance de este proyecto de tesis.
- **Apéndice A. Funciones de benchmark:** En este apartado se describen las 24 funciones de benchmark así como su detalle de implementación.

Derivado del presente trabajo de tesis se publicó el siguiente artículo en uno de los dos congresos más importantes del área de algoritmos evolutivos y de inteligencia colectiva:

Efrén Mezura-Montes, Mauricio Damián-Araoz and Omar Cetina-Domínguez, Smart Flight and Dynamic Tolerances in the Artificial Bee Colony for Constrained Optimization, in Proceedings of the IEEE Congress on Evolutionary Computation (CEC'2010), pages: 4118–4125, IEEE Press, July, 2010.

# Capítulo 2

## Optimización

### 2.1. Definición

Optimización es definida como:

- Búsqueda de la mejor manera de realizar una actividad.[23]
- Obtener el óptimo; sacar el máximo provecho de, obtener el mejor.[17]

### 2.2. Optimización en el mundo real

Los algoritmos de optimización [11] se han convertido incrementalmente populares en actividades de ingeniería, principalmente por la viabilidad y el alcance de alto desempeño en un equipo de cómputo. Éstos se utilizan altamente en problemas de ingeniería donde los objetivos son obtener mínimos o máximos. Por ejemplo los algoritmos de optimización se utilizan comúnmente en diseños de aeronaves donde se trata de minimizar el sobrepeso. Así, la minimización del peso en los componentes de una aeronave es lo que les importa a los diseñadores aeroespaciales. A los ingenieros mecánicos por otra parte les importa diseñar componentes mecánicos con el propósito de lograr cualquier ahorro en costos de manufactura o maximizar la vida de un componente. Los ingenieros civiles están interesados en diseñar edificios, puentes o cualquier tipo de construcciones para lograr un mínimo costo, obtener un máximo grado de seguridad o ambas. Ingenieros eléctricos están interesados en diseñar redes de comunicación para minimizar el tiempo de respuesta en la comunicación de un nodo a otro.

Todos los problemas anteriormente mencionados están involucrados en tareas de minimización o maximización, por lo que se puede resumir que los problemas de optimización son aquellos que, consisten en buscar en un conjunto de soluciones posibles para el problema en cuestión, aquella que satisfaga de mejor manera una serie de condiciones planteadas. De esta manera los problemas de optimización se clasifican en dos tipos:

- **Problemas de optimización numérica:** Se busca un conjunto de valores para las variables del problema de manera que al sustituirse en la función objetivo se maximice o minimice el valor de esta función. Un ejemplo de este problema podrá ser el diseño de una pieza mecánica donde se buscan valores óptimos de sus dimensiones para minimizar su costo de fabricación.
- **Problemas de optimización combinatoria:** Se busca encontrar el ordenamiento de un conjunto de elementos de manera que minimice o maximice el valor de la función objetivo. Un ejemplo de este tipo de problemas es el del agente viajero, que debe recorrer un conjunto de ciudades, pasando por ellas sólo una vez, de manera que regrese a su punto de salida y se minimice el costo del viaje. Aquí se desea encontrar el orden óptimo de recorrido de las ciudades.

A continuación se listan los problemas detectados al resolver, en general, problemas de optimización [20]:

- El número posible de soluciones (espacios de búsqueda) es demasiado grande.
- El problema es tan complicado que, con la intención de obtener una solución, se deben utilizar modelos simplificados del mismo y por ende la solución es poco útil.
- La función de evaluación que describe la calidad de cada solución en el espacio de búsqueda varía con el tiempo y/o tiene ruido.
- Las soluciones posibles están altamente restringidas, lo cual dificulta incluso la generación de al menos una solución factible.

En esta sección se tratan a detalle los problemas de optimización numérica; se abordan los conceptos de problemas de optimización sin restricciones, problemas de optimización con restricciones y restricciones activas.

### 2.2.1. Problemas de optimización numérica sin restricciones

Un problema de optimización numérica sin restricciones no tiene limitantes en los valores que las variables pueden tomar, excepto por la restricción opcional de los límites superior e inferior que cada variable posee.

En los problemas de optimización numérica se intenta optimizar (minimizar o maximizar) una función, esto es posible ejemplificarlo de manera general de la siguiente manera, donde el valor óptimo del problema ( $x^*$ ) es ambos, el valor mínimo de la función  $f(\vec{x})$  y el valor máximo de la función  $-f(\vec{x})$  mostrado en la figura 2.1.

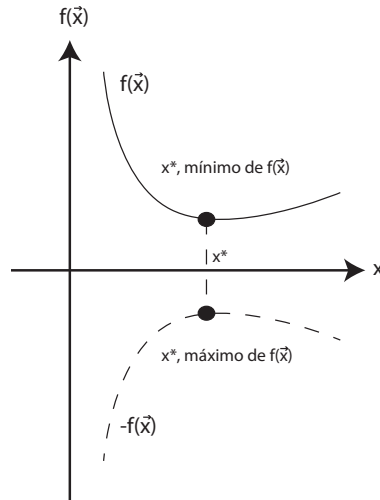


Figura 2.1: Mínimo de  $f(\vec{x})$

### 2.2.2. Problemas de optimización numérica con restricciones

El algoritmo de la Colonia Artificial de Abejas se enfoca en los problemas conocidos como problemas generales de programación no lineal. La mayoría de los problemas del mundo real requieren soluciones a problemas con restricciones, por ejemplo, minimizar el costo de producción sujeta a insumos, desperdicio de materiales, tiempo, etc. Se puede definir un problema sujeto a restricciones, sin pérdida de generalidad como:

$$\text{Minimizar } f(\vec{x}) \tag{2.1}$$

Sujeta a:

$$g_i(\vec{x}) \leq 0, i = 1, \dots, m \quad (2.2)$$

$$h_j(\vec{x}) = 0, j = 1, \dots, p \quad (2.3)$$

donde  $x \in \mathbf{R}^n$  es un vector de  $n$  variables de decisión  $\vec{x} = [x_1, x_2, \dots, x_n]^T$  y cada  $x_i, i = 1, \dots, n$  está acotado por límites inferiores y superiores  $L_i \leq x_i \leq U_i$ , los cuales definen el espacio de búsqueda  $\mathcal{P}, \mathcal{F}$  que es el conjunto de todas las funciones que satisfacen las restricciones del problema y se llama zona factible, siendo claro que  $\mathcal{F} \in \mathcal{P}, m$  es el número de restricciones de igualdad y  $p$  es el número de restricciones de igualdad[18]. Las restricciones y función objetivo pueden ser lineales o no lineales. La Figura 2.2 ilustra la región factible de un problema en dos dimensiones.

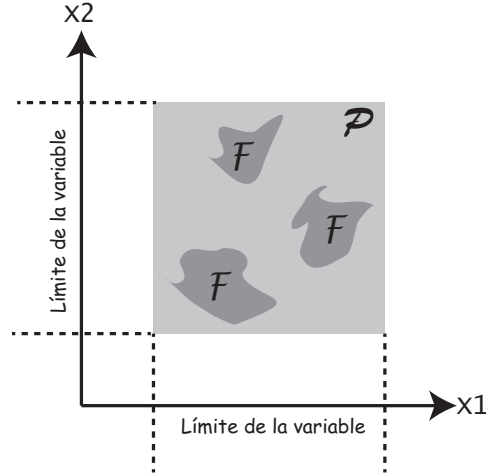


Figura 2.2: Ejemplo de la región factible  $F$  en un espacio de búsqueda  $P$

En este problema se pretende encontrar una solución que minimice cierta medida de calidad conocida como función objetivo (Ecuación 2.1) que además debe satisfacer un conjunto de restricciones de desigualdad e igualdad asociadas al problema (Ecuación 2.2 y 2.3).

Si la función objetivo a optimizar o alguna de sus restricciones son no lineales, el problema es conocido como programación no lineal (NLP por sus siglas en inglés Non Linear Problem) [22].

Para la solución de este tipo de problemas existen las soluciones tradicionales que proporciona la optimización clásica [22] y los métodos no tradicionales.

### 2.2.3. Restricciones activas

Se dice que la restricción  $g(\vec{x}) \leq 0$  esta activa en el punto  $\vec{x}$  si  $g(\vec{x}) = 0$ . Las restricciones de igualdad  $h(\vec{x})$  se encuentran activas por definición. La importancia de las restricciones activas es que su existencia en un problema incrementa la dificultad para localizar la solución óptima global factible.

La mayoría de los métodos para la solución de problemas con restricciones comienzan con una solución que se encuentra fuera de la región de soluciones factibles, después de cierto tiempo computacional, la búsqueda alcanza la región de factibilidad. Cuando se intentan buscar soluciones que se encuentren en los límites de entre las regiones factibles y no factibles, como muestra la figura 2.3, al algoritmo le tomará mucho tiempo generar soluciones que se encuentren dentro de dicha región.

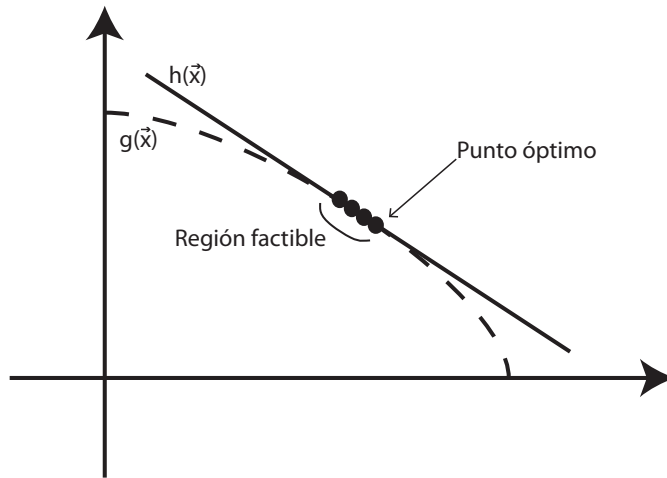


Figura 2.3: Ejemplo de la región factible  $F$  formada por una restricción de igualdad y una restricción de desigualdad un espacio de búsqueda  $P$

## 2.3. Técnicas de Optimización

La Investigación de Operaciones (IO) es el área dentro de las matemáticas dedicada al estudio, entre otros temas, de técnicas de solución para problemas de optimización [18]. En la investigación de operaciones no se tiene una sola técnica general con la que se resuelvan todos los modelos matemáticos que surgen en la práctica. En lugar de ello, la clase y complejidad del modelo definen la naturaleza del método de solución. La técnica más importante de la investigación de operaciones es la **programación lineal**. Se diseña para modelos con funciones objetivos y restricciones estrictamente lineales. Deb [11] clasifica las técnicas para resolver problemas de optimización numérica en las siguientes categorías:

### 1. Métodos tradicionales.

- Técnicas de variable simple: Divididas en métodos directos (utilizan la función a optimizar para guiar la búsqueda) y de gradiente (utiliza información de gradiente para guiar la búsqueda).
- Técnicas multivariable: Divididas en métodos directos y de gradiente. Realizan búsquedas en múltiples dimensiones, valiéndose, en ocasiones, de técnicas de variable simple.
- Técnicas para problemas con restricciones: Realizan búsquedas en espacios restringidos, usualmente utilizando técnicas de multivariable y/o de variable simple de manera iterativa.
- Técnicas especializadas: Métodos para problemas particulares como programación entera (variables que sólo toman valores enteros).

### 2. Métodos no tradicionales. Métodos que incorporan conceptos heurísticos para mejorar la búsqueda.

Una peculiaridad de la mayor parte de las técnicas de investigación de operaciones es que en general las soluciones no se obtienen en formas cerradas. En lugar de ello se determinan mediante algoritmos. Un algoritmo proporciona reglas fijas de cómputo que se aplican en forma repetitiva en el problema y cada repetición (llamada iteración) obtiene una solución cada vez más cercana a la óptima [24].

Algunos problemas de optimización pueden tener características que hacen difícil el aplicar técnicas de investigación de operaciones. Más aún, de poderse aplicar, el resultado podría no ser el esperado.

### 2.3.1. Optimización mediante algoritmos metaheurísticos

Cuando los problemas a resolver se tornan demasiado complejos para utilizar métodos tradicionales es conveniente la utilización de métodos metaheurísticos también llamados métodos de aproximación, para la solución de dichos problemas. Existen diversos problemas dónde estos métodos pueden ser aplicados, sin embargo, los resultados y/o el tiempo requerido para obtener una solución factible (que satisfaga todas las restricciones del problema) pueden no ser los esperados por quien aplica este método.

Los métodos tradicionales siempre deben considerarse como la primera opción para resolver un problema de optimización dado que pueden garantizar la convergencia al óptimo global (la mejor de todas las soluciones) bajo ciertas condiciones. Sin embargo, si la aplicación del método tradicional es compleja, el costo computacional es alto y/o los resultados son buenos pero no los esperados, entonces puede considerarse el uso de un método no tradicional, un algoritmo metaheurístico.

Dentro de los métodos metaheurísticos se tiene un grupo que basa su funcionamiento en fenómenos “inteligentes” encontrados en la naturaleza y conocidos como metaheurísticas bio-inspiradas [8] o algoritmos bio-inspirados. Dos sub-grupos se distinguen de acuerdo con el tipo de fenómeno natural en el que se basan:

1. Algoritmos Evolutivos [7]
2. Inteligencia Colectiva [8]

Los algoritmos evolutivos basan su funcionamiento en la teoría de la evolución de las especies, la supervivencia del más apto y la transmisión de características de padres a hijos [10]. Por otro lado la inteligencia colectiva toma comportamientos de seres vivos que interactúan de manera local en su ambiente, y de esa interacción surgen comportamientos sociales que permiten a esos seres resolver problemáticas complejas de manera conjunta (parvadas de aves, banco de peces, cúmulo de insectos, hormigas, bacterias entre otros) [8].

## Capítulo 3

# Colonia Artificial de Abejas (ABC)

Del acrónimo *Artificial Bee Colony (ABC) Algorithm*. Uno de los más recientes algoritmos desarrollados en 2005 por **Dervis Karaboga** [12] motivado por el comportamiento inteligente de las abejas. El modelo utiliza operadores inspirados en la *Optimización mediante Cúmulos de Partículas* (Particle Swarm Optimization, PSO [8]) y el algoritmo de *Evolución Diferencial* (Differential Evolution DE [15]) y usa parámetros comunes de control como el tamaño de la colonia y el número máximo de ciclos. ABC es un algoritmo de optimización en la que procedimientos de búsqueda basados en soluciones llamados fuentes alimento se modifican por las *abejas artificiales* donde el objetivo es descubrir lugares donde se encuentren fuentes de alimento con altas cantidades de néctar (las mejores soluciones) y, finalmente, seleccionar la que tenga mayor calidad. En el algoritmo ABC las abejas artificiales (*empleadas y observadoras*) viajan en un espacio multidimensional para realizar búsquedas basadas en la experiencia para posteriormente reposicionarse. Algunas otras llamadas *exploradoras* vuelan y buscan la comida de manera aleatoria sin hacer uso de la experiencia. Si la cantidad de néctar de la nueva fuente es mayor a la anterior que ellas recuerdan, entonces borran de su memoria esa posición para memorizar la nueva fuente. En resumen, el algoritmo ABC combina métodos de búsqueda casi-local llevada a cabo por abejas empleadas y observadoras con métodos de búsqueda global administrada por abejas observadoras y exploradoras tratando de equilibrar los procesos de exploración y explotación.

### 3.1. Inteligencia Colectiva

La Inteligencia Colectiva (Swarm Intelligence) es una rama del Cómputo Inteligente (ver Figura 3.1) que surge a través de la observación de las sociedades en la naturaleza como colonias de hormigas, parvadas de aves, bancos de peces, colonias de bacterias y en nuestro caso colonias de abejas. La observación de estos fenómenos naturales muestran cómo todos los individuos en conjunto trabajan para alcanzar sus objetivos, por ejemplo, una colonia de hormigas trabajando para encontrar el camino más corto a su fuente de alimento. Estos tipos de comportamientos se usan en el ámbito computacional para la optimización de recursos siendo estos algoritmos de alguna manera eficientes, adaptativos y robustos.

El término Inteligencia Colectiva (Swarm Intelligence) se introdujo por primera vez por Beni y Wang [2] en el área de *robótica celular* y desde entonces se trabajó dando inicio a los algoritmos basados en este pensamiento como el algoritmo de *Optimización de la Colonia de Hormigas* (Ant Colony Optimization ACO) creada por Marco Dorigo [3] una metaheurística para la solución de problemas combinatorios.

Lo más interesante del comportamiento de las especies que viven en grupos de auto-organización es que no es posible identificar un control centralizado como por ejemplo dentro de una parvada de aves o dentro de un banco de peces. Dentro de estas sociedades los individuos no son conscientes del comportamiento global ni tienen información total sobre el medio en el que se encuentran. Es por esto que en base a la comunicación poseen la habilidad de reagruparse y moverse de manera

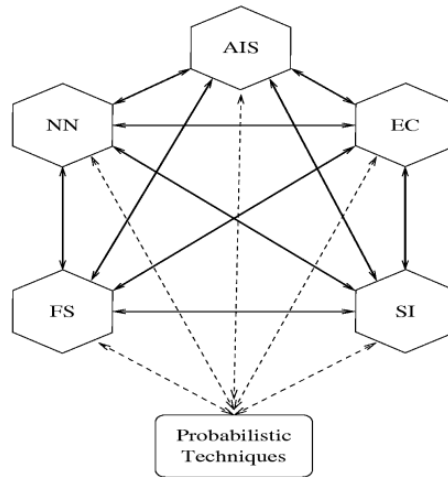


Figura 3.1: Paradigmas del Cómputo Inteligente [9]. (FS)Fuzzy Systems, (NN) Artificial Neural Networks, (AIS) Artificial Immune Systems, (EC) Evolutionary Computation, (SI) Swarm Intelligence.

sincronizada.

Bonabeu y Mayer identifican las características que logran un comportamiento social exitoso dentro de las sociedades animales:

- **Flexibilidad:** La colonia es capaz de adaptarse a cambios en el ambiente
- **Robustez:** Incluso si uno o más individuos llegasen a fallar, la colonia es capaz de seguir trabajando.
- **Auto-organización:** El control de las actividades no se encuentran centralizadas.

Existen distintos trabajos en algoritmos de inteligencia colectiva. El modelo llamado Artificial Bee Colony es uno de ellos, en el cual se intenta emular el comportamiento de forrajeo de la abeja melífera, pero cuando se intenta comprender el funcionamiento de dicho algoritmo algunas veces resulta con cierto grado de complejidad para lo que el autor Dervis Karaboga [12] creó una aplicación demostrativa del algoritmo disponible para descargar en <http://mf.erciyes.edu.tr/abc/software.htm> en el que se muestra un panel para ajustar los parámetros y una gráfica de convergencia en la cual es notable la rapidez del algoritmo para encontrar soluciones.

Para generar un ambiente gráfico de mayor comprensión a este algoritmo el autor del presente trabajo de tesis elaboró un sistema visualizador en tres dimensiones durante el periodo de residencias [5] en el Laboratorio Nacional de Informática Avanzada (LANIA) donde resuelve un problema sencillo de optimización numérica y puede observarse el comportamiento del algoritmo ABC al resolverlo. La Figura 3.2 muestra una imagen del visualizador durante la fase de exploración de las abejas melíferas.



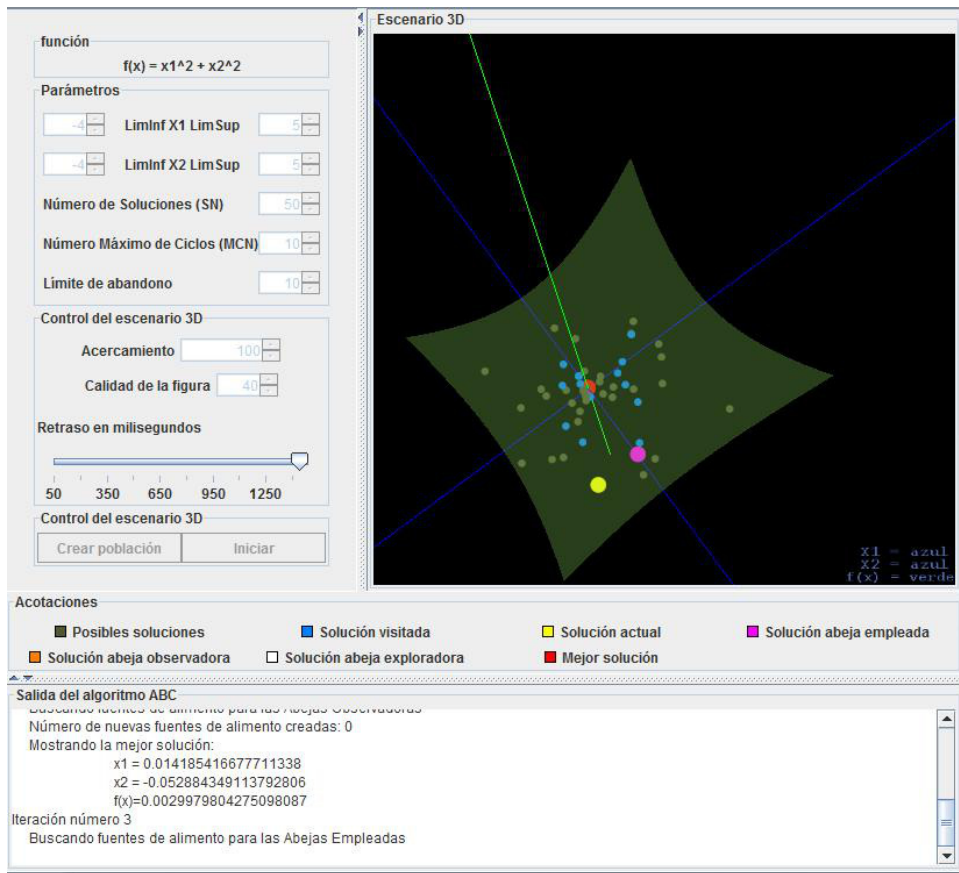


Figura 3.2: Proceso de exploración de las abejas melíferas en el sistema visualizador de la Colonia Artificial de Abejas

### 3.2. Motivación biológica

El concepto básico del modelo de inteligencia colectiva basado en el enjambre de abejas melíferas está compuesto de tres componentes principales: (1) fuente de alimentos, (2) recolectoras empleadas y (3) recolectoras desempleadas. Se definen dos modelos de comportamiento: La recolección del néctar y el abandono de una fuente de alimento.

- I. **Fuente de alimento:** El cálculo del valor de la fuente depende de muchos factores como la proximidad a la colmena, el nivel de concentración de alimento y la fácil extracción de éste.
- II. **Recolectoras empleadas:** Están asociadas a una fuente de alimento actualmente explotada. Son las encargadas de llevar la información de la fuente en particular, la distancia y la dirección con respecto a la colmena, así como la factibilidad de llegar a ella para las recolectoras.
- III. **Recolectoras desempleadas:** Continuamente buscando nuevas fuentes de alimento. Existen dos tipos de recolectoras desempleadas: las exploradoras que buscan nuevas fuentes de alimento cercanas a la colmena y las observadoras, que esperan en el nido y seleccionan una fuente de alimento con base en la información compartida por una recolectora empleada.

La parte más importante dentro de la colmena es el intercambio de información por medio de una danza cuya duración indica la rentabilidad de la fuente, el ángulo con respecto al sol indica la dirección de la fuente y el número de movimientos en zig-zag indica la distancia.

Cuando una fuente de alimento se agota, la abeja o abejas empleadas en ellas se convierten en abejas desempleadas y tendrán que decidir entre convertirse en abejas exploradoras o regresar a la colmena como abejas observadoras. La Figura 3.3 muestra los elementos básicos en el forrajeo de las abejas.

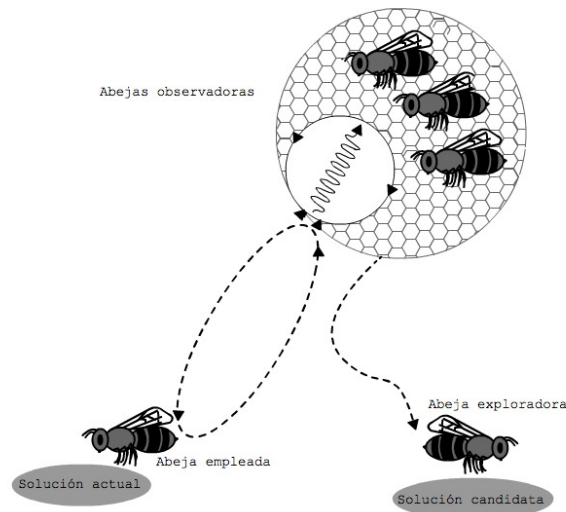


Figura 3.3: Elementos básicos del forrajeo de las abejas

### 3.3. Algoritmo

En el algoritmo ABC la colonia de abejas artificiales se compone también de 3 grupos de abejas: *abejas empleadas*, *abejas observadoras* y *abejas exploradoras*. El número de abejas empleadas es usualmente igual al número de fuentes de alimento y se asignará una abeja empleada a cada una de las fuentes. Al llegar a dicha fuente, la abeja calculará una nueva solución (volará hacia otra fuente de alimento cercana) a partir de ésta y conservará la mejor solución. El número de abejas observadoras es también usualmente igual al de abejas empleadas y se asignarán a una fuente de alimento con base en la aptitud de éstas, al igual que las abejas empleadas, calculará una nueva solución a partir de su fuente de alimento. Cuando una fuente no mejora después de un cierto número de iteraciones se abandona, siendo reemplazada por aquella encontrada por una abeja exploradora, lo cual implica generar una nueva solución aleatoriamente. La ventaja de este algoritmo es el bajo número de parámetros que se requieren como se ve en el Cuadro 3.1.

Cuadro 3.1: Parámetros del ABC

Nombre	Símbolo	Descripción
Número de soluciones	$SN$	Número de soluciones (fuentes de alimento)
Número de ciclos	$MCN$	Número de ciclos (iteraciones) que ejecutará el ABC
Límite	$limit$	Número de ciclos que se conservará una solución sin mejora antes de ser reemplazada por una generada por alguna abeja exploradora

En el ABC, las abejas son vistas como operadores de variación, pues cuando una de ellas llega a una fuente de alimento, calcula una nueva solución candidata  $v_{i,g}$  utilizando la formula 3.1, donde  $x_{i,g}$  representa la solución en la que la abeja se encuentra en ese momento,  $x_{k,g}$  es una fuente de alimento aleatoria (y distinta de  $x_{i,g}$ ),  $g$  es el número de ciclo actual del programa y  $\phi$  es un número

real aleatorio en el intervalo  $[-1,1]$ .

$$v_{i,g} = x_{i,g} + \phi \cdot (x_{i,g} - x_{k,g}) \quad (3.1)$$

El Cuadro 3.2 muestra los pasos del algoritmo ABC, donde se puede observar también que si una solución (fuente de alimento) no es mejorada por las abejas empleadas u observadores dentro de un número de iteraciones determinada por el valor del parámetro "limit", se elimina y en su lugar se genera una nueva solución de manera aleatoria (la abeja exploradora encuentra una nueva fuente de alimento).

Cuadro 3.2: Algoritmo de la Colonia Artificial de Abejas (ABC)

1	<b>Begin</b>
2	Inicialización de la población de soluciones $x_{i,0}, i = 1, \dots, SN$
3	Evaluar población
4	$g = 1$
5	<b>Repeat</b>
6	Producir nuevas soluciones $v_{i,g}$ para las abejas empleadas utilizando la ecuación 3.1 y evaluarlas
7	Conservar la mejor solución entre la actual y la candidata
8	Seleccionar las soluciones que serán visitadas por una abeja observadora según la aptitud
9	Producir nuevas soluciones $v_{i,g}$ para las abejas observadoras utilizando la ecuación 3.1 y evaluarlas
10	Conservar la mejor solución entre la actual y la candidata
11	Determinar si existe una fuente abandonada y reemplazarla utilizando una abeja exploradora
12	Memorizar la mejor solución encontrada hasta ese momento
13	$g = g + 1$
14	<b>Until</b> $g = MCN$
15	<b>End</b>

# Capítulo 4

## Algoritmo propuesto

En este capítulo se alcanza el objetivo de esta tesis el cual es adaptar el algoritmo SM-ABC [4] de manera novedosa para la solución de problemas de optimización con restricciones donde dichas adaptaciones corresponden en agregar un mecanismo para el manejo de restricciones bajo el enfoque  $\epsilon$  - *constraint* y una modificación al operador de variación que representan las abejas empleadas y observadoras. Ambas modificaciones son descritas más adelante. El enfoque manejado es llamado “Static  $\phi - \epsilon$  constraint”. A continuación se describen los elementos del algoritmo tomado como base para este trabajo (SM-ABC) y se presentan, de considerarse, los cambios realizados para dar origen a la nueva propuesta SPEC-ABC.

### 4.1. El algoritmo ABC y los problemas de optimización con restricciones

Como los algoritmos metaheurísticos inspirados en la naturaleza, el algoritmo ABC originalmente se diseñó para el manejo de problemas de optimización sin restricciones. La primera versión del ABC que permite el manejo de problemas de optimización con restricciones implementa una técnica diseñada originalmente para algoritmos genéticos elaborada por Deb [6].

El método del manejo de restricciones de Deb consiste en tres simples reglas basadas en la factibilidad de las soluciones: (1) Siempre se prefiere una solución factible sobre aquella que no lo es, (2) entre dos soluciones factibles, se selecciona aquella que tenga un mejor valor en su función objetivo, (3) entre dos soluciones no factibles, se selecciona aquella que tiene el menor valor de violación de restricciones.

### 4.2. Manejo de restricciones de igualdad

Como se indica en [4] la presencia de restricciones de igualdad disminuye significativamente el tamaño de la zona factible del espacio de búsqueda. De ahí que es una práctica común re-escribir cada restricción de igualdad como una restricción de desigualdad como se muestra en (4.1):

$$|h(\vec{x})| - \psi \leq 0 \quad (4.1)$$

Donde  $\psi$  es la tolerancia dada para la restricción de igualdad extendiendo el área de búsqueda. La desventaja es que el uso de este método puede arrojar resultados fuera del área factible dada la tolerancia. La Figura 4.1 mostrada a continuación explica el funcionamiento de la tolerancia para los problemas de optimización con restricciones de igualdad.

Existen problemas en los cuales una tolerancia bastante pequeña es insuficiente para alcanzar buenos resultados, además, si la tolerancia ( $\psi$ ) usada es ajustada con un valor elevado, los resultados pueden localizarse alejados de la zona factible lo cual los haría poco útiles. En estos casos el usar

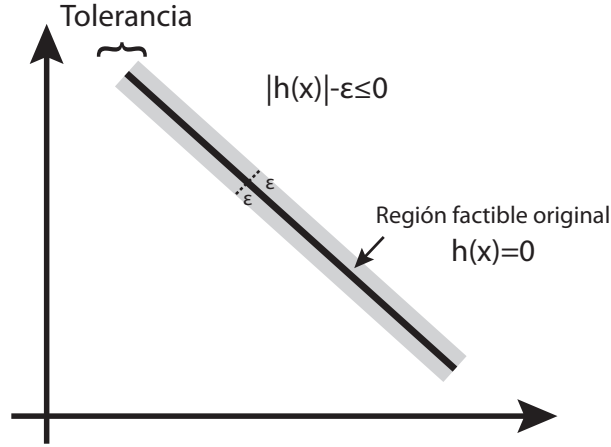


Figura 4.1: Gráfica que ejemplifica la tolerancia de una restricción de igualdad convertida a una restricción de desigualdad

una tolerancia dinámica resulta ser bastante útil. Una configuración dinámica para el parámetro  $\psi$  fue propuesta en [4], es adoptada en este trabajo y se define usando la ecuación.

$$\psi(t+1) = \frac{\psi(t)}{dec} \quad (4.2)$$

Donde  $t$  es el ciclo actual y  $dec$  es el valor decreciente para cada ciclo ( $dec > 1$ ). El objetivo es iniciar con una zona factible mayor a la original. Conforme pasen los ciclos del algoritmo, la tolerancia es reducida de manera que la violación de restricciones de las soluciones obtenidas al final de la ejecución del algoritmo es menor que las calculadas en los primeros ciclos.

En nuestro algoritmo que será utilizado para resolver un conjunto de problemas de prueba, el valor inicial de  $\psi$  es 1.0, el valor final de la tolerancia es de 0.0001 y el valor decreciente ( $dec$ ) es 1.00299. Estos valores se obtuvieron mediante experimentos preliminares.

### 4.3. Generación de la fuente candidata

De acuerdo con [4] para producir fuentes candidatas (tanto para abejas empleadas como observadoras) se usa la Ecuación (4.3):

$$v_{i,j}^g = \begin{cases} x_{i,j}^g + \phi \cdot (x_{i,j}^g - x_{k,j}^g) & , \text{ if } R_j < MR \\ x_{i,j}^g & , \text{ de otra manera} \end{cases} \quad (4.3)$$

Donde  $x_{i,j}^g$  representa la solución en la cual la abeja se encuentra,  $x_{k,j}^g$  es una fuente de alimento seleccionada al azar pero diferente de  $x_{i,j}^g$ ,  $i \in \{1,2,\dots,SN\}$ ,  $j \in \{1,2,\dots,n\}$  y  $\phi$ , es un valor real aleatorio generado dentro del rango  $[-1,1]$ . A diferencia del ABC tradicional donde se genera un valor de  $\phi$  para cada variable, en esta propuesta el valor se mantiene fijo para todas las variables de una solución (enfoque de "Static  $\phi$ ").  $MR$  es el porcentaje probabilístico que define si la variable  $j$  debe ser modificada, en nuestra implementación  $MR$  se establece con un valor de 0.8.

La figura 4.2 muestra gráficamente el operador de las abejas empleadas y observadoras.

### 4.4. Operador de la abeja exploradora

Se ha reportado en la literatura especializada sobre ABC que para problemas de optimización numérica con restricciones resulta difícil para la abeja exploradora generar una buena nueva solución

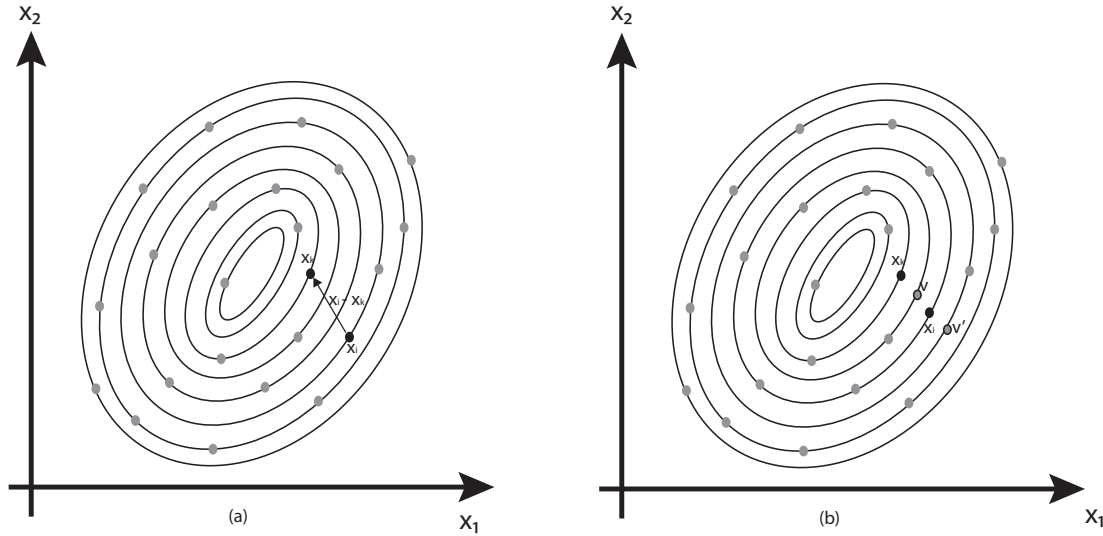


Figura 4.2: La figura **a** muestra el vector resultante de  $x_k$  y  $x_j$ . La figura **b** muestra las posibles soluciones candidatas obtenidas con 4.3 para cuando  $\phi$  adquiere un valor positivo o negativo

desde el enfoque del “azar”. Una posible solución a este problema es elaborar la búsqueda cerca del espacio de soluciones factibles. Así una modificación originalmente propuesta para trabajar con la optimización mediante cúmulos de partículas es adaptada en [4] dentro de SM-ABC para el funcionamiento de la abeja. En lugar de generar una solución al azar, se genera una solución  $v_i^g$  utilizando la fuente de alimento a ser remplazada  $x_i^g$  como base, para generar una nueva fuente de alimento con dirección a la mejor solución encontrada hasta el momento  $x_B^g$ , y una fuente de alimento seleccionada al azar  $x_k^g$  pero diferente a  $x_i^g$  como indica la ecuación 4.4.

$$v_{i,j}^g = x_{i,j}^g + \phi \cdot (x_{i,j}^g - x_{k,j}^g) + (1 - \phi) \cdot (x_{B,j}^g - x_{i,j}^g) \quad (4.4)$$

El objetivo de este operador para la abeja exploradora es que si la mejor solución se encuentra dentro de una zona no factible, cabe la posibilidad de que la nueva fuente de alimento generada se encuentre dentro de el área factible. Por otra parte, si la mejor solución se encuentra dentro de la zona factible la nueva fuente de alimento permitirá explorar dentro de la zona factible.

La figura 4.3 muestra gráficamente el funcionamiento modificado en SM-ABC de la abeja exploradora.

## 4.5. Manejo de límites

En el caso de que los operadores definidos para las abejas generen soluciones fuera de los límites permitidos, se utilizará el mecanismo utilizado en la Ecuación (4.5) basado en el trabajo de Kukkonen y Lampinen [14].

$$x_{i,j}^g = \begin{cases} 2 \cdot L_i^g - x_{i,j}^g & , \text{ if } x_{i,j} < L_i \\ 2 \cdot U_i^g - x_{i,j}^g & , \text{ if } x_{i,j} > U_i \\ x_{i,j}^g & , \text{ de otra manera} \end{cases} \quad (4.5)$$

En donde  $x_{i,j}^g$  es la variable  $i$  de la fuente de alimento  $j$ ,  $L_i$  es el límite inferior de la variable  $i$ , y  $U_i$  es el límite superior de la variable  $i$ . Es notable que algunas veces las soluciones generadas por las abejas exploradoras, observadoras y empleadas llegan a ser generadas fuera del área de búsqueda, por lo que la ecuación (4.5) mantiene las soluciones dentro del rango permitido y definido por el

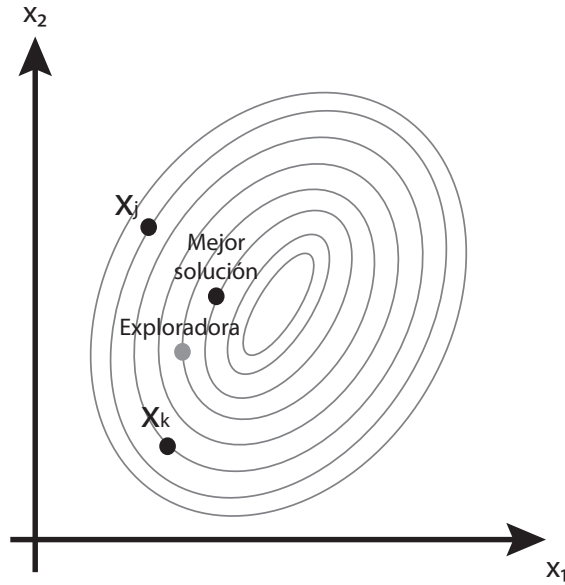


Figura 4.3: Figura que muestra un ejemplo del mecanismo de la abeja exploradora

programa mismo, además funciona para mantener diversidad de soluciones de manera más eficiente que en el ABC original.

## 4.6. Mecanismo de selección para manejo de restricciones

En nuestra implementación a diferencia del SM-ABC, en el mecanismo de las abejas observadoras para el proceso de selección de fuentes de alimento a mejorar, se hace uso de una serie de reglas basadas en el enfoque del “ $\epsilon$  – constraint” originalmente propuesto por Takahama y Sakai [25]. El objetivo de  $\epsilon$  – constraint es el transformar el problema original de optimización numérica con restricciones en un problema de optimización sin restricciones usando la suma de violación de restricciones ( $\Phi$ ) de una solución dada (ver Ecuación 4.6).

$$\Phi(x_k^g) = \sum_{i=1}^m \max(0, g_i(x)) + \sum_{j=1}^p \max(0, |h_j(x)| - \psi) \quad (4.6)$$

En donde la primera sumatoria calcula la violación total de las restricciones de desigualdad ( $g_i(x)$ ) y la segunda sumatoria calcula la violación total de las restricciones de igualdad ( $h_j(x)$ ) determinada por la tolerancia  $\psi$ .

La tolerancia  $\epsilon$  nos permite comparar entre dos soluciones  $x_i^g$  y  $x_j^g$  utilizando la función objetivo  $f(x_i^g)$  y  $f(x_j^g)$  cuando el valor de  $\Phi$  en ambas soluciones  $\Phi(x_i^g)$  y  $\Phi(x_j^g)$  se encuentran dentro de la tolerancia definida por  $\epsilon$ , incluso cuando ambos  $\Phi$  son iguales como se encuentra definido en la Ecuación 4.7.

$$x_i^g \leq_{\epsilon} x_j^g \Leftrightarrow \begin{cases} f(x_i^g) \leq f(x_j^g) & , \text{ if } \Phi(x_i^g), \Phi(x_j^g) < \epsilon \\ f(x_i^g) \leq f(x_j^g) & , \text{ if } \Phi(x_i^g) = \Phi(x_j^g) \\ \Phi(x_i^g) \leq \Phi(x_j^g) & , \text{ de otra manera} \end{cases} \quad (4.7)$$

De esta forma, la comparación se hace entre soluciones ligeramente no factibles utilizando el valor de su función objetivo, promoviendo así la conservación de las soluciones localizadas dentro áreas prometedoras cercanas a los límites de la zona factible. El valor de  $\epsilon$  varía conforme avanza el algoritmo en sus iteraciones como se muestra en la Ecuación (4.8).

$$\epsilon(0) = \Phi(x_B^0)$$

$$\epsilon(g) = \begin{cases} \epsilon(0) \cdot \left(1 - \frac{g}{gc}\right)^{cp} & , \text{ if } g < gc \\ 0 & , \text{ de otra manera} \end{cases} \quad (4.8)$$

Donde  $g$  es el número de ciclo,  $gc$  es el ciclo en el que  $\epsilon$  se convertirá en cero y  $cp$  es el parámetro de control de la velocidad con que se reducirá el valor de la tolerancia. El valor inicial de  $\epsilon$  corresponde a la suma de violación de restricciones de la mejor solución encontrada en la población inicial. Para efectos de nuestra implementación  $g$  es inicializado en 1,  $gc$  es el 20% del total del número de ciclos ( $MCN$ ) y  $cp$  es ajustado a 46 (valores altos son preferidos).

La Figura 4.4 muestra gráficamente el funcionamiento de la tolerancia  $\epsilon$  donde una solución ligeramente no factible  $V'$  se puede considerar como factible y compararse mediante su valor de la función objetivo.

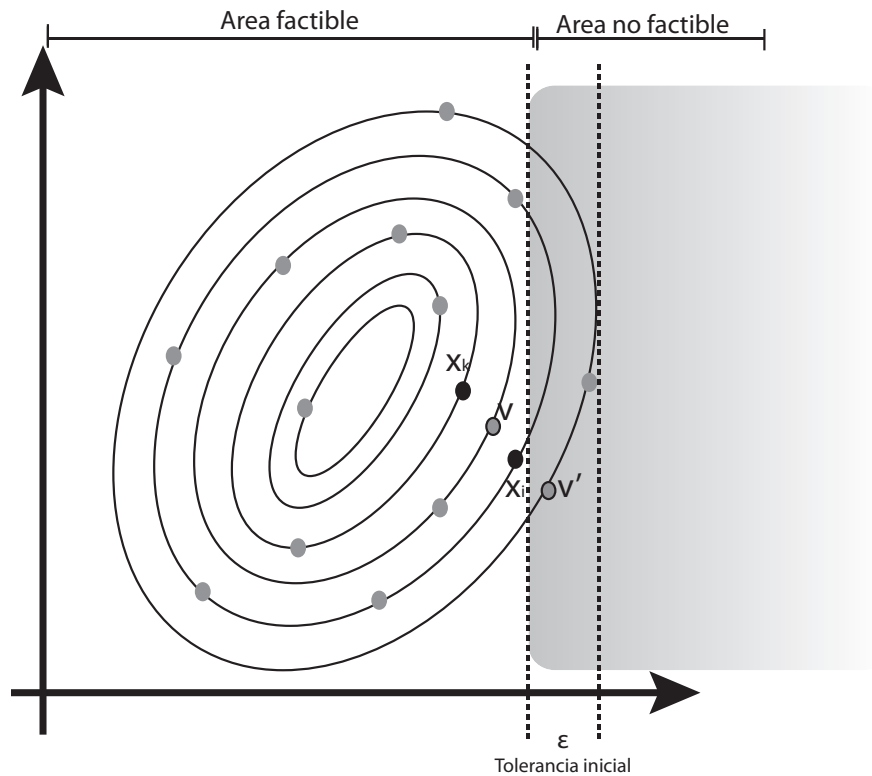


Figura 4.4: Figura que muestra el espacio de tolerancia en regiones no factibles establecidas por las restricciones del problema a resolver.

## 4.7. Límite de una solución

En la versión original del algoritmo ABC para optimización sin restricciones [4] el parámetro *limit* es calculado usando la fórmula  $SN * D$ , donde  $SN$  es el número de soluciones y  $D$  es el número de variables del problema.

La fórmula para calcular el parámetro *limit* tiene la propiedad de que varía de un problema a otro dependiendo de la dimensión del problema. En el caso de los problemas de Benchmark usados en este trabajo de tesis, el número de variables va de 2 a 24 variables, por lo tanto el valor de *limit*



va de 40 a 480 ciclos, resultando que en algunos problemas la abeja exploradora sea en muchas o pocas ocasiones utilizada.

El exceso de uso de la abeja exploradora en el algoritmo puede resultar en una convergencia prematura atascando los resultados en óptimos locales, por otra parte, el uso carente de la abeja exploradora reduce el posible impacto que puede tener en el mecanismo de búsqueda. Por estos motivos y de la misma manera que lo hace el algoritmo SM-ABC [4], se utiliza la siguiente fórmula para calcular la variable *limit* :  $limit = \frac{MCN}{(2*SN)}$  donde *MCN* es el número máximo de ciclos y  $2*SN$  es el número de individuos en la colonia (número de empleadas y observadoras juntas). Así el parámetro *limit* es independiente del problema dándole el mismo valor para todos los problemas que se ejecuten bajo el mismo ajuste de parámetros.

## 4.8. El algoritmo Static Phi Epsilon Constraint Artificial Bee Colony (SPEC-ABC)

Nuestro algoritmo propuesta tiene los mismos parámetros que el algoritmo SM-ABC, como se muestra en el cuadro 4.1.

Cuadro 4.1: Parámetros del SPEC-ABC

Nombre	Símbolo	Descripción
Número de soluciones	<i>SN</i>	Número de soluciones (fuentes de alimento)
Número de ciclos	<i>MCN</i>	Número de ciclos (iteraciones) que ejecutará el ABC
Límite	<i>limit</i>	Número de ciclos que se conservará una solución sin mejora antes de ser reemplazada por una generada por alguna abeja exploradora
Rango de modificacion	<i>MR</i>	Controla si la variable se modifica o no cuando la solución candidata es generada

El algoritmo propuesto ya completo se muestra en la figura 4.8. Éste inicia creando las fuentes de alimento y la población en el nido. Se evalúan cada una de las soluciones. Si el problema contiene restricciones de igualdad, el valor inicial de la tolerancia  $\psi$  se establece en 1.0.

Figura 4.5: Pseudocódigo del Static Phi Epsilon Constraint ABC (SPEC-ABC)

```

1  BEGIN
2  Inicializar las fuente de alimento y la población  $x_i^0$ ,  $i = 1, \dots, SN$ 
3  Evaluar para cada  $x_i^0$ ,  $i = 1, \dots, SN$ 
4   $g = 1$ 
5  IF Existen restricciones de igualdad
6  Inicializar  $\psi = 1.0$ 
7  END IF
8  REPEAT
9  Calcular el valor de  $\epsilon$  usando la Ecuación (4.8)
10 IF Existen restricciones de igualdad
11 Evaluar para cada  $x_i^0$ ,  $i = 1, \dots, SN$  contra el valor
    de la nueva tolerancia  $\psi$ 
12 END IF
13 FOR  $i = 1$  TO  $SN$ 
14 Generar  $v_i^g$  con  $x_i^{g-1}$  usando la Ecuación (4.3)
15 Evaluar  $v_i^g$ 
16 IF  $v_i^g$  es mejor que  $x_i^{g-1}$  (basado en la ecuación 4.7)
17  $x_i^g = v_i^g$ 
18 ELSE
19  $x_i^g = x_i^{g-1}$ 
20 END FOR
21 FOR  $i = 1$  TO  $SN$ 
22 Seleccionar, con base en la selección por torneo
    la fuente de alimento  $x_l^g$  usando la Ecuación (4.7)
23 Generar  $v_l^g$  con  $x_l^g$  usando la Ecuación (4.3)
24 Evaluar  $v_l^g$ 
25 IF  $v_l^g$  es mejor que  $x_l^g$  (basado en la Ecuación 4.7)
26  $x_l^g = v_l^g$ 
27 END IF
28 END FOR
29 Aplicar el uso de la abeja exploradora (4.4) para aquellas soluciones
    en los cuales el límite para ser mejoradas se haya alcanzado
30 Conservar la mejor solución hasta ahora
31 IF Existen restricciones de igualdad
32 Decrementar  $\psi$  usando la Ecuación (4.2)
33 END IF
34  $g = g + 1$ 
35 UNTIL  $g = MCN$ 
36 END

```

Basado en el enfoque  $\epsilon$ -constraint se calcula el valor de  $\epsilon$  utilizando la Ecuación 4.8. Si existen restricciones de igualdad se evalúa cada función para la nueva tolerancia  $\psi$ . Las abejas empleadas generan una nueva solución utilizando la Ecuación 4.3, se evalúa la solución generada y con base en el enfoque  $\epsilon$ -constraint se selecciona la fuente de alimento más apta utilizando la Ecuación 4.7. A este proceso se le conoce como el proceso de “exploración” del algoritmo donde todas las fuentes de alimento son visitadas y se intentan mejorar. Posteriormente se pasa al proceso de “explotación” donde se seleccionan las mejores fuentes de alimento por medio de torneos binarios determinísticos (aleatoriamente se toman dos fuentes de alimento y se escoge la mejor de ellas). Una vez que la abeja observadora haya seleccionado una fuente se procede a crear una nueva fuente de alimento utilizando la ecuación 4.3 y se selecciona la fuente de alimento más apta utilizando la ecuación 4.7. Si existe alguna fuente de alimento que haya alcanzado su límite y no se ha podido mejorar, se utiliza una abeja exploradora para buscar una nueva fuente de alimento utilizando la ecuación 4.4. Se graba la mejor solución existente hasta el momento. Si existen restricciones de igualdad se decrementa la tolerancia  $\psi$  utilizando la ecuación 4.2. Finalmente se repite nuevamente el proceso hasta que se hayan alcanzado el número máximo de ciclos (MCN).

Como puede notarse, los cambios de SPEC-ABC sobre SM-ABC se resumen a el manejo de restricciones usando  $\epsilon$ -constraint en lugar de las reglas de Deb que usa SM-ABC y el mantener estático del valor de  $\phi$  al usar el operador para generar nuevas soluciones por parte de las abejas empleadas y observadoras.

# Capítulo 5

## Experimentos y resultados

### 5.1. Diseño Experimental

Para probar el desempeño de nuestra versión del ABC usando la técnica  $\epsilon$  – *constraint* para el manejo de restricciones se realizaron 4 experimentos. Se utilizaron 24 funciones de prueba bien conocidas en la literatura especializada [16]. Los detalles de cada función se encuentran en un apéndice al final de este documento y las características de cada una de ellas se resumen en la Tabla 5.1.

Cuadro 5.1: Resumen con las principales características de los problemas a resolver

Función	$n$	Tipo de función	$p$	LI	NI	LE	NE
g01	13	cuadrática	0.0003 %	9	0	0	0
g02	20	no lineal	99.9973 %	2	0	0	0
g03	10	no lineal	0.0026 %	0	0	0	1
g04	5	cuadrática	27.0079 %	4	2	0	0
g05	4	no lineal	0.0000 %	2	0	0	3
g06	2	no lineal	0.0057 %	0	2	0	0
g07	10	cuadrática	0.0000 %	3	5	0	0
g08	2	no lineal	0.8581 %	0	2	0	0
g09	7	no lineal	0.5199 %	0	4	0	0
g10	8	lineal	0.0020 %	6	0	0	0
g11	2	cuadrática	0.0973 %	0	0	0	1
g12	3	cuadrática	4.7697 %	0	1	0	0
g13	5	no lineal	0.0000 %	0	0	1	2
g14	10	no lineal	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

En la Tabla 5.1  $n$  es el número de variables del problema,  $LI$  Es el número de restricciones lineales de desigualdad,  $NI$  es el número de restricciones no lineales de desigualdad,  $LE$  es el número de

restricciones lineales de igualdad,  $NE$  es el número de restricciones no lineales de igualdad y  $\rho$  es la medida para encontrar el tamaño aproximado la zona factible con respecto a todo el espacio de búsqueda con:

$$\rho = \frac{|F|}{|S|} \quad (5.1)$$

Donde  $|F|$  es el número de soluciones factibles y  $|S|$  es el número total de soluciones generadas de manera aleatoria. Michalewicz y Shoenauer [21] sugieren un total de 1,000,000 de soluciones para  $|S|$ .

En primer lugar se realizó una comparación indirecta del ABC original [12], el SM-ABC propuesto en [4] y nuestra propuesta SPEC-ABC donde los resultados fueron obtenidos de las publicaciones correspondientes. Posteriormente se realizó una comparación directa donde se implementaron las 3 versiones y obtuvieron los resultados mostrados en este trabajo, de igual manera se realizó una comparación directa de medidas de desempeño para finalmente realizar una comparación indirecta con algoritmos del estado del arte.

Los 4 experimentos mencionados anteriormente son detallados a continuación:

- I. Se realiza una comparación indirecta entre el ABC [12], SM-ABC [4] y nuestra propuesta SPEC-ABC. Fueron comparadas a partir de 30 ejecuciones independientes por cada algoritmo la mejor solución (best), solución media (mean), peor solución (worst) y desviación estándar (Std. Dev.) en las 24 funciones (g01-g24) de benchmark de las cuales los primero 13 resultados del ABC se obtuvieron de los resultados publicados en [13] y los 24 del SM-ABC (g01-g24) así como los restantes 11 del ABC (g14-g24) se obtuvieron en [4].
- II. Se realizó una comparación directa mediante la implementación de los algoritmos ABC, SM-ABC contra nuestra propuesta SPEC-ABC. Los parámetros comunes fueron tomados de [4] como la tolerancia adaptativa para restricciones de igualdad, la selección de la fuente de alimento mediante torneo el  $MCN = 5800$ , el comportamiento de la abeja scout el operador de cruza y el manejo de los límites. El mejor resultado, resultado promedio y peor resultado así como la desviación estándar fueron comparados a partir de 30 ejecuciones independientes por cada algoritmo en las 24 funciones de benchmark (g01-g24). Cada medida de desempeño se explicará más adelante en este capítulo.
- III. Se realizó una comparación directa de medidas de desempeño a través de la implementación de los algoritmos ABC, SM-ABC y nuestra implementación SPEC-ABC en las 24 funciones de benchmark (g01-g24)
- IV. Una comparación indirecta fue desarrollada contra los resultados publicados de 3 algoritmos del estado del arte. Éstos son:
  - **SAPF-GA** por sus siglas en ingles *Self Adaptive Penalty Function Genetic Algorithm* propuesto en [26]. Éste implementa una función de penalización adaptativa para manejo de restricciones que no necesita modificación alguna de los valores de penalización.
  - **HCOEA** por sus siglas en ingles *Hybrid Constrained Optimization Evolutionary Algorithm* propuesto en [27]. Éste es un algoritmo evolutivo que combina componentes de optimización multi-objetivo para resolver problemas de optimización con restricciones mono-objetivo.
  - **ATMES** por sus siglas en ingles *Adaptive Tradeoff Model Evolutionary Strategy* propuesto en [28]. Éste incluye una variedad de esquemas de asignación de aptitudes dependiendo de la factibilidad de las soluciones en la población de un algoritmo evolutivo.

Se comparan el mejor resultado, resultado promedio y peor resultado así como la desviación estándar para 13 funciones de benchmark (g01-g13).

Estos 3 algoritmos fueron seleccionados por mostrar resultados competitivos así como por haber sido publicados en revistas especializadas en el área de optimización usando algoritmos evolutivos y de inteligencia colectiva.

### 5.1.1. Parámetros de nuestro algoritmo

Para realizar la comparación de resultados con los demás algoritmos los valores comunes entre las diferentes versiones del ABC fueron tomadas de [4]; además, se agregan los parámetros del enfoque  $\epsilon$  – *constraint* propios de nuestra implementación mostrados en la tabla 5.2. Un total de 30 corridas fueron ejecutadas por cada problema usando diferentes semillas obtenidas de manera aleatoria.

Cuadro 5.2: Parámetros del algoritmo propuesto

Parámetro	Símbolo	Valor
Número de soluciones	$SN$	20
Número máximo de ciclos	$MCM$	5800
Limite	$limit$	$MCM/(2 * SN) = 145$
Modification Rate	$MR$	0.8
Ciclo en que se $\epsilon$ se convertirá en cero	$gc$	$MCM * 0,2 = 1160$
Control de velocidad para reducir la tolerancia de $\epsilon$	$cp$	46

El número total de evaluaciones que realizó el algoritmo dentro de cada ejecución independiente estuvo entre  $232000 < \text{evaluaciones} < 240000$  incluso en los problemas con restricciones de igualdad que requieren un recálculo de la solución hasta el momento encontrada en cada ciclo cuando se reajusta la tolerancia.

### 5.1.2. Medidas de Desempeño

En esta sección se describen las medidas de desempeño utilizadas en este trabajo de tesis y algunos conceptos básicos para dar noción de ellas y así favorecer el entendimiento de los resultados [4].

#### Conceptos Básicos:

- **Evaluación:** Se refiere al cálculo de la función objetivo y de las restricciones para una solución. El número total de evaluaciones es una importante medida para determinar el costo computacional de un algoritmo como el propuesto en este trabajo de tesis.
- **Solución factible:** Se refiere a una solución que cumple con todas las restricciones del problema.
- **Ejecución factible:** Se refiere a una ejecución donde al menos se genera una solución factible para el problema.
- **Solución exitosa:** Se refiere a una solución factible que es igual o mejor que la mejor solución conocida para un determinado problema. Para ello, la diferencia entre los valores objetivo de la solución conocida debe ser mayor que 0.0001.
- **Ejecución exitosa:** Se refiere a una ejecución del problema donde se genera al menos una solución exitosa.

#### Medidas de desempeño

- **AFES:** Es la cantidad promedio de evaluaciones requeridas para encontrar una solución exitosa. Ésta es calculada con la fórmula  $\frac{\sum \text{evaluaciones necesarias para encontrar una solución exitosa}}{\text{número de ejecuciones exitosas}}$ . Para esta medida de rendimiento un valor bajo es preferible.
- **P:** Es el porcentaje de ejecuciones exitosas, calculada por la fórmula  $\frac{\text{número de ejecuciones exitosas}}{\text{número total de ejecuciones}}$ . En esta medida de rendimiento un alto valor es preferible.

- **SP:** Esta es calculada con la fórmula  $\frac{AFES}{P}$ . En esta medida de rendimiento un valor bajo es preferible.
- **FP:** Es el porcentaje de ejecuciones factibles, calculada con la fórmula  $\frac{\text{número de ejecuciones factibles}}{\text{número total de ejecuciones}}$ . En esta medida de desempeño un alto valor es preferible.
- **EVALS:** Propuesto en [15], el objetivo es calcular el número de evaluaciones necesarias para encontrar la primera solución factible en cada corrida del algoritmo. En esta medida de rendimiento se reporta el mejor resultado, resultado promedio, peor valor y la desviación estándar de un conjunto de ejecuciones independientes. Para esta medida de desempeño un valor bajo es preferible.
- **PROGRESS RATIO:** Propuesta en [19], el objetivo es medir la capacidad de mejora del algoritmo dentro de la zona factible. En esta medida de rendimiento un alto valor es preferible. Este es calculado con la Fórmula:

$$\text{PROGRESS RATIO} = \left| \ln \sqrt{\frac{f_{\min}(G_{ff})}{f_{\min}(T)}} \right| \quad (5.2)$$

Donde  $f_{\min}(G_{ff})$  es el valor de la función objetivo de la primera solución factible encontrada y  $f_{\min}(T)$  es el valor de la función objetivo de la mejor solución factible encontrada. Para nuestra implementación hemos usado como se sugiere en [4] una modificación de la fórmula 5.2 para evitar problemas en el cálculo de esta medida de desempeño.

$$\text{PROGRESS RATIO} = \begin{cases} \left| \ln \sqrt{\frac{f_{\min}(G_{ff})}{f_{\min}(T)}} \right| & , \text{if } f_{\min}(T) > 0 \\ \left| \ln \sqrt{\frac{f_{\min}(G_{ff})+1}{f_{\min}(T)+1}} \right| & , \text{if } f_{\min}(T) = 0 \\ \left| \ln \sqrt{\frac{f_{\min}(G_{ff})+2*|f_{\min}(T)+2|}{f_{\min}(T)+2*|f_{\min}(T)+2|}} \right| & , \text{if } f_{\min}(T) < 0 \end{cases} \quad (5.3)$$

En esta medida de desempeño se reporta el mejor resultado, resultado promedio, peor valor y la desviación estándar de un conjunto de ejecuciones independientes.

## 5.2. Comparativo indirecto

Los resultados estadísticos (mejor, media, peor y desviación estándar) para el comparativo indirecto entre ABC, SM-ABC y SPEC-ABC se presentan en los Cuadros 5.3 y 5.4.

- **Mejor resultado:** Nuestro algoritmo propuesto alcanzó una mejor solución en 7 problemas en comparación con las otras dos versiones del ABC en los problemas: g09, g10, g13, g18, g19, g21 y g23, el ABC obtuvo un mejor resultado en el problemas g05 y el SM-ABC obtuvo mejores resultado en los problemas g14, g15, g17. En los problemas g01, g03, g04, g06, g07, g08, g11, g12, g16 y g24 se alcanzó igualar al mejor resultado obtenido por los algoritmos o al mejor conocido hasta el momento. Ninguno de los 3 algoritmos (ABC, SM-ABC y SPEC-ABC) alcanzaron una solución factible para los problemas g20 y g22.
- **Resultado promedio:** Nuestro algoritmo propuesto alcanzó una mejor solución en 8 problemas en comparación con las otras dos versiones del ABC en los problemas: g05, g10, g13, g15, g17, g19, g21, g23, el ABC obtuvo mejores resultados en los problemas g09 y g18 y el SM-ABC obtuvo mejores resultados en los problemas g02, g07, g14. En los problemas g03, g04, g06, g08, g11, g12, g16 y g23 se alcanzó igualar al mejor resultado obtenido por alguna implementación del ABC o al mejor conocido hasta el momento.
- **Peor resultado:** Nuestro algoritmo propuesto alcanzó una mejor solución en 6 problemas en comparación con las otras dos versiones del ABC en los problemas: g05, g10, g15, g17, g19 y g21 el ABC obtuvo mejores resultados en los problemas g09 y g18 y el SM-ABC obtuvo mejores resultados en los problemas g02, g07, g13, g14 y g23. En los problemas g03, g04, g06, g08, g11, g12, g16 y g23 se alcanzó igualar al mejor resultado obtenido por alguna implementación del ABC o al mejor conocido hasta el momento.
- **Desviación estándar:** Nuestro algoritmo propuesto alcanzó una mejor solución en 8 problemas en comparación con las otras dos versiones del ABC en los problemas: g05, g10, g17 y g19 el ABC obtuvo mejores resultados en los problemas g04, g09, g11, g13 y g18 y el SM-ABC obtuvo mejores resultados en los problemas g02, g07, g14 y g21. En los problemas g03, g04, g06, g08, g11, g12, y g24 se alcanzó igualar al mejor resultado obtenido por alguna implementación del ABC o al mejor conocido hasta el momento.

### Discusión

En este experimento desarrollado nuestra propuesta fue mejor o igual que el ABC o el SM-ABC con respecto al mejor resultado en 17 de los 24 problemas de benchmark. Con respecto al peor resultado, el SPEC-ABC mostró igual o mejores resultados que el ABC o el SM-ABC en 14 de los 24 problemas. Esta nueva propuesta ha demostrado que mejora considerablemente en problemas con restricciones de igualdad y aquellos considerados complejos como g21 y g23.

Cuadro 5.3: Comparación indirecta de los primeros 13 problemas entre el algoritmo ABC de Dervis Karaboga, SM-ABC de Omar Cetina y SPEC-ABC. Los mejores resultados se enmarcan en negritas.

Función/ Óptimo		Methods		
		ABC	SM-ABC	SPEC-ABC
<b>g01</b> <b>-15</b>	Best	<b>-15</b>	<b>-15</b>	<b>-15</b>
	Mean	<b>-15</b>	<b>-15</b>	-14.13
	Worst	<b>-15</b>	<b>-15</b>	-12.45
	Std. Dev.	<b>0.00E+00</b>	<b>0.00E+00</b>	9.16E-01
<b>g02</b> <b>-0.803619</b>	Mejor	-0.803598	<b>-0.803615</b>	-0.709034
	Media	-0.792412	<b>-0.799336</b>	-0.471210
	Worst	-0.749797	<b>-0.777438</b>	-0.319046
	Std. Dev.	1.20E-02	<b>6.84E-03</b>	1.05E-01
<b>g03</b> <b>-1</b>	Best	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Mean	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Worst	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Std. Dev.	<b>0.00E+00</b>	<b>4.68E-05</b>	<b>2.73E-05</b>
<b>g04</b> <b>-30665.539</b>	Best	<b>-30665.539</b>	<b>-30665.539</b>	<b>-30665.539</b>
	Mean	<b>-30665.539</b>	<b>-30665.539</b>	<b>-30665.539</b>
	Worst	<b>-30665.539</b>	<b>-30665.539</b>	<b>-30665.539</b>
	Std. Dev.	<b>0.00E+00</b>	2.22E-11	<b>0.00E+00</b>
<b>g05</b> <b>5126.498</b>	Best	<b>5126.484</b>	5126.736	5126.496
	Mean	5185.714	5178.139	<b>5126.526</b>
	Worst	5438.387	5317.195	<b>5126.859</b>
	Std. Dev.	7.54E+01	5.61E+01	<b>7.98E-02</b>
<b>g06</b> <b>-6961.814</b>	Best	<b>-6961.814</b>	<b>-6961.814</b>	<b>-6961.814</b>
	Mean	-6961.813	<b>-6961.814</b>	<b>-6961.814</b>
	Worst	-6961.805	<b>-6961.814</b>	<b>-6961.814</b>
	Std. Dev.	2.00E-03	<b>0.00E+00</b>	<b>0.00E+00</b>
<b>g07</b> <b>24.306</b>	Best	24.33	<b>24.314</b>	<b>24.316</b>
	Mean	24.473	<b>24.4154</b>	24.657
	Worst	25.19	<b>24.853</b>	25.544
	Std. Dev.	1.86E-01	<b>1.24E-01</b>	3.29E-01
<b>g08</b> <b>-0.095825</b>	Best	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>
	Mean	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>
	Worst	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>
	Std. Dev.	<b>0.00E+00</b>	4.23E-17	<b>0.00E+00</b>
<b>g09</b> <b>680.63</b>	Best	680.634	680.631	<b>680.630</b>
	Mean	<b>680.640</b>	680.647	680.643
	Worst	<b>680.653</b>	680.691	680.857
	Std. Dev.	<b>4.00E-03</b>	1.55E-02	4.09E-02
<b>g10</b> <b>7049.248</b>	Best	7053.904	7051.706	<b>7049.547</b>
	Mean	7224.407	7233.882	<b>7116.934</b>
	Worst	7604.132	7473.108	<b>7362.639</b>
	Std. Dev.	1.34E+02	1.10E+02	<b>8.21E+01</b>
<b>g11</b> <b>0.75</b>	Best	<b>0.75</b>	<b>0.749</b>	<b>0.749</b>
	Mean	<b>0.75</b>	<b>0.749</b>	<b>0.749</b>
	Worst	<b>0.75</b>	<b>0.749</b>	<b>0.749</b>
	Std. Dev.	<b>0.00E+00</b>	2.30E-05	<b>0.00E+00</b>
<b>g12</b> <b>-1</b>	Best	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Mean	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Worst	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Std. Dev.	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
<b>g13</b> <b>0.05395</b>	Best	0.760000	0.053985	<b>0.053942</b>
	Mean	0.968000	0.158552	<b>0.263967</b>
	Worst	1	<b>0.442905</b>	1
	Std. Dev.	<b>5.50E-02</b>	1.73E-01	2.36E-01



Cuadro 5.4: Comparación indirecta de los últimos 11 problemas entre el algoritmo ABC de Dervis Karaboga, SM-ABC de Omar Cetina y SPEC-ABC. Los mejores resultados se enmarcan en negritas. ”Significa que no fueron encontradas soluciones factibles en ninguna ejecución independiente.

Funcion/ Optimal		Methods		
		ABC	SM-ABC	SPEC-ABC
<b>g14</b> <b>-47.764411</b>	Best	-	<b>-47.641</b>	-46.667
	Mean	-	<b>-47.270</b>	-46.468
	Worst	-	<b>-46.537</b>	-43.871
	Std. Dev.	-	<b>2.46E-01</b>	5.20E-01
<b>g15</b> <b>961.715172</b>	Best	967.517	<b>961.714</b>	961.715
	Mean	968.639	961.718	<b>961.715</b>
	Worst	972.317	961.793	<b>961.720</b>
	Std. Dev.	2.06E+00	1.42E-02	<b>1.59E-03</b>
<b>g16</b> <b>-1.905155</b>	Best	<b>-1.905</b>	<b>-1.905</b>	<b>-1.905</b>
	Mean	<b>-1.905</b>	<b>-1.905</b>	<b>-1.905</b>
	Worst	<b>-1.905</b>	<b>-1.905</b>	<b>-1.905</b>
	Std. Dev.	<b>4.52E-16</b>	<b>4.52E-16</b>	1.49E-14
<b>g17</b> <b>8876.98068</b>	Best	8959.545	<b>8866.618</b>	8927.597
	Mean	9002.701	8987.458	<b>8928.864</b>
	Worst	9025.873	9165.219	<b>8938.617</b>
	Std. Dev.	3.74E+01	9.57E+01	<b>3.12E+00</b>
<b>g18</b> <b>-0.865735</b>	Best	-0.865677	-0.866006	<b>-0.866025</b>
	Mean	<b>-0.859821</b>	-0.795019	-0.740724
	Worst	<b>-0.771239</b>	-0.672216	-0.500000
	Std. Dev.	<b>1.69E-02</b>	-8.66E-01	1.40E-01
<b>g19</b> <b>32.655593</b>	Best	35.212	33.285	<b>32.662</b>
	Mean	36.481	34.266	<b>33.107</b>
	Worst	37.771	35.746	<b>34.914</b>
	Std. Dev.	6.64E-01	6.31E-01	<b>5.11E-01</b>
<b>g20</b> <b>0.188446</b>	Best	-	-	-
	Mean	-	-	-
	Worst	-	-	-
	Std. Dev.	-	-	-
<b>g21</b> <b>193.7783493</b>	Best	277.156	266.500	<b>193.724</b>
	Mean	454.137	306.609	<b>270.758</b>
	Worst	1000	329.960	<b>328.751</b>
	Std. Dev.	2.55E+02	<b>1.98E+01</b>	5.30E+01
<b>g22</b> <b>382.902205</b>	Best	-	-	-
	Mean	-	-	-
	Worst	-	-	-
	Std. Dev.	-	-	-
<b>g23</b> <b>-400.0025</b>	Best	-0.007	-159.738	<b>-350.125</b>
	Mean	-0.001	-35.271	<b>-121.373</b>
	Worst	-0.001	<b>109.010</b>	276.002
	Std. Dev.	<b>1.97E-03</b>	<b>8.28E+01</b>	1.58E+02
<b>g24</b> <b>-5.508013</b>	Best	<b>-5.508</b>	<b>-5.508</b>	<b>-5.508</b>
	Mean	<b>-5.508</b>	<b>-5.508</b>	<b>-5.508</b>
	Worst	<b>-5.508</b>	<b>-5.508</b>	<b>-5.508</b>
	Std. Dev.	<b>0.00E+00</b>	2.71E-15	<b>0.00E+00</b>

### 5.3. Comparativo directo

Los resultados estadísticos (mejor, media, peor y desviación estándar) para el comparativo directo entre ABC, SM-ABC y SPEC-ABC se presentan en los Cuadros 5.5 y 5.6.

- **Mejor resultado:** Nuestro algoritmo propuesto alcanzó una mejor solución en 13 problemas en comparación con las otras dos versiones del ABC en los problemas: g03, g05, g07, g09, g10, g13, g14, g15, g17, g18, g19, g21 y g23, el ABC no pudo superar a alguna de las dos implementaciones y finalmente el SM-ABC obtuvo mejores resultado en el problema g02. En los problemas g01, g04, g06, g08, g11, g12, g16 y g24 se alcanzó igualar al mejor resultado obtenido por los algoritmos o al mejor conocido hasta el momento. Ninguno de los 3 algoritmos (ABC, SM-ABC y SPEC-ABC) alcanzaron una solución factible para los problemas g20 y g22.
- **Resultado promedio:** Nuestro algoritmo propuesto alcanzó una mejor solución en 11 problemas en comparación con las otras dos versiones del ABC en los problemas: g03, g05, g10, g13, g14, g15, g17, g18, g19, g21 y g23, el ABC obtuvo mejores resultados en los problemas g02 y g09 y el SM-ABC obtuvo mejores resultados en el problema g07. En los problemas g03, g04, g05, g06, g08, g10, g11, g12, g13, g14, g15, g16, g17, g18, g19, g21, g23, y g24 se alcanzó igualar al mejor resultado obtenido por alguna implementación del ABC o al mejor conocido hasta el momento.
- **Peor resultado:** Nuestro algoritmo propuesto alcanzó una mejor solución en 9 problemas en comparación con las otras dos versiones del ABC en los problemas: g03, g05, g10, g14, g15, g17, g19, g21 y g23, el ABC obtuvo mejores resultados en los problemas g07, g09 y g13 y el SM-ABC obtuvo mejores resultados en el problema g18. En los problemas g03, g04, g05, g08, g10, g11, g12, g14, g15, g16, g17, g19, g21, g23 y g24 se alcanzó igualar al mejor resultado obtenido por alguna implementación del ABC o al mejor conocido hasta el momento.
- **Desviación estándar:** Nuestro algoritmo propuesto alcanzó una mejor solución en 9 problemas en comparación con las otras dos versiones del ABC en los problemas: g03, g05, g10, g14, g15, g17, g19, g21 y g23 el ABC obtuvo mejores resultados en los problemas g02, g18 y g24 el SM-ABC obtuvo mejores resultados en los problemas g07, g09 y g13. En los problemas g03, g04, g05, g06, g08, g10, g11, g12, g14, g15, g17, g19, g21, g23 y g24 se alcanzó igualar al mejor resultado obtenido por alguna implementación del ABC o al mejor conocido hasta el momento.

#### Discusión

En este experimento desarrollado nuestra propuesta fue mejor o igual que el ABC o el SM-ABC con respecto al mejor resultado en 20 de los 24 problemas de benchmark. Con respecto al peor resultado, el SPEC-ABC mostró igual o mejores resultados que el ABC o el SM-ABC en 16 de los 24 problemas. Esta nueva propuesta ha demostrado que mejora considerablemente en problemas con restricciones de igualdad y, de nuevo, en problemas difíciles como g21 y g23.

Cuadro 5.5: Comparación directa ABC, SM-ABC and SPEC-ABC en las primeras trece funciones de prueba. Los mejores resultados se remarcan en negritas.

Funcion/ Optimal		Methods		
		ABC	SM-ABC	SPEC-ABC
<b>g01</b> <b>-15</b>	Best	<b>-15</b>	<b>-15</b>	<b>-15</b>
	Mean	<b>-15</b>	<b>-15</b>	-14.13
	Worst	<b>-15</b>	<b>-15</b>	-12.45
	Std. Dev.	<b>0.00E+00</b>	<b>0.00E+00</b>	9.16E-01
<b>g02</b> <b>-0.803619</b>	Best	-0.803604	<b>-0.803608</b>	-0.709034
	Mean	<b>-0.798295</b>	-0.795860	-0.471210
	Worst	<b>-0.771740</b>	<b>-0.771734</b>	-0.319046
	Std. Dev.	<b>8.05E-03</b>	9.16E-03	1.05E-01
<b>g03</b> <b>-1</b>	Best	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Mean	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Worst	-0.999	-0.999	<b>-1</b>
	Std. Dev.	1.52E-04	1.32E-04	<b>2.73E-05</b>
<b>g04</b> <b>-30665.539</b>	Best	<b>-30665.539</b>	<b>-30665.539</b>	<b>-30665.539</b>
	Mean	<b>-30665.539</b>	<b>-30665.539</b>	<b>-30665.539</b>
	Worst	<b>-30665.539</b>	<b>-30665.539</b>	<b>-30665.539</b>
	Std. Dev.	<b>0.00E+00</b>	1.12E-11	<b>0.00E+00</b>
<b>g05</b> <b>5126.498</b>	Best	5130.825	5126.569	<b>5126.496</b>
	Mean	5142.169	5356.145	<b>5126.526</b>
	Worst	5159.767	5585.721	<b>5126.859</b>
	Std. Dev.	1.55E+01	3.25E+02	<b>7.98E-02</b>
<b>g06</b> <b>-6961.814</b>	Best	<b>-6961.814</b>	<b>-6961.814</b>	<b>-6961.814</b>
	Mean	<b>-6961.814</b>	<b>-6961.814</b>	<b>-6961.814</b>
	Worst	<b>-6961.814</b>	<b>-6961.814</b>	<b>-6961.814</b>
	Std. Dev.	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
<b>g07</b> <b>24.306</b>	Best	24.343	24.339	<b>24.316</b>
	Mean	24.533	<b>24.509</b>	24.657
	Worst	<b>24.883</b>	24.919	25.544
	Std. Dev.	1.46E-01	<b>1.37E-01</b>	3.29E-01
<b>g08</b> <b>-0.095825</b>	Best	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>
	Mean	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>
	Worst	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>
	Std. Dev.	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
<b>g09</b> <b>680.63</b>	Best	680.633	680.633	<b>680.630</b>
	Mean	<b>680.640</b>	680.641	680.643
	Worst	<b>680.649</b>	680.655	680.857
	Std. Dev.	4.65E-03	<b>5.39E-03</b>	4.09E-02
<b>g10</b> <b>7049.248</b>	Best	7053.593	7049.654	<b>7049.547</b>
	Mean	7196.234	7212.830	<b>7116.934</b>
	Worst	7469.072	7847.558	<b>7362.639</b>
	Std. Dev.	1.16E+02	1.84E+02	<b>8.21E+01</b>
<b>g11</b> <b>0.75</b>	Best	<b>0.749</b>	<b>0.749</b>	<b>0.749</b>
	Mean	<b>0.749</b>	<b>0.749</b>	<b>0.749</b>
	Worst	<b>0.749</b>	<b>0.749</b>	<b>0.749</b>
	Std. Dev.	1.51E-06	2.08E-06	<b>0.00E+00</b>
<b>g12</b> <b>-1</b>	Best	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Mean	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Worst	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Std. Dev.	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
<b>g13</b> <b>0.05395</b>	Best	0.195139	0.442644	<b>0.053942</b>
	Mean	0.380739	0.466536	<b>0.263967</b>
	Worst	<b>0.445825</b>	0.521331	1
	Std. Dev.	1.24E-01	<b>3.71E-02</b>	2.36E-01

Cuadro 5.6: Comparación directa entre ABC, SM-ABC and SPEC-ABC en las últimas once funciones de prueba. Los mejores resultados se remarcan en negritas. "Significa que no fueron encontradas soluciones factibles en ninguna ejecución independiente.

Funcion/ Optimal		Methods		
		ABC	SM-ABC	SPEC-ABC
<b>g14</b> <b>-47.764411</b>	Best	-	-	<b>-46.667</b>
	Mean	-	-	<b>-46.468</b>
	Worst	-	-	<b>-43.871</b>
	Std. Dev.	-	-	<b>5.20E-01</b>
<b>g15</b> <b>961.715172</b>	Best	961.715	961.715	<b>961.715</b>
	Mean	962.474	963.075	<b>961.715</b>
	Worst	964.540	966.764	<b>961.720</b>
	Std. Dev.	1.22E+00	2.17E+00	<b>1.59E-03</b>
<b>g16</b> <b>-1.905155</b>	Best	<b>-1.905</b>	<b>-1.905</b>	<b>-1.905</b>
	Mean	<b>-1.905</b>	<b>-1.905</b>	<b>-1.905</b>
	Worst	<b>-1.905</b>	<b>-1.905</b>	<b>-1.905</b>
	Std. Dev.	<b>1.10E-15</b>	<b>1.10E-15</b>	1.49E-14
<b>g17</b> <b>8876.98068</b>	Best	8946.969	8947.715	<b>8927.597</b>
	Mean	8952.787	8947.715	<b>8928.864</b>
	Worst	8958.605	8947.715	<b>8938.617</b>
	Std. Dev.	8.23E+00	-	<b>3.12E+00</b>
<b>g18</b> <b>-0.865735</b>	Best	-0.865671	-0.865999	<b>-0.866025</b>
	Mean	-0.734206	-0.739236	<b>-0.740724</b>
	Worst	-0.668302	<b>-0.668373</b>	-0.500000
	Std. Dev.	<b>8.31E-02</b>	8.89E-02	1.40E-01
<b>g19</b> <b>32.655593</b>	Best	34.484	34.767	<b>32.662</b>
	Mean	36.087	35.945	<b>33.107</b>
	Worst	37.248	38.063	<b>34.914</b>
	Std. Dev.	8.12E-01	8.10E-01	<b>5.11E-01</b>
<b>g20</b> <b>0.188446</b>	Best	-	-	-
	Mean	-	-	-
	Worst	-	-	-
	Std. Dev.	-	-	-
<b>g21</b> <b>193.7783493</b>	Best	-	-	<b>193.724</b>
	Mean	-	-	<b>270.758</b>
	Worst	-	-	<b>328.751</b>
	Std. Dev.	-	-	<b>5.30E+01</b>
<b>g22</b> <b>382.902205</b>	Best	-	-	-
	Mean	-	-	-
	Worst	-	-	-
	Std. Dev.	-	-	-
<b>g23</b> <b>-400.0025</b>	Best	-	-	<b>-350.125</b>
	Mean	-	-	<b>-121.373</b>
	Worst	-	-	<b>276.002</b>
	Std. Dev.	-	-	<b>1.58E+02</b>
<b>g24</b> <b>-5.508013</b>	Best	<b>-5.508</b>	<b>-5.508</b>	<b>-5.508</b>
	Mean	<b>-5.508</b>	<b>-5.508</b>	<b>-5.508</b>
	Worst	<b>-5.508</b>	<b>-5.508</b>	<b>-5.508</b>
	Std. Dev.	<b>0.00E+00</b>	2.71E-15	<b>0.00E+00</b>

## 5.4. Comparativo con medidas de desempeño

Los resultados obtenidos para las primeras cuatro medidas de desempeño AFES, P, SP y FP; se presentan en los Cuadros 5.7 y 5.8.

### 5.4.1. Análisis de AFES, P, SP y FP

- **AFES:** Nuestro algoritmo propuesto alcanzó un mejor valor de AFES en 13 problemas en comparación con las otras dos versiones del ABC: g01, g03, g04, g05, g06, g08, g09, g11, g13, g16, g18, g21 y g24. El ABC no pudo superar a alguna de las dos implementaciones y finalmente el SM-ABC obtuvo mejores resultado en el problema g02, g12 y g15. En los problemas g01, g03, g04, g05, g06, g08, g09, g11, g13, g16, g18, g21 y g24 se alcanzó igualar al mejor resultado obtenido por los algoritmos.
- **P:** Nuestro algoritmo propuesto alcanzó un mejor valor de P en 6 problemas en comparación con las otras dos versiones del ABC: g05, g09, g13, g15, g18 y g21. El ABC obtuvo un mejor resultados en el problemas g02 y el SM-ABC no obtuvo mejores resultados que las otras dos versiones del algoritmo. En los problemas g03, g04, g05, g06, g08, g09, g11, g12, g13, g15, g16, g18, g21 y g24 se alcanzó igualar al mejor resultado obtenido por alguna implementación del ABC.
- **SP:** Nuestro algoritmo propuesto alcanzó un mejor valor de SP en 13 problemas en comparación con las otras dos versiones del ABC: g03, g04, g05, g06, g08, g09, g11, g13, g15, g16, g18, g21 y g24. El ABC obtuvo un mejor resultado en el problema g02 y el SM-ABC obtuvo mejores resultados en los problema g01 y g12. En los problemas g03, g04, g05, g06, g08, g09, g11, g13, g15, g16, g18, g21 y g24 se alcanzó igualar al mejor resultado obtenido por alguna implementación del ABC .
- **FP:** Nuestro algoritmo propuesto alcanzó un mejor valor de FP en 7 problemas en comparación con las otras dos versiones del ABC: g05, g13, g14, g15, g17, g21 y g23. El ABC no obtuvo mejores resultados que las otras dos versiones del ABC; finalmente el SM-ABC de igual manera no pudo mejorar sus resultados. En los problemas g01, g02, g03, g04, g05, g06, g07, g08, g09, g10, g11, g12, g13, g14, g15, g16, g17, g18, g19, g21, g23 y g24 se alcanzó igualar al mejor resultado obtenido por alguna implementación del ABC.

### Discusión

Durante de la ejecución de nuestras pruebas es notorio que el algoritmo SPEC-ABC superó los resultados obtenidos por el ABC y el SM-ABC por lo que dichos resultados sugieren que nuestra implementación requiere de una menor cantidad de evaluaciones para alcanzar un resultado exitoso, así mismo es mayor la cantidad de corridas exitosas por prueba realizada y nuestra propuesta es más consistente al obtener soluciones dentro de la zona factible del espacio de búsqueda.

Cuadro 5.7: Comparativo de las primeras 4 medidas de desempeño en las primeras trece funciones de prueba. Los mejores resultados se remarcan con negritas. "Significa que la medida de desempeño no se puede calcular dada la ausencia de ejecuciones exitosas y/o factibles.

MEDIDAS DE RENDIMIENTO PARTE 1				
Funcion		ABC	SM-ABC	SPEC-ABC
g01	AFES	14222.53	14140.3	<b>12388.14</b>
	P	<b>1</b>	<b>1</b>	0.47
	SP	14222.53	<b>14140.3</b>	26546.02
	FP	<b>1</b>	<b>1</b>	<b>1</b>
g02	AFES	76981.61	<b>63513.17</b>	-
	P	<b>0.6</b>	0.4	0
	SP	<b>128302.69</b>	158782.92	-
	FP	<b>1</b>	<b>1</b>	<b>1</b>
g03	AFES	149707.3	152021.33	<b>109157.57</b>
	P	<b>1</b>	<b>1</b>	<b>1</b>
	SP	149707.3	152021.33	<b>109157.57</b>
	FP	<b>1</b>	<b>1</b>	<b>1</b>
g04	AFES	49405.77	48293.5	<b>9313.2</b>
	P	<b>1</b>	<b>1</b>	<b>1</b>
	SP	49405.77	48293.5	<b>9313.2</b>
	FP	<b>1</b>	<b>1</b>	<b>1</b>
g05	AFES	-	-	<b>211251</b>
	P	0	0	<b>0.73</b>
	SP	-	-	<b>288069.55</b>
	FP	0.1	0.07	<b>1</b>
g06	AFES	61765.43	63686.1	<b>7218.33</b>
	P	<b>1</b>	<b>1</b>	<b>1</b>
	SP	61765.43	63686.1	<b>7218.33</b>
	FP	<b>1</b>	<b>1</b>	<b>1</b>
g07	AFES	-	-	-
	P	0	0	0
	SP	-	-	-
	FP	<b>1</b>	<b>1</b>	<b>1</b>
g08	AFES	687.47	681.43	<b>607.93</b>
	P	<b>1</b>	<b>1</b>	<b>1</b>
	SP	687.47	681.43	<b>607.93</b>
	FP	<b>1</b>	<b>1</b>	<b>1</b>
g09	AFES	-	-	<b>115534.6</b>
	P	0	0	<b>0.17</b>
	SP	-	-	<b>693207.6</b>
	FP	<b>1</b>	<b>1</b>	<b>1</b>
g10	AFES	-	-	-
	P	0	0	0
	SP	-	-	-
	FP	<b>1</b>	<b>1</b>	<b>1</b>
g11	AFES	46396.47	43864.37	<b>57261.2</b>
	P	<b>1</b>	<b>1</b>	<b>1</b>
	SP	46396.47	43864.37	<b>57261.2</b>
	FP	<b>1</b>	<b>1</b>	<b>1</b>
g12	AFES	825.77	<b>701.3</b>	813.17
	P	<b>1</b>	<b>1</b>	<b>1</b>
	SP	825.77	<b>701.3</b>	813.17
	FP	<b>1</b>	<b>1</b>	<b>1</b>
g13	AFES	-	-	<b>173765.27</b>
	P	0	0	<b>0.37</b>
	SP	-	-	<b>473905.29</b>
	FP	0.13	0.13	<b>0.97</b>

Cuadro 5.8: Comparativo de las primeras cuatro medidas de rendimiento en las últimas once funciones de prueba. Los mejores resultados se remarcan con negritas. "Significa que la medida de desempeño no se puede calcular dada la ausencia de ejecuciones exitosas y/o factibles.

MEDIDAS DE RENDIMIENTO PARTE 2				
Funcion		ABC	SM-ABC	SPEC-ABC
g14	AFES	-	-	-
	P	-	-	0
	SP	-	-	-
	FP	-	-	1
g15	AFES	178565	<b>122056.67</b>	187204.32
	P	0.03	0.1	<b>0.73</b>
	SP	5356950	1220566.67	<b>255278.62</b>
	FP	0.17	0.47	1
g16	AFES	14394.8	14137.87	<b>5194.73</b>
	P	1	1	1
	SP	14394.8	14137.87	<b>5194.73</b>
	FP	1	1	1
g17	AFES	-	-	-
	P	0	0	0
	SP	-	-	-
	FP	0.07	0.03	1
g18	AFES	134903	56346.5	<b>42587.47</b>
	P	0.13	0.07	<b>0.5</b>
	SP	1011772.5	845197.5	<b>85174.93</b>
	FP	1	1	1
g19	AFES	-	-	-
	P	0	0	0
	SP	-	-	-
	FP	1	1	1
g20	AFES	-	-	-
	P	-	-	-
	SP	-	-	-
	FP	-	-	-
g21	AFES	-	-	<b>191340.2</b>
	P	-	-	<b>0.17</b>
	SP	-	-	<b>1148041.2</b>
	FP	-	-	1
g22	AFES	-	-	-
	P	-	-	-
	SP	-	-	-
	FP	-	-	-
g23	AFES	-	-	-
	P	0	0	0
	SP	-	-	-
	FP	-	-	1
g24	AFES	3454.97	3453.13	<b>1824.3</b>
	P	1	1	1
	SP	3454.97	3453.13	<b>1824.3</b>
	FP	1	1	1

### 5.4.2. EVALS

Los resultados estadísticos (mejor, media, peor y desviación estándar) para la medida de desempeño EVALS entre ABC, SM-ABC y SPEC-ABC se presentan en los Cuadros 5.9 y 5.10.

- **Mejor resultado:** Nuestro algoritmo propuesto alcanzó un mejor valor EVALS en 10 problemas en comparación con las otras dos versiones del ABC: g01, g06, g08, g10, g13, g14, g17, g18, g21, y g23. El ABC no pudo superar a alguna de las dos implementaciones y finalmente el SM-ABC obtuvo mejores resultado en los problemas g03, g05, g09, g11 y g15. En los problemas g01, g02, g04, g06, g08, g10, g12, g13, g14, g17, g18, g19, g21, g23 y g24 se alcanzó igualar al mejor resultado obtenido por los algoritmos. Ninguno de los 3 algoritmos (ABC, SM-ABC y SPEC-ABC) alcanzaron una solución factible para los problemas g20 y g22.
- **Resultado promedio:** Nuestro algoritmo propuesto alcanzó un mejor valor EVALS en 8 problemas en comparación con las otras dos versiones del ABC: g01, g10, g14, g16, g17, g18, g21 y g23. El ABC obtuvo mejores resultados en los problemas g15, g19 y g24 y el SM-ABC obtuvo mejores resultados en los problemas g03, g04, g08, g09, g11, g13 y g18. En los problemas g01, g02, g10, g14, g16, g17, g21 y g23 se alcanzó igualar al mejor resultado obtenido por alguna implementación del ABC.
- **Peor resultado:** Nuestro algoritmo propuesto alcanzó un mejor valor EVALS en 9 problemas en comparación con las otras dos versiones del ABC: g05, g06, g10, g14, g15, g16, g17, g21 y g23. El ABC obtuvo mejores resultados en los problemas g01, g07 y g12 y el SM-ABC obtuvo mejores resultados en los problemas g03, g04, g08, g09, g11, g13, g18 y g19. En los problemas g02, g05, g06, g10, g14, g15, g16, g17, g21, g23 y g24 se alcanzó igualar al mejor resultado obtenido por alguna implementación del ABC.
- **Desviación estándar:** Nuestro algoritmo propuesto alcanzó un mejor valor EVALS en 9 problemas en comparación con las otras dos versiones del ABC: g03, g06, g09, g10, g14, g15, g16, g21 y g23. El ABC obtuvo mejores resultados en los problemas g01, g07, g12 y g24 y el SM-ABC obtuvo mejores resultados en los problemas g04, g08, g11, g13, g17, g18 y g19. En los problemas g02, g03, g05, g06, g09, g10, g14, g15, g16, g21 y g23 se alcanzó igualar al mejor resultado obtenido por alguna implementación del ABC.

**Discusión** De acuerdo a los resultados obtenidos se puede ver que nuestra propuesta requiere, en lo general, de menos evaluaciones (requiere de un menor costo computacional) para generar una solución factible con respecto a las requeridas por las otras dos versiones del algoritmo ABC.



Cuadro 5.9: Comparación de resultados para la medida de rendimiento EVALS en las primeras trece funciones de prueba. Los mejores resultados se remarcan en negritas

EVALS PARTE 1				
Funcion		ABC	SM-ABC	SPEC-ABC
g01	Best	3.92E+02	3.18E+02	<b>2.41E+02</b>
	Mean	5.80E+02	5.72E+02	<b>5.17E+02</b>
	Worst	<b>7.19E+02</b>	8.27E+02	8.29E+02
	Std. Dev.	<b>8.36E+01</b>	1.17E+02	1.46E+02
g02	Best	<b>1E+00</b>	<b>1E+00</b>	<b>1E+00</b>
	Mean	<b>1E+00</b>	<b>1E+00</b>	<b>1E+00</b>
	Worst	<b>1E+00</b>	<b>1E+00</b>	<b>1E+00</b>
	Std. Dev.	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
g03	Best	2.49E+04	<b>3.25E+02</b>	8.25E+04
	Mean	4.20E+04	<b>4.22E+04</b>	1E+05
	Worst	5.31E+04	<b>5.76E+04</b>	1.11E+05
	Std. Dev.	6.68E+03	9.72E+03	<b>6.25E+03</b>
g04	Best	<b>1E+00</b>	<b>1E+00</b>	<b>1E+00</b>
	Mean	3.90E+00	<b>3.07E+00</b>	4.37E+00
	Worst	1.40E+01	<b>1.10E+01</b>	1.30E+01
	Std. Dev.	3.38E+00	<b>2.72E+00</b>	3.81E+00
g05	Best	1.62E+05	<b>1.52E+05</b>	1.73E+05
	Mean	1.82E+05	<b>1.79E+05</b>	1.83E+05
	Worst	2.15E+05	2.04E+05	<b>1.84E+05</b>
	Std. Dev.	2.01E+04	2.12E+04	<b>2.44E+03</b>
g06	Best	1.80E+02	1.52E+02	<b>1.31E+02</b>
	Mean	4.18E+02	<b>3.57E+02</b>	4E+02
	Worst	6.39E+02	5.86E+02	<b>5.49E+02</b>
	Std. Dev.	1.20E+02	1.17E+02	<b>9.23E+01</b>
g07	Best	<b>2.76E+02</b>	<b>2.76E+02</b>	4.28E+02
	Mean	5.67E+02	<b>5.62E+02</b>	8.60E+02
	Worst	<b>8.76E+02</b>	8.93E+02	1.38E+03
	Std. Dev.	<b>1.29E+02</b>	1.49E+02	2.67E+02
g08	Best	3E+00	9E+00	<b>2E+00</b>
	Mean	96.57E+00	<b>84E+00</b>	92.83E+00
	Worst	2.50E+02	<b>2.11E+02</b>	2.29E+02
	Std. Dev.	6.24E+01	<b>4.82E+01</b>	6.37E+01
g09	Best	3E+00	<b>2E+00</b>	47E+00
	Mean	9.28E+01	<b>7.79E+01</b>	9.91+01
	Worst	2.35E+02	<b>1.68E+02</b>	2.49E+02
	Std. Dev.	5.22E+01	4.64E+01	<b>4.32E+01</b>
g10	Best	5.82E+02	8.81E+02	<b>5.44E+02</b>
	Mean	1.66E+03	1.62E+03	<b>1.15E+03</b>
	Worst	3.22E+03	3.03E+03	<b>2.14E+03</b>
	Std. Dev.	6.36E+02	4.96E+02	<b>3.82E+02</b>
g11	Best	3.50E+03	<b>132</b>	8293
	Mean	2.65E+04	<b>22.34E+03</b>	4.06E+04
	Worst	5.83E+04	<b>5.21E+04</b>	1.06E+05
	Std. Dev.	1.34E+04	<b>1.27E+04</b>	2.30E+04
g12	Best	<b>1E+00</b>	<b>1E+00</b>	<b>1E+00</b>
	Mean	23.97E+00	<b>28.8E+00</b>	37.33E+00
	Worst	<b>70E+00</b>	89E+00	1.10E+02
	Std. Dev.	<b>2.28E+01</b>	2.61E+01	3.12E+01
g13	Best	1.65E+05	1.75E+05	<b>1.55E+05</b>
	Mean	1.79E+05	<b>1.78E+05</b>	1.80E+05
	Worst	1.85E+05	<b>1.82E+05</b>	1.84E+05
	Std. Dev.	9.07E+03	<b>2.77E+03</b>	6.90E+03

Cuadro 5.10: Comparación de resultados para la medida de rendimiento EVALS en las últimas once funciones de prueba. Los mejores resultados se remarcan en negritas. "Significa que no fueron encontradas soluciones factibles en ninguna ejecución independiente.

EVALS PARTE 2				
Función		ABC	SM-ABC	SPEC-ABC
g14	Best	-	-	<b>1.30E+05</b>
	Mean	-	-	<b>1.75E+05</b>
	Worst	-	-	<b>1.84E+05</b>
	Std. Dev.	-	-	<b>1.46E+04</b>
g15	Best	1.02E+05	<b>1.02E+05</b>	1.32E+05
	Mean	<b>1.52E+05</b>	1.68E+05	1.70E+05
	Worst	2.17E+05	2.30E+05	<b>1.83E+05</b>
	Std. Dev.	3.18E+04	4.56E+04	<b>1.36E+04</b>
g16	Best	<b>75E+00</b>	1.82E+02	1.09E+02
	Mean	5.97E+02	5.66E+02	<b>4.88E+02</b>
	Worst	1.38E+03	1.15E+03	<b>1.07E+03</b>
	Std. Dev.	2.97E+02	2.56E+02	<b>2.45E+02</b>
g17	Best	1.71E+05	1.73E+05	<b>1.37E+05</b>
	Mean	1.71E+05	1.81E+05	<b>1.77E+05</b>
	Worst	1.71E+05	1.88E+05	<b>1.84E+05</b>
	Std. Dev.	-	<b>7.16E+03</b>	1.28E+04
g18	Best	1.26E+03	1.33E+03	<b>8.91E+02</b>
	Mean	2.05E+03	<b>1.82E+03</b>	2.02E+03
	Worst	3.31E+03	<b>2.70E+03</b>	4.70E+03
	Std. Dev.	4.67E+02	<b>3.21E+02</b>	9.18E+02
g19	Best	<b>1E+00</b>	<b>1E+00</b>	<b>1E+00</b>
	Mean	<b>2.4E+00</b>	3.1E+00	2.87E+00
	Worst	11E+00	<b>8E+00</b>	9E+00
	Std. Dev.	2.40E+00	<b>2.02E+00</b>	2.05E+00
g20	Best	-	-	-
	Mean	-	-	-
	Worst	-	-	-
	Std. Dev.	-	-	-
g21	Best	-	-	<b>1.75E+05</b>
	Mean	-	-	<b>1.90E+05</b>
	Worst	-	-	<b>2.15E+05</b>
	Std. Dev.	-	-	<b>9.48E+03</b>
g22	Best	-	-	-
	Mean	-	-	-
	Worst	-	-	-
	Std. Dev.	-	-	-
g23	Best	-	-	<b>1.29E+05</b>
	Mean	-	-	<b>1.67E+05</b>
	Worst	-	-	<b>1.92E+05</b>
	Std. Dev.	-	-	<b>1.55E+04</b>
g24	Best	<b>1E+00</b>	<b>1E+00</b>	<b>1E+00</b>
	Mean	<b>2.07E+00</b>	2.6E+00	2.23E+00
	Worst	<b>7E+00</b>	10E+00	<b>7E+00</b>
	Std. Dev.	<b>1.28E+00</b>	2.09E+00	1.72E+00

### 5.4.3. PROGRESS RATIO

Los resultados estadísticos (mejor, media, peor y desviación estándar) para la medida de desempeño PROGRESS RATIO entre ABC, SM-ABC, SPEC-ABC se presentan en los Cuadros 5.11 y 5.12.

- **Mejor resultado:** Nuestro algoritmo propuesto alcanzó un mejor valor de PROGRESS RATIO en 10 problemas en comparación con las otras dos versiones del ABC: g04, g05, g11, g13, g14, g16, g17, g19, g21 y g23. El ABC obtuvo mejores resultados en los problemas g08, g10, g15 y g18 y finalmente el SM-ABC obtuvo mejores resultados en los problemas g01, g02, g03, g06, g07, g09, g12 y g24. En los problemas g04, g05, g11, g13, g14, g16, g17, g19, g21 y g23 se alcanzó igualar al mejor resultado obtenido por los algoritmos. Ninguno de los 3 algoritmos (ABC, SM-ABC y SPEC-ABC) alcanzaron una solución factible para los problemas g20 y g22.
- **Resultado promedio:** Nuestro algoritmo propuesto alcanzó un mejor valor de PROGRESS RATIO en 9 problemas en comparación con las otras dos versiones del ABC: g05, g07, g13, g14, g16, g17, g19, g21 y g23. El ABC obtuvo mejores resultados en los problemas g03, g08, g10, g11, g15, g18 y el SM-ABC obtuvo mejores resultados en los problemas g01, g02, g04, g06, g09, g12 y g24. En los problemas g05, g07, g13, g14, g16, g17, g19, g21 y g23 se alcanzó igualar al mejor resultado obtenido por alguna implementación del ABC.
- **Peor resultado:** Nuestro algoritmo propuesto alcanzó un mejor valor de PROGRESS RATIO en 9 problemas en comparación con las otras dos versiones del ABC: g05, g06, g08, g14, g16, g17, g19, g21 y g23. El ABC obtuvo mejores resultados en los problemas g03, g10, g18 y g24 y el SM-ABC obtuvo mejores resultados los problemas g01, g02, g04, g07, g09, g11, g12, g13 y g15. En los problemas g05, g06, g08, g14, g16, g17, g19, g21 y g23 se alcanzó igualar al mejor resultado obtenido por alguna implementación del ABC.
- **Desviación estándar:** Nuestro algoritmo propuesto alcanzó un mejor valor de PROGRESS RATIO en 9 problemas en comparación con las otras dos versiones del ABC: g03, g07, g12, g13, g14, g15, g19, g21 y g23. El ABC obtuvo mejores resultados en los problemas g01, g02, g05, g09, g10, g16, g18 y g24 el SM-ABC obtuvo mejores resultados en los problemas g04, g06, g08, g11 y g17. En los problemas g05, g07, g12, g13, g14, g15, g19, g21, y g23 se alcanzó igualar al mejor resultado obtenido por alguna implementación del ABC.

#### Discusión

En este estudio realizado es notorio que el SPEC-ABC mejora la capacidad de búsqueda dentro del área factible en comparación con sus otras dos versiones aunque en este caso, la diferencia es menos marcada que para las otras medidas de desempeño. La capacidad de mejora del algoritmo dentro del área factible de nuestra propuesta pudiera deberse a el valor estático para  $\phi$  que hace movimientos con el mismo tamaño de paso en cada dimensión del espacio de búsqueda.

Cuadro 5.11: Comparación de resultados para la medida de rendimiento PROGRESS RATIO en las primeras trece funciones de prueba. Los mejores resultados se remarcan en negritas.

PROGRESS RATIO 1				
Funcion		ABC	SM-ABC	SPEC-ABC
g01	Best	3.48E-01	<b>3.63E-01</b>	3.54E-01
	Mean	3.10E-01	<b>3.16E-01</b>	3.04E-01
	Worst	2.69E-01	<b>2.41E-01</b>	2.13E-01
	Std. Dev.	<b>2.47E-02</b>	3.19E-02	3.04E-02
g02	Best	3.24E-01	<b>3.30E-01</b>	3.27E-01
	Mean	3.12E-01	<b>3.17E-01</b>	2.93E-01
	Worst	2.90E-01	<b>3.00E-01</b>	2.55E-01
	Std. Dev.	<b>8.33E-03</b>	6.85E-03	1.58E-02
g03	Best	3.29E-01	<b>0.3416302</b>	9.81E-03
	Mean	<b>1.13E-01</b>	9.44E-02	2.05E-03
	Worst	<b>1.97E-02</b>	1.09E-02	3.4E-03
	Std. Dev.	8.22E-02	7.04E-02	<b>2.00E-03</b>
g04	Best	9.07E-02	963E-02	<b>9.63E-02</b>
	Mean	5.78E-02	<b>6.38E-02</b>	6.26E-02
	Worst	1.76E-02	<b>2.93E-02</b>	2.34E-02
	Std. Dev.	1.95E-02	<b>1.94E-02</b>	2.16E-02
g05	Best	1E-07	2E-07	<b>7.42E-05</b>
	Mean	5E-08	1E-07	<b>1.42E-05</b>
	Worst	1E-08	3E-08	<b>1E-07</b>
	Std. Dev.	<b>2.63E-08</b>	7.47E-08	1.93E-05
g06	Best	2.81E-01	<b>2.94E-01</b>	2.92E-01
	Mean	2.01E-01	<b>2.07E-01</b>	2.06E-01
	Worst	3.65E-02	4.95E-02	<b>6.18E-02</b>
	Std. Dev.	6.82E-02	<b>6.26E-02</b>	6.87E-02
g07	Best	2.43E-00	<b>2.4498995</b>	2.39E-00
	Mean	1.87E-00	1.92E-00	<b>1.95E-00</b>
	Worst	1.28E-00	<b>1.32E-00</b>	1.18E-00
	Std. Dev.	3.41E-01	3.44E-01	<b>2.99E-01</b>
g08	Best	<b>6.57E-01</b>	4.66E-01	5.27E-01
	Mean	<b>3.54E-01</b>	3.39E-01	3.37E-01
	Worst	0.1418201	1.21E-01	<b>1.75E-01</b>
	Std. Dev.	9.90E-02	<b>6.34E-02</b>	7.41E-02
g09	Best	4.18E-00	<b>4.78E-00</b>	4.699E-00
	Mean	2.11E-00	<b>3.14E-00</b>	1.603E-00
	Worst	0.35E-00	<b>3.92E-01</b>	2.65E-01
	Std. Dev.	<b>1.21E+00</b>	1.25E+00	1.34E+00
g10	Best	<b>6.34E-01</b>	6.29E-01	6.02E-01
	Mean	<b>5.17E-01</b>	5.05E-01	4.39E-01
	Worst	<b>4.19E-01</b>	3.41E-01	2.85E-01
	Std. Dev.	<b>6.15E-02</b>	6.63E-02	9.17E-02
g11	Best	1.20E-01	1.43E-01	<b>1.43E-01</b>
	Mean	<b>2.71E-02</b>	1.83E-02	1.48E-02
	Worst	6.42E-05	<b>1.12E-04</b>	6.78E-05
	Std. Dev.	3.80E-02	<b>3.55E-02</b>	3.57E-02
g12	Best	1.66E-01	<b>1.967E-01</b>	1.74E-01
	Mean	8.01E-02	<b>9.42E-02</b>	8.29E-02
	Worst	7.83E-03	<b>1.72E-02</b>	2.98E-04
	Std. Dev.	4.61E-02	4.54E-02	<b>4.23E-02</b>
g13	Best	5.85E-05	2.77E-04	<b>7.05E-03</b>
	Mean	2.82E-05	9.66E-05	<b>6.27E-04</b>
	Worst	8.2E-05	<b>1.85E-05</b>	6.4E-06
	Std. Dev.	2.22E-05	1.23E-04	<b>1.37E-03</b>

Cuadro 5.12: Comparación de resultados para la medida de rendimiento PROGRESS RATIO en las últimas once funciones de prueba. Los mejores resultados se remarcan en negritas. ”Significa que no fueron encontradas soluciones factibles en ninguna ejecución independiente.

PROGRESS RATIO PARTE 2				
Funcion		ABC	SM-ABC	SPEC-ABC
g14	Best	-	-	<b>1.27E-02</b>
	Mean	-	-	<b>1.84E-03</b>
	Worst	-	-	<b>1.94E-05</b>
	Std. Dev.	-	-	<b>3.48E-03</b>
g15	Best	<b>5.43E-04</b>	3.59E-05	6.9E-06
	Mean	<b>4.01E-05</b>	4.3E-06	1.8E-06
	Worst	1.E-08	<b>3.E-08</b>	2.7E-08
	Std. Dev.	1.40E-04	1.19E-05	<b>1.74E-06</b>
g16	Best	2.03E-01	2.16E-01	<b>2.47E-01</b>
	Mean	1.58E-01	1.60E-01	<b>1.64E-01</b>
	Worst	1.10E-02	7.21E-02	<b>9.63E-02</b>
	Std. Dev.	<b>2.72E-02</b>	3.27E-02	3.78E-02
g17	Best	2E-09	1E-07	<b>2.96E-05</b>
	Mean	2E-09	4E-08	<b>3.9E-06</b>
	Worst	2E-09	1E-08	<b>2E-09</b>
	Std. Dev.	-	<b>5.94E-08</b>	8.60E-06
g18	Best	<b>3.29E-01</b>	3.19E-01	3.22E-01
	Mean	<b>2.73E-01</b>	2.59E-01	2.56E-01
	Worst	<b>1.83E-01</b>	1.32E-01	8.24E-02
	Std. Dev.	<b>4.13E-02</b>	4.22E-02	5.88E-02
g19	Best	3.56E-00	3.46E-00	<b>3.57E-00</b>
	Mean	3.09E-00	3.12E-00	<b>3.20E-00</b>
	Worst	2.06E-00	2.53E-00	<b>2.57E-00</b>
	Std. Dev.	3.97E-01	2.79E-01	<b>2.22E-01</b>
g20	Best	-	-	-
	Mean	-	-	-
	Worst	-	-	-
	Std. Dev.	-	-	-
g21	Best	-	-	<b>5.53E-00</b>
	Mean	-	-	<b>6.51E-02</b>
	Worst	-	-	<b>3.02E-05</b>
	Std. Dev.	-	-	<b>1.61E-01</b>
g22	Best	-	-	-
	Mean	-	-	-
	Worst	-	-	-
	Std. Dev.	-	-	-
g23	Best	-	-	<b>4.42E-01</b>
	Mean	-	-	<b>6.80E-02</b>
	Worst	-	-	<b>1.63E-04</b>
	Std. Dev.	-	-	<b>8.99E-02</b>
g24	Best	2.91E-01	<b>3.34E-01</b>	3.07E-01
	Mean	1.95E-01	<b>2.14E-01</b>	1.94E-01
	Worst	<b>2.7E-02</b>	5.73E-02	4.63E-02
	Std. Dev.	<b>5.90E-02</b>	6.13E-02	7.22E-02

## 5.5. Comparativo con algoritmos del estado del arte

Los resultados estadísticos (mejor, media, peor y desviación estándar) de SPEC-ABC y de tres algoritmos del estado del arte en optimización con restricciones usando algoritmos evolutivos y de inteligencia colectiva se presentan en el Cuadro 5.13. Sólo se muestran los primeros trece problemas de prueba pues fueron los resueltos por los tres algoritmos usados en el comparativo.

### 5.5.1. Comparativo entre SAPF-GA y nuestro algoritmo

- **Mejor resultado:** Nuestro algoritmo propuesto mostró mejores resultados que los provistos por SAPF-GA en los problemas g03, g04, g05, g06, g07, g09 y g10. En 5 problemas (g01, g08, g11, g12 y g13) ambos algoritmos alcanzaron los mismos resultados al mejor conocido.
- **Resultado promedio:** Nuestro algoritmo propuesto mostró mejores resultados en 11 problemas (g03-g13).
- **Peor resultado:** Nuestro algoritmo propuesto mostró mejores resultados en 10 problemas (g03-g12).
- **Desviación estándar:** Nuestro algoritmo propuesto mostró mejores resultados en 12 problemas (g02-g13).

### 5.5.2. Comparativo entre HCOEA y nuestro algoritmo

- **Mejor resultado:** Nuestro algoritmo propuesto mostró un mejor resultado que el provisto por HCOEA en el problema g05. El algoritmo HCOEA alcanzó mejores resultados en los problemas g02, g07 y g10. Ambos algoritmos mostraron iguales resultados a los conocidos actualmente en los problemas g01, g03, g04, g06, g08, g09, g11, g12 y g13.
- **Resultado promedio:** El algoritmo HCOEA alcanzó mejores resultados que nuestra propuesta en los problemas g01, g02, g05, g07, g09, g10 y g13. Ambos algoritmos alcanzaron el mejor resultado a los actualmente publicados en los problemas g03, g04, g06, g07, g08, g11 y g12.
- **Peor resultado:** El algoritmo HCOEA alcanzó mejores resultados que nuestra propuesta en los problemas g01, g02, g05, g07, g09, g10 y g13. Ambos algoritmos alcanzaron igual resultado a los actualmente publicados en los problemas g03, g04, g06, g08, g11 y g12.
- **Desviación estándar:** Nuestro algoritmo alcanzó mejores resultados que nuestra propuesta en los problemas g04, g06, g08 y g11. Ambos algoritmos alcanzaron el mismo resultado en los problemas g08 y g12.

### 5.5.3. Comparativo entre ATMES y nuestro algoritmo

- **Mejor resultado:** Nuestro algoritmo propuesto mostró un mejor resultado que el provisto por ATMES en el problema g05. Ambos algoritmos mostraron iguales resultados a los conocidos actualmente en los problemas g01, g03, g04, g06, g08, g11, g12 y g13.
- **Resultado promedio:** Nuestro algoritmo propuesto mostró mejores resultados que los provistos por ATMES en los problemas g05 y g10; ATMES alcanzó mejores resultados que los provistos por nuestro algoritmo en los problemas g01, g02, g07, g09 y g13. Ambos algoritmos alcanzaron el resultado óptimo en los problemas g03, g04, g06, g08, g11 y g12.
- **Peor resultado:** Nuestro algoritmo mostró mejores resultados que el ATMES en los problemas g05, g10 y g12. Ambos alcanzaron el mejor resultado conocido hasta el momento en los problemas g03, g04, g06, g08 y g11.

- **Desviación estándar:** Nuestro algoritmo alcanzó mejores resultados que el ATMES en los problemas g03, g04, g05, g06, g08, g10, g11 y g12.

### Discusión

Con respecto a la calidad de los resultados obtenidos (la mejor solución obtenida) se puede observar lo siguiente: en comparación con los resultados publicados por SAPF-GA nuestro algoritmo alcanzó mejor o igual resultado en 12 problemas, para el caso de el resultado peor resultado nuestro algoritmo alcanzó mejores resultados en 10 problemas. Nuestra propuesta ha presentado mejores resultados que el SAPF-GA en la mayoría de los problemas.

En comparación con los resultados publicados por HCOEA nuestro algoritmo sólo logró la obtención de un mejor resultado en el problema g05. Nuestra propuesta alcanzó iguales resultados al HCOEA en 9 problemas y para el peor resultado lo equiparó en 6 problemas de los 13 planteados. En general el HCOEA presenta mejor calidad en resultados aunque nuestro algoritmo lo equipara en el 69% de los problemas.

En comparación con los resultados publicados por ATMES nuestro algoritmo sólo logró la obtención de un mejor resultado en el problema g05. Nuestra propuesta equiparó los resultados del ATMES en 8 de los 13 problemas; para el peor resultado nuestra propuesta lo superó en 3 problemas y lo equiparó en 5 problemas. En general el ATMES presenta mejor calidad en sus resultados aunque nuestra propuesta lo equipara en el el 69% de los problemas.

Con respecto a la consistencia de los resultados obtenidos (mejores valores en la media, peor y desviación estándar) se puede observar lo siguiente: en comparación con los resultados del SAPF-GA nuestro algoritmo lo superó o mejoró en 12 problemas para el mejor resultado, 11 problemas para el resultado promedio, 10 para el peor resultado y 12 para los resultados obtenidos de la desviación estándar por lo que nuestro algoritmo es más consistente en la obtención de resultados que el SAPF-GA.

En comparación con los resultados mostrados por el HCOEA nuestro algoritmo los superó en 1 problema e igualó 9 de los 13 problemas. Nuestra propuesta no pudo superar el resultado promedio y peor resultado publicado por el HCOEA aunque para el resultado promedio lo igualó en 7 problemas, para el peor resultado lo igualó en 6 problemas. Al obtener la desviación estándar nuestro algoritmo superó al HCOEA en 4 problemas y lo igualó en 2 más, a pesar de que nuestra propuesta mostró resultados competitivos el HCOEA comprobó ser más consistente en sus resultados.

Para la comparación contra el ATMES nuestro algoritmo sólo lo superó en 1 problema y lo igualó en otros 8; en el caso del resultado promedio nuestro algoritmo mostró mejores resultados en 2 problemas y lo igualó en 6 más; para el peor resultado nuestro algoritmo mostró mejores resultados en 3 problemas y lo igualó en 5 más y finalmente en la desviación estándar nuestro algoritmo mostró mejores resultados en 8 problemas; con todo esto es posible notar que nuestro algoritmo se mantiene competitivo con respecto al ATMES pero éste termina siendo más consistente en la obtención de resultados.

Cuadro 5.13: Comparación de nuestro algoritmo (titulado SPEC-ABC) con respecto a los algoritmos del estado del arte (SAPF-GA [26], HCOEA [27] and ATMES [28]) en las 13 funciones de benchmark (g01-g13). El resultado marcado en negritas indica el mejor resultado u óptimo global (mejor solución conocida) alcanzado.

Function/ Optimal		Methods			
		SAPF-GA (2006)	HCOEA (2007)	ATMES (2008)	SPEC-ABC
g01 -15	Best	<b>-15</b>	<b>-15</b>	<b>-15</b>	<b>-15</b>
	Mean	-14.552	<b>-15</b>	<b>-15</b>	-14.13
	Worst	-13.097	-14.999	<b>-15</b>	-12.45
	Std. Dev.	7.00E-01	4.30E-07	<b>1.60E-14</b>	9.16E-01
g02 <b>-0.803619</b>	Best	-0.803202	-0.803241	<b>-0.803388</b>	-0.709034
	Mean	-0.755798	<b>-0.801258</b>	-0.790148	-0.471210
	Worst	-0.745712	<b>-0.792363</b>	-0.756986	-0.319046
	Std. Dev.	1.33E-01	<b>3.83E-03</b>	1.30E-02	1.05E-01
g03 -1	Best	<b>-1</b>	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Mean	-0.964	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Worst	-0.887	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Std. Dev.	3.01E-01	<b>1.30E-12</b>	5.90E-05	2.73E-05
g04 <b>-30665.539</b>	Best	-30665.401	<b>-30665.539</b>	<b>-30665.539</b>	<b>-30665.539</b>
	Mean	-306659.221	<b>-30665.539</b>	<b>-30665.539</b>	<b>-30665.539</b>
	Worst	-30656.471	<b>-30665.539</b>	<b>-30665.539</b>	<b>-30665.539</b>
	Std. Dev.	2.04E+00	5.40E-07	7.40E-12	<b>0.00E00</b>
g05 <b>5126.498</b>	Best	5126.907	5126.498	5126.498	<b>5126.496</b>
	Mean	5214.232	<b>5126.498</b>	5127.648	5126.526
	Worst	5564.642	<b>5126.498</b>	5135.256	5126.859
	Std. Dev.	2.47E+02	<b>1.73E-07</b>	1.80E+00	7.98E-02
g06 <b>-6961.814</b>	Best	-6961.046	<b>-6961.813</b>	<b>-6961.814</b>	<b>-6961.814</b>
	Mean	-6953.061	<b>-6961.813</b>	<b>-6961.814</b>	<b>-6961.814</b>
	Worst	-6943.304	<b>-6961.813</b>	<b>-6961.814</b>	<b>-6961.814</b>
	Std. Dev.	5.88E+00	8.51E-12	4.60E-12	<b>0.00E+00</b>
g07 <b>24.306</b>	Best	24.838	24.306	<b>24.306</b>	24.316
	Mean	27.328	<b>24.307</b>	24.316	24.657
	Worst	33.095	<b>24.309</b>	24.359	25.544
	Std. Dev.	2.17E+00	<b>7.12E-04</b>	1.10E-02	3.29E-01
g08 <b>-0.095825</b>	Best	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>
	Mean	-0.095635	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>
	Worst	-0.092697	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>
	Std. Dev.	1.06E-03	2.42E-17	2.80E-17	<b>0.00E00</b>
g09 <b>680.63</b>	Best	680.773	<b>680.630</b>	<b>680.63</b>	<b>680.630</b>
	Mean	681.246	<b>680.630</b>	680.639	680.643
	Worst	682.081	<b>680.630</b>	680.673	680.857
	Std. Dev.	3.22E-01	<b>9.41E-08</b>	1.00E-02	4.09E-02
g10 <b>7049.248</b>	Best	7069.981	<b>7049.286</b>	7052.253	7049.547
	Mean	7238.964	<b>7049.525</b>	7250.437	7116.934
	Worst	7489.406	<b>7049.984</b>	7560.224	7362.639
	Std. Dev.	1.38E+02	<b>1.50E-01</b>	1.20E+02	8.21E+01
g11 <b>0.75</b>	Best	<b>0.749</b>	<b>0.75</b>	<b>0.75</b>	<b>0.75</b>
	Mean	0.751	<b>0.75</b>	<b>0.75</b>	<b>0.75</b>
	Worst	0.757	<b>0.75</b>	<b>0.75</b>	<b>0.75</b>
	Std. Dev.	2.00E-03	1.55E-12	3.40E-04	<b>0.00E00</b>
g12 -1	Best	<b>-1</b>	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Mean	-0.99994	<b>-1</b>	<b>-1</b>	<b>-1</b>
	Worst	-0.999548	<b>-1</b>	-0.994	<b>-1</b>
	Std. Dev.	1.41E-04	<b>0.00E+00</b>	1.00E-03	<b>0.00E+00</b>
g13 <b>0.05395</b>	Best	<b>0.053941</b>	<b>0.0539498</b>	<b>0.05395</b>	<b>0.053942</b>
	Mean	0.28627	<b>0.0539498</b>	<b>0.053959</b>	0.263967
	Worst	0.885276	<b>0.0539499</b>	<b>0.053999</b>	1
	Std. Dev.	2.75E-01	<b>8.68E-08</b>	<b>1.30E-05</b>	2.36E-01



## Capítulo 6

# Conclusiones y trabajo futuro

En este trabajo se presentó la inclusión de la técnica  $\epsilon$  – *constraint* dentro de una versión del algoritmo de la colonia artificial de abejas conocido como SM-ABC. Esta nueva versión, llamada SPEC-ABC modificó también el operador usado por las abejas empleadas y en espera al usar valores fijos de  $\phi$  para todas las variables del problema (dimensiones del espacio de búsqueda). Cuatro experimentos fueron diseñados para comparar a SPEC-ABC con dos versiones previas basadas en ABC: El ABC original y SM-ABC. Se utilizaron seis medidas de desempeño encontradas en la literatura especializada y finalmente se comparó el desempeño de SPEC-ABC con tres algoritmos del estado del arte en optimización con restricciones usando algoritmos evolutivos y de inteligencia colectiva. Los resultados obtenidos mostraron un mejor desempeño de SPEC-ABC con respecto a sus predecesores basados en ABC, tanto en un comparativo directo como en uno indirecto. Más aún, los resultados de las medidas de desempeño sugieren que SPEC-ABC es capaz de generar un mayor número de corridas exitosas con un menor número de evaluaciones y también puede alcanzar la zona factible del espacio de búsqueda con menos evaluaciones con respecto a las requeridas por sus dos algoritmos predecesores. Más aún, SPEC-ABC fue capaz de mejorar una ligera mayor frecuencia soluciones dentro de la zona factible. Finalmente, SPEC-ABC obtuvo resultados comparables e incluso mejores que SAPF-GA, uno de los algoritmos del estado del arte. Sin embargo, SPEC-ABC sólo pudo obtener resultados comparables e incluso fue superado, sobre todo en consistencia, por los otros dos algoritmos del estado del arte: HCOEA y ATMES.

El trabajo futuro se centra en precisamente mejorar los operadores de los tres tipos de abejas para lograr un mejor desempeño final así como el utilizar algoritmos de búsqueda local para mejorar el desempeño del algoritmo. Finalmente, se planea usar SPEC-ABC para resolver problemas del mundo real, principalmente en el dominio mecatrónico.

# Apéndice A

## Funciones de benchmark

En este apéndice son descritas las 24 funciones de Benchmark evaluadas en este trabajo de tesis.  
**g01**

Minimizar:

$$f(\vec{x}) = \sum_{i=1}^4 x_i - 5 \sum_{i=1}^4 x_i^2 - \sum_{i=5}^1 3x_i \quad (\text{A.1})$$

Sujeto a:

$$\begin{aligned} g_1(\vec{x}) &= 2x_1 + 2x_2 + x_10 + x_11 - 10 \leq 0 \\ g_2(\vec{x}) &= 2x_1 + 2x_3 + x_10 + x_12 - 10 \leq 0 \\ g_3(\vec{x}) &= 2x_2 + 2x_3 + x_10 + x_12 - 10 \leq 0 \\ g_4(\vec{x}) &= -8x_1 + x_10 \leq 0 \\ g_5(\vec{x}) &= -8x_2 + x_11 \leq 0 \\ g_6(\vec{x}) &= -8x_3 + x_12 \leq 0 \\ g_7(\vec{x}) &= -2x_4 - x_5 + x_10 \leq 0 \\ g_8(\vec{x}) &= -2x_6 - x_7 + x_11 \leq 0 \\ g_9(\vec{x}) &= -2x_8 - x_9 + x_12 \leq 0 \end{aligned}$$

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

**g02**

Minimizar:

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

Sujeto a:

$$\begin{aligned} g_1(\vec{x}) &= 0,75 - \prod_{i=1}^n x_i \leq 0 \\ g_2(\vec{x}) &= \sum_{i=1}^n x_i - 0,75n \leq 0 \end{aligned}$$

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

**g03**

Minimizar:

$$f(\vec{x}) = -(\sqrt{n})^n \prod_{i=0}^n x_i \quad (\text{A.3})$$

Sujeto a:

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

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

**g04**

Minimizar:

$$f(\vec{x}) = 5,3578547x_3^2 + 0,8356891x_1x_5 + 37,293239x_1 - 40792,141 \quad (\text{A.4})$$

Sujeto a:

$$\begin{aligned} g_1(\vec{x}) &= 85,334407 + 0,0056858x_2x_5 + 0,0006262x_1x_4 - 0,0022053x_3x_5 - 92 \leq 0 \\ g_2(\vec{x}) &= -85,334407 - 0,0056858x_2x_5 - 0,0006262x_1x_4 + 0,0022053x_3x_5 \leq 0 \\ g_3(\vec{x}) &= 80,51249 + 0,0071317x_2x_5 + 0,0029955x_1x_2 + 0,0021813x_3^2 - 110 \leq 0 \\ g_4(\vec{x}) &= -80,51249 - 0,0071317x_2x_5 - 0,0029955x_1x_2 - 0,0021813x_3^2 + 90 \leq 0 \\ g_5(\vec{x}) &= 9,300961 + 0,0047026x_3x_5 + 0,0012547x_1x_3 + 0,0019085x_3x_4 - 25 \leq 0 \\ g_6(\vec{x}) &= -9,300961 - 0,0047026x_3x_5 - 0,0012547x_1x_3 - 3 - 0,0019085x_3x_4 + 20 \leq 0 \end{aligned}$$

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

**g05**

Minimizar:

$$f(\vec{x}) = 3x_1 + 0,000001x_1^3 + 2x_2 + (0,000002/3)x_2^3 \quad (\text{A.5})$$

Sujeto a:

$$\begin{aligned} g_1(\vec{x}) &= -x_4 + x_3 - 0,55 \leq 0 \\ g_2(\vec{x}) &= -x_3 + x_4 - 0,55 \leq 0 \end{aligned}$$

$$\begin{aligned} h_3(\vec{x}) &= 1000\text{sen}(-x_3 - 0,25) + 1000\text{sen}(-x_4 - 0,25) + 894,8 - x_1 = 0 \\ h_4(\vec{x}) &= 1000\text{sen}(x_3 - 0,25) + 1000\text{sen}(x_3 - x_4 - 0,25) + 894,8 - x_2 = 0 \\ h_5(\vec{x}) &= 1000\text{sen}(x_4 - 0,25) + 1000\text{sen}(x_4 - x_3 - 0,25) + 1294,8 = 0 \end{aligned}$$

Donde los límites son  $0 \leq x_1 \leq 1200$ ,  $0 \leq x_2 \leq 1200$ ,  $-0,55 \leq x_3 \leq 0,55$  y  $-0,55 \leq x_4 \leq 0,55$ .

**g06**

Minimizar:

$$f(\vec{x}) = (x_1 - 10)^3 + (x_2 - 20)^3 \quad (\text{A.6})$$

Sujeto a:

$$\begin{aligned} g_1(\vec{x}) &= -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0 \\ g_2(\vec{x}) &= (x_1 - 6)^2 + (x_2 - 5)^2 - 82,81 \leq 0 \end{aligned}$$

Donde los límites son  $13 \leq x_1 \leq 100$ ,  $0 \leq x_2 \leq 100$ .

**g07**

Minimizar:

$$f(\vec{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10}?)^2 + 45 \quad (\text{A.7})$$

Sujeto a:

$$\begin{aligned} g_1(\vec{x}) &= -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0 \\ g_2(\vec{x}) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0 \\ g_3(\vec{x}) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0 \\ g_4(\vec{x}) &= 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0 \\ g_5(\vec{x}) &= 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0 \\ g_6(\vec{x}) &= x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0 \\ g_7(\vec{x}) &= 0,5(x_1 - 8)^2 + 2(x_2 - 4)2 + 3x_5^2 - x_6 - 30 \leq 0 \\ g_8(\vec{x}) &= -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \end{aligned}$$

Donde los límites son  $-10 \leq x_i \leq 10 (i = 1, \dots, 10)$ .

**g08**

Minimizar:

$$f(\vec{x}) = -\frac{\text{sen}^3(2\pi x_1)\text{sen}(2\pi x_2)}{x_1^3(x_1 + x_2)} \quad (\text{A.8})$$

Sujeto a:

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

Donde los límites son  $0 \leq x_1 \leq 10$  y  $0 \leq x_2 \leq 10$ .

**g09**

Minimizar:

$$f(\vec{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \quad (\text{A.9})$$

Sujeto a:

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

Donde los límites son  $-10 \leq x_i \leq 10 (i = 1, \dots, 7)$ .

**g10**

Minimizar:

$$f(\vec{x}) = x_1 + x_2 + x_3 \quad (\text{A.10})$$

Sujeto a:

$$\begin{aligned}
g_1(\vec{x}) &= -1 + 0,0025(x_4 + x_6) \leq 0 \\
g_2(\vec{x}) &= -1 + 0,0025(x_5 + x_7 - x_4) \leq 0 \\
g_3(\vec{x}) &= -1 + 0,01(x_8 - x_5) \leq 0 \\
g_4(\vec{x}) &= -x_1x_6 + 833,33252x_4 + 100x_1 - 83333,333 \leq 0 \\
g_5(\vec{x}) &= -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0 \\
g_6(\vec{x}) &= -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0
\end{aligned}$$

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

**g11**

Minimizar:

$$f(\vec{x}) = x_1^2 + (x_2 - 1)^2 \quad (\text{A.11})$$

Sujeto a:

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

Donde los límites son  $-1 \leq x_1 \leq 1$  y  $-1 \leq x_2 \leq 1$ .

**g12**

Minimizar:

$$f(\vec{x}) = -(100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2)/100 \quad (\text{A.12})$$

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  $p, q, r = 1, 2, \dots, 9$ . La región factible del área de búsqueda consiste en  $9^3$  esferas disjuntas. Un punto  $(x_1, x_2, x_3)$  es factible si y sólo si existe  $p, q$  y  $r$  tal cual muestra la desigualdad de la parte superior.

**g13**

Minimizar:

$$f(\vec{x}) = e^{x_1x_2x_3x_4x_5} \quad (\text{A.13})$$

Sujeto a:

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

$$h_2(\vec{x}) = x_2x_3 - 5x_4x_5 = 0$$

$$h_3(\vec{x}) = x_1^3 + x_2^3 + 1 = 0$$

Donde los límites son  $-2,3 \leq x_i \leq 2,3 (i = 1, 2)$  y  $-3,2 \leq x_i \leq 3,2 (i = 3, 4, 5)$ .

**g14**

Minimizar:

$$f(\vec{x}) = \sum_{i=1}^1 0x_i \left( c_i + \ln \frac{x_i}{\sum_{j=1}^{10} x_j} \right) \quad (\text{A.14})$$

Sujeto a:

$$h_1(\vec{x}) = x_1 + 2x_2 + 2x_3 + x_6 + x_{10} - 2 = 0$$

$$h_2(\vec{x}) = x_4 + 2x_5 + x_6 + x_7 - 1 = 0$$

$$h_3(\vec{x}) = x_3 + x_7 + x_8 + 2x_9 + x_{10} - 1 = 0$$

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

$$f(\vec{x}) = 1000 - x_1^2 - 2x_2 - x_3^2 - x_1x_2 - x_1x_3 \quad (\text{A.15})$$

Sujeto a:

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

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

**g16**

Minimizar:

$$\begin{aligned} f(\vec{x}) &= 0,000117y_{14} + 0,1365 + 0,00002358y_{13} + 0,000001502y_{16} + 0,0321y_{12} \\ &+ 0,004324y_5 + 0,0001 \frac{c_{15}}{c_{16}} + 37,48 \frac{y_2}{c_{12}} - 0,0000005843y_{17} \end{aligned} \quad (\text{A.16})$$

Sujeto a:

$$\begin{aligned}g_1(\vec{x}) &= \frac{0,28}{0,75}y_5 - y_4 \leq 0 \\g_2(\vec{x}) &= x_3 - 1,5x_2 \leq 0 \\g_3(\vec{x}) &= 3496\frac{y_2}{c_{12}} - 21 \leq 0 \\g_4(\vec{x}) &= 110,6 + y_1 - \frac{62212}{c_{17}} \leq 0 \\g_5(\vec{x}) &= 213,1 - y_1 \leq 0 \\g_6(\vec{x}) &= y_1 - 405,23 \leq 0 \\g_7(\vec{x}) &= 17,505 - y_2 \leq 0 \\g_8(\vec{x}) &= y_2 1053,6667 \leq 0 \\g_9(\vec{x}) &= 11,275 - y_3 \leq 0 \\g_{10}(\vec{x}) &= y_3 - 35,03 \leq 0 \\g_{11}(\vec{x}) &= 214,228 - y_4 \leq 0 \\g_{12}(\vec{x}) &= y_4 - 665,585 \leq 0 \\g_{13}(\vec{x}) &= 7,458 - y_5 \leq 0 \\g_{14}(\vec{x}) &= y_5 - 584,463 \leq 0 \\g_{15}(\vec{x}) &= 0,961 - y_6 \leq 0 \\g_{16}(\vec{x}) &= y_6 - 265,916 \leq 0 \\g_{17}(\vec{x}) &= 1,612 - y_7 \leq 0 \\g_{18}(\vec{x}) &= y_7 - 7,046 \leq 0 \\g_{19}(\vec{x}) &= 0,146 - y_8 \leq 0 \\g_{20}(\vec{x}) &= y_8 - 0,222 \leq 0 \\g_{21}(\vec{x}) &= 107,99 - y_9 \leq 0 \\g_{22}(\vec{x}) &= y_9 - 273,366 \leq 0 \\g_{23}(\vec{x}) &= 922,693 - y_{10} \leq 0 \\g_{24}(\vec{x}) &= y_{10} - 1286,105 \leq 0 \\g_{25}(\vec{x}) &= 926,832 - y_{11} \leq 0 \\g_{26}(\vec{x}) &= y_{11} - 1444,046 \leq 0 \\g_{27}(\vec{x}) &= 18,766 - y_{12} \leq 0 \\g_{28}(\vec{x}) &= y_{12} - 537,141 \leq 0 \\g_{29}(\vec{x}) &= 1072,163 - y_{13} \leq 0 \\g_{30}(\vec{x}) &= y_{13} 3247,039 \leq 0 \\g_{31}(\vec{x}) &= 8961,448 - y_{14} \leq 0 \\g_{32}(\vec{x}) &= y_{14} - 26844,086 \leq 0 \\g_{33}(\vec{x}) &= 0,063 - y_{15} \leq 0 \\g_{34}(\vec{x}) &= y_{15} - 0,386 \leq 0 \\g_{35}(\vec{x}) &= 71084,33 - y_{16} \leq 0 \\g_{36}(\vec{x}) &= -140000 + y_{16} \leq 0 \\g_{37}(\vec{x}) &= 2802713 - y_{17} \leq 0 \\g_{38}(\vec{x}) &= y_{17} - 12146108 \leq 0\end{aligned}$$

Donde:

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

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:

$$f(\vec{x}) = f(x_1) + f(x_2) \quad (\text{A.17})$$

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:

$$h_1(\vec{x}) = -x_1 + 300 - \frac{x_3x_4}{131,078} \cos(1,48477 - x_6) + \frac{0,90798x_3^2}{131,078} \cos(1,47588) = 0$$

$$h_2(\vec{x}) = -x_2 - \frac{x_3x_4}{131,078} \cos(1,48477 + x_6) + \frac{0,90798x_4^2}{131,078} \cos(1,47588) = 0$$

$$h_3(\vec{x}) = -x_5 - \frac{x_3x_4}{131,078} \text{sen}(1,48477 + x_6) + \frac{0,90798x_4^2}{131,078} \text{sen}(1,47588) = 0$$

$$h_4(\vec{x}) = 200 - \frac{x_3x_4}{131,078} \text{sen}(1,48477 - x_6) + \frac{0,90798x_4^2}{131,078} \text{sen}(1,47588) = 0$$

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:

$$f(\vec{x}) = -0,5(x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7) \quad (\text{A.18})$$

Sujeto a:

$$g_1(\vec{x}) = x_3^2 + x_4^2 - 1 \leq 0$$

$$g_2(\vec{x}) = x_9^2 - 1 \leq 0$$

$$g_3(\vec{x}) = x_5^2 + x_6^2 - 1 \leq 0$$

$$g_4(\vec{x}) = x_1^2 + (x_2 - x_9)^2 - 1 \leq 0$$

$$g_5(\vec{x}) = (x_1 - x_5)^2 + (x_2 - x_6)^2 - 1 \leq 0$$

$$g_6(\vec{x}) = (x_1 - x_7)^2 + (x_2 - x_8)^2 - 1 \leq 0$$

$$g_7(\vec{x}) = (x_3 - x_5)^2 + (x_4 - x_6)^2 - 1 \leq 0$$

$$g_8(\vec{x}) = (x_3 - x_7)^2 + (x_4 - x_8)^2 - 1 \leq 0$$

$$g_9(\vec{x}) = x_7^2 + (x_8 - x_9)^2 - 1 \leq 0$$

$$g_{10}(\vec{x}) = x_2x_3 - x_1x_4 \leq 0$$

$$g_{11}(\vec{x}) = -x_3x_9 \leq 0$$

$$g_{12}(\vec{x}) = x_5x_9 \leq 0$$

$$g_{13}(\vec{x}) = x_6x_7 - x_5x_8$$

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

**g19**

Minimizar:

$$f(\vec{x}) = \sum_{j=15} i = 15c_{ij}x_{(10+i)x_{(10+j)}} + 2 \sum_{j=15} d_j x_{10+j}^3 - \sum_{i=110} b_i x_i \quad (\text{A.19})$$

Sujeto a:

$$g_j(\vec{x}) = -2 \sum_{i=15} c_{ij} x_{(10+i)} - e_j + \sum_{i=110} a_{ij} x_i \leq 0, \quad j = 1, \dots, 5$$

Donde  $\vec{b} = [-40, -2, -0, 25, -4, -4, -1, -40, -60, 5, 1]$  y los datos restantes se encuentran en la tabla A.1. Los límites son  $0 \leq x_i \leq 10 (i = 1, \dots, 15)$ .

Cuadro A.1: Conjunto de datos del problema g19.

$j$	1	2	3	4	5
$e_j$	-15	-27	-36	-18	-12
$c_{1j}$	30	-20	-10	32	-10
$c_{2j}$	-20	39	-6	-31	32
$c_{3j}$	-10	-6	10	-6	-10
$c_{4j}$	32	-31	-6	39	-20
$c_{5j}$	-10	32	-10	-20	30
$d_j$	4	8	10	6	2
$a_{1j}$	-16	2	0	1	0
$a_{2j}$	0	-2	0	0.4	2
$a_{3j}$	-3.5	0	2	0	0
$a_{4j}$	0	-2	0	-4	-1
$a_{5j}$	0	-9	-2	1	-2.8
$a_{6j}$	2	0	-4	0	0
$a_{7j}$	-1	-1	-1	-1	-1
$a_{8j}$	-1	-2	-3	-2	-1
$a_{9j}$	1	2	3	4	5
$a_{10j}$	1	1	1	1	1

**g20**

Minimizar:

$$f(\vec{x}) = \sum_{i=124} a_i x_i \quad (\text{A.20})$$

Sujeto a:

$$g_i(\vec{x}) = \frac{(x_i + x_{i+12})}{\sum_{j=124} x_j + e_i} \leq 0, \quad i = 1, 2, 3$$

$$g_i(\vec{x}) = \frac{(x_{i+3} + x_{i+15})}{\sum_{j=124} x_j + e_i} \leq 0, \quad i = 4, 5, 6$$

$$h_i(\vec{x}) = \frac{x_{i+12}}{b_{i+12} \sum_{j=1324} \frac{x_j}{b_j}} - \frac{c_i x_i}{40b_i \sum_{j=112} \frac{x_j}{b_j}} = 0, \quad i = 1, \dots, 12$$

$$h_{13}(\vec{x}) = \sum_{i=124} x_i - 1 = 0$$

$$h_{14}(\vec{x}) = \sum_{i=112} x_i + k \sum_{i=1324} \frac{x_i}{b_i} - 1,671 = 0$$

Donde  $k = (0,7302)(530)\left(\frac{14,7}{40}\right)$  y el conjunto de datos es mostrado en el tabla A.2. Los límites son  $0 \leq x_i \leq 10 (i = 1, \dots, 24)$ .

**g21**

Minimizar:

Cuadro A.2: Conjunto de datos del problema g20.

$i$	$a_i$	$b_i$	$c_i$	$d_i$	$e_i$
1	0.0693	44.094	123.7	31.244	0.1
2	0.0577	58.12	31.7	36.12	0.3
3	0.5	58.12	45.7	34.784	0.4
4	0.2	137.4	14.7	92.7	0.3
5	0.26	120.9	84.7	82.7	0.6
6	0.55	170.9	27.7	91.6	0.3
7	0.06	62.501	49.7	56.708	
8	0.1	84.94	7.1	82.7	
9	0.12	133.425	2.1	80.8	
10	0.18	82.507	17.7	64.517	
11	0.1	46.07	0.85	49.4	
12	0.09	60.097	0.64	49.1	
13	0.0693	44.094			
14	0.5777	58.12			
15	0.05	58.12			
16	0.2	137.4			
17	0.26	120.9			
18	0.55	170.9			
19	0.06	62.501			
20	0.1	84.94			
21	0.12	133.425			
22	0.18	82.507			
23	0.1	46.07			
24	0.09	60.097			

$$f(\vec{x}) = x_1 \tag{A.21}$$

Sujeto a:

$$\begin{aligned}
 g_1(\vec{x}) &= -x_1 + 35x_2^{0,6} + 35x_3^{0,6} \leq 0 \\
 h_1(\vec{x}) &= -300x_3 + 7500x_5 - 7500x_6 - 25x_4x_5 + 25x_4x_6 + x_3x_4 = 0 \\
 h_2(\vec{x}) &= 100x_2 + 155,365x_4 + 2500x_7 - x_2x_4 - 24x_4x_7 - 15536,5 = 0 \\
 h_3(\vec{x}) &= -x_5 + \ln(-x_4 + 900) = 0 \\
 h_4(\vec{x}) &= -x_6 + \ln(x_4 + 300) = 0 \\
 h_5(\vec{x}) &= -x_7 + \ln(-2x_4 + 700) = 0
 \end{aligned}$$

Donde los límites son  $-10 \leq x_1 \leq 1000$ ,  $0 \leq x_2, x_3 \leq 40$ ,  $100 \leq x_4 \leq 300$ ,  $6,3 \leq x_5 \leq 6,7$ ,  $5,9 \leq x_6 \leq 6,4$  y  $4,5 \leq x_7 \leq 6,5$ .

**g22**

Minimizar:

$$f(\vec{x}) = x_1 \tag{A.22}$$

Sujeto a:

$$\begin{aligned}
g_1(\vec{x}) &= -x_1 + x_2^{0,6} + x_3^{0,6} x_0,6_4 \leq 0 \\
h_1(\vec{x}) &= x_5 - 100000x_8 + 1?10^7 = 0 \\
h_2(\vec{x}) &= x_6 - 100000x_8 - 100000x_9 = 0 \\
h_3(\vec{x}) &= x^7 - 100000x_9 - 5?10^7 = 0 \\
h_4(\vec{x}) &= x_5 - 100000x_{10} - 3,3?10^7 = 0 \\
h_5(\vec{x}) &= x_6 - 100000x_{11} - 4,4?10^7 = 0 \\
h_6(\vec{x}) &= x_7 - 100000x_{12} - 6,6?10^7 = 0 \\
h_7(\vec{x}) &= x_5 - 120x_2x_{13} = 0 \\
h_8(\vec{x}) &= x_6 - 80x_3x_{14} = 0 \\
h_9(\vec{x}) &= x_7 - 40x_4x_{15} = 0 \\
h_{10}(\vec{x}) &= x_8 - x_{11} + x_{16} = 0 \\
h_{11}(\vec{x}) &= x_9 - x_{12} + x_{17} = 0 \\
h_{12}(\vec{x}) &= -x_{18} + \ln(x_{10} - 100) = 0 \\
h_{13}(\vec{x}) &= -x_{19} + \ln(-x_8 + 300) = 0 \\
h_{14}(\vec{x}) &= -x_{20} + \ln(x_{16}) = 0 \\
h_{15}(\vec{x}) &= -x_{21} + \ln(-x_9 + 400) = 0 \\
h_{16}(\vec{x}) &= -x_{22} + \ln(x_{17}) = 0 \\
h_{17}(\vec{x}) &= -x_8 - x_{10} + x_{13}x_{18} - x_{13}x_{19} + 400 = 0 \\
h_{18}(\vec{x}) &= x_8 - x_9 - x_{11} + x_{14}x_{20} - x_{14}x_{21} + 400 = 0 \\
h_{19}(\vec{x}) &= x_9 - x_{12} - 4,60517x_{15} + x_{15}x_{22} + 100 = 0
\end{aligned}$$

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

**g23**

Minimizar:

$$f(\vec{x}) = -9x_5 - 15x_8 + 6x_1 + 16x_2 + 10(x_6 + x_7) \quad (\text{A.23})$$

Sujeto a:

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

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:

$$f(\vec{x}) = -x_1 - x_2 \quad (\text{A.24})$$

Sujeto a:

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

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

Donde los límites son  $0 \leq x_1 \leq 3$  y  $0 \leq x_2 \leq 4$ .

# Bibliografía

- [1] Adil Baykasoglu, Lale Ozbakir, and Pinar Tapkan, *Swarm intelligence, focus on ant and particle swarm optimization*, ch. Artificial bee colony algorithm and its application to generalized assignment problem, pp. 113–144, I-Tech Education and Publishing, Vienna, Austria, December 2007.
- [2] Gerardo Beni and Jin Wang., *Self organizing sensory systems*, ch. Highly Redundant Sensing in Robotic Systems: Proceedings of NATO Advanced Workshop on Highly Redundant Sensing in Robotic Systems, pp. 251–262, Springer-Verlag, 1990.
- [3] Eric Bonabeu, Marco Dorigo, and Guy Theraulaz, *Swarm intelligence : From natural to artificial systems*, Oxford University Press, 1999.
- [4] Omar Cetina Domínguez, *Una adaptación del comportamiento de la abeja exploradora en el algoritmo de la colonia artificial de abejas para resolver problemas de optimización con restricciones*, Master's thesis, Laboratorio Nacional de Informática Avanzada, 2009.
- [5] Mauricio Damián Araoz, *Sistema visualizador de la colonia artificial de abejas*, Laboratorio Nacional de Informática Avanzada (LANIA A.C.), Enero 2010, Proyecto de residencias.
- [6] Kalyanmoy Deb, *An efficient constraint handling method for genetic algorithms*, ch. Computer Methods in Applied Mechanics and Engineering, pp. 186(2/4):311–338, 1998.
- [7] Agoston E. Eiben and J. E. Smith, *Introduction to evolutionary computing*, Springer, 2003.
- [8] Andries P. Engelbrecht, *Fundamentals of computational swarm intelligence*, John Wiley and Sons, 2005.
- [9] ———, *Computational intelligence, an introduction*, 3rd. edition ed., ch. 1. Introduction to Computational Intelligence, pp. 3–13, Wiley, 2007.
- [10] David B. Fogel, *An introduction to simulated evolutionary optimization*, pp. 5:3–14, Transactions on Neural Networks, IEEE, 1994.
- [11] kalyanmoy Deb, *Optimization for engineering design algorithms and examples*, Prentice-Hall, 2000.
- [12] Dervis Karaboga, *An idea based on honey bee swarm for numerical optimization*, Tech. report, Erciyes University, Engineering Faculty, 2005.
- [13] Dervis Karaboga and Bahriye Basturk, *Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems*, LCNS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing (2007), 789–798.
- [14] Saku Kukkonen and Jouni Lampinen, *Constrained real-parameter optimization with generalized differential evolution*, In Proceedings of IEEE Congress on Evolutionary Computation (Vancouver, BC, Canada), Institute of Electrical and Electronics Engineers (IEEE), 2006.

- [15] Jouko Lampinen, *In proceedings of the congression evolutionary computation 2002 (cec'2002)*, A constraint handling approach for the differential evolution algorithm (Piscataway, New Jersey) (IEEE Service Center, ed.), vol. 2, 2002, pp. 1468–1473.
- [16] J. J. Liang, Thomas Philip Runarsson, Efrén Mezura Montes, Maurice Clerc, P. N. Suganthan, Carlos A. Coello Coello, and Kalyanmoy Deb, *Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization*, Tech. report, School of IEEE, 2006.
- [17] Merriam-Webster, *Optimization*, <http://www.merriam-webster.com/dictionary/optimization>, 2011.
- [18] Efrén Mezura Montes, Omar Cetina Domínguez, and Betania Hernández Ocaña, *Nuevas heurísticas inspiradas en la naturaleza para optimización numérica*, (2010), 255–278.
- [19] Efrén Mezura Montes and Carlos 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 (CEC'2005), September 2005.
- [20] Zbigniew Michalewicz and David B. Fogel, *How to solve it: Modernheuristics*, second edition ed., Springer-Verlag, 2004.
- [21] Zbigniew Michalewicz and Marc Schoenauer, *Evolutionary algorithms for constrained parameter optimization problems*, pp. 1–32, Evolutionary Computation, 1996.
- [22] Singiresu S. Rao, *Engineering optimization, theory and practice*, 3rd. edition ed., ch. 1. Introduction to Optimization, pp. 1–64, Wiley and Sons, 1996.
- [23] Real Academia Española, *Optimizar*, <http://www.rae.es/>, 2011.
- [24] Hamdy A. Taha, *Investigación de operaciones*, séptima ed., Pearson Educación de México, 2004.
- [25] T. Takahama, S.Sakai, and Efrén Mezura Montes, *Constraint-handling in evolutionary optimization*, ch. Solving Difficult Constrained Optimization Problems by the epsilon Constrained Differential Evolution with Gradient-Based Mutation, Springer-Verlag, 2009.
- [26] B. Tessama and G. G. Yen, *A self adaptive penalty function based algorithm for constrained optimization*, In Proceedings of IEEE Congress on Evolutionary Computation 2006 (Vancouver, BC, Canada) (Institute of Electrical and Electronics Engineers (IEEE), eds.), July 2006.
- [27] Yong Wang, Zixing Cai, Guanqi Guo, and Yuren Zhou, *Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems*, IEEE Transactions on Systems, Man, and Cybernetics, Part B **37** (2007), no. 3, 560–575.
- [28] Yong Wang, Zixing Cai, Yuren Zhou, and Wei Zeng, *An adaptive tradeoff model for constrained evolutionary optimization*, IEEE Transactions on Evolutionary Computation **12** (2008), no. 1, 80–92.