

REAL POTENCIAL DEL USO DEL MÉTODO DE ANÁLISIS DE INTERVALOS
PARA LA OPTIMIZACIÓN CON RESTRICCIONES FRENTE AL PSO DE
CONVERGENCIA GARANTIZADA

KAROLSEBASTIÁN BARRAGÁN NIÑO
DIEGO ARMANDO VANEGAS ARDILA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECÁNICAS
ESCUELA DE INGENIERIAS ELECTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
2010

REAL POTENCIAL DEL USO DEL MÉTODO DE ANÁLISIS DE INTERVALOS
PARA LA OPTIMIZACIÓN CON RESTRICCIONES FRENTE AL PSO DE
CONVERGENCIA GARANTIZADA

KAROLSEBASTIÁN BARRAGÁN NIÑO
DIEGO ARMANDO VANEGAS ARDILA

Trabajo de grado

Director
Rodrigo Correa
Profesor E3T

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECÁNICAS
ESCUELA DE INGENIERIAS ELECTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
2010

DEDICATORIA

A Dios por darme el privilegio de disfrutar este momento al lado de mis seres queridos.

A mi madre, esperanza, por ser mi motivo más grande de inspiración y la mujer que cada día me enseña que las mejores cosas de la vida se consiguen con esfuerzo y dedicación.

A mi padre, Alfredo, por ser un padre de vida, justo, honesto y un digno ejemplo de rectitud y entrega.

A mi mama Matilde, aquella que con sus consejos, virtudes y esa combinación perfecta entre sabiduría y gracia ha fortalecido mi paso a la adultez.

A mis hermanos Vanegas y mis hermanos Capacho, por estar ahí siempre, en cada éxito y en cada fracaso y darme fortaleza para no dejarme vencer por las adversidades.

A mis familiares y amigos por compartir mis alegrías y brindarme todo su apoyo en cada etapa de mi vida.

Diego Armando Vanegas Ardila

A la virgen María por brindarme la luz en el camino toda mi vida

A mis padres Miguel y Luzmila por su amor, confianza y sacrificio

A mi nonito Miguel por ser tan especial y por encomendarme tanto a Dios en sus oraciones

A mi hermana Juliana por estar pendiente de mí y apoyarme cuando más fuerza necesitaba

A mi novia Danita por quererme tanto y brindarme el tiempo para lograrlo

A mis amigos y familiares por su compañía y buenos deseos

Karolsebastian Barragán Niño

AGRADECIMIENTOS

A nuestro director Rodrigo Correa por brindarnos la guía y el apoyo necesario para poder cumplir satisfactoriamente con el desarrollo de este proyecto de grado.

Al señor Tiago Montanher de la Universidad de Sao Paulo por facilitarnos su ayuda desinteresada y proveernos herramientas computacionales para la realización de este trabajo.

Al profesor Siegfried M. Rump porque nuestro trabajo se pudo realizar gracias al software INTLAB, del cual fue desarrollador.

A la Universidad Industrial de Santander por ofrecernos los recursos bibliográficos y el acceso a la base de datos para la consulta de información valiosa.

A todas aquellas personas referenciadas en el contenido del texto, por suministrar la implementación de herramientas de software y métodos de optimización de manera gratuita.

CONTENIDO

	Pág.
INTRODUCCIÓN	16
1. OPTIMIZACIÓN GLOBAL	18
2. OPTIMIZACIÓN POR ENJAMBRE DE PARTICULAS (PSO)	22
2.1 HISTORIA Y DESARROLLO DE PSO	22
2.2 CLASIFICACION Y VARIANTES DE PSO	24
3. OPTIMIZACIÓN GLOBAL POR ANÁLISIS DE INTERVALOS	30
3.1 HISTORIA E IMPORTANCIA DEL ANÁLISIS DE INTERVALOS	30
3.2 FUNDAMENTOS Y PRINCIPIOS DEL ANÁLISIS DE INTERVALOS	32
3.3 MÉTODO DE INTERVALOS DE NEWTON MULTIVARIABLE.	34
4. DESARROLLO DE ALGORITMOS PARA PSO Y ANALISIS DE INTERVALOS.	37
4.1 DESARROLLO DEL ALGORITMO PSO	37
4.2 DESARROLLO DEL ALGORITMO DE OPTIMIZACIÓN POR ANÁLISIS DE INTERVALOS	40
5. ESTUDIO DEL COMPORTAMIENTO DE PSO CON RESTRICCIONES	46
5.1 FUNCIONES DE PRUEBA	46
5.3 EXPERIMENTO 2 PSO: VARIACIÓN CANTIDAD DE ITERACIONES.	53
5.4 EXPERIMENTO 3 PSO: COMPORTAMIENTO DEL FACTOR INERCIA Y EL FACTOR DE ESTRECHAMIENTO.	55
5.5 EXPERIMENTO 4 PSO: COMPORTAMIENTO DEL ALGORTIMO AL AUMENTAR EL NÚMERO DE VARIABLES.	59
5.6 ESTUDIO ESTADÍSTICO PSO	60
6. ESTUDIO COMPARATIVO ENTRE EL METODO DE OPTIMIZACION POR ENJAMBRE DE PARTÍCULAS (PSO) Y EL METODO DE OPTIMIZACION DE ANÁLISIS DE INTERVALOS.	69
CONCLUSIONES	81

TRABAJO FUTURO	83
REFERENCIAS BIBLIOGRÁFICAS	84
ANEXOS	90

LISTA DE TABLAS

	Pág.
Tabla 1. Conceptos básicos en aritmética de intervalos	32
Tabla 2. Operaciones básicas en aritmética de intervalos	33
Tabla 3. Funciones objetivo y restricciones	46
Tabla 4. Variación de la cantidad de partículas para las funciones de prueba y resultados obtenidos con 200 iteraciones máximas.	51
Tabla 5. Variación de la cantidad de iteraciones para las funciones de prueba y resultados obtenidos una población de $N=200$ partículas.	53
Tabla 6. Promedio límite de iteraciones en la evaluación de la función y las restricciones para cada función objetivo.	55
Tabla 7. Variantes de PSO con factor de inercia y factor de estrechamiento. Estudio estadístico de 33 ejecuciones independientes.	58
Tabla 8. Variación de los valores óptimos de PSO en F5 al cambiar el número de variables.	59
Tabla 9. Estudio característico de ventajas y desventajas de PSO y la optimización por análisis de intervalos.	69
Tabla 10. Resultados de la comparación de los métodos de PSO y análisis de intervalos para 1 variable.	71
Tabla 11. Estadísticas de 33 ejecuciones independientes para los métodos de PSO y análisis de intervalos para 2 variables.	72
Tabla 12. Estadística de 33 ejecuciones independientes para los métodos de PSO y análisis de intervalos para F1.	74

LISTA DE FIGURAS

	Pág.
Figura 1. Esquema de optimización global	21
Figura 2. Ejemplo del comportamiento en una bandada de aves.	23
Figura 3. Modelos de comunicación básicos en PSO.	25
Figura 4. Representación gráfica del comportamiento de una partícula	27
Figura 5. Valores de intervalos de $F(X)$ en 2 subregiones	41
Figura 6. Comparación entre f_m y f_{der}	41
Figura 7. Algoritmo de optimización con restricciones por análisis de intervalos	45
Tabla 3. Funciones objetivo y restricciones	46
Figura 8. Función objetivo F2 con la restricción respectiva de la tabla 3.	48
Figura 9. Función objetivo F3 con la restricción respectiva de la tabla 3.	48
Figura 10. Función objetivo F4 con la restricción respectiva de la tabla 3.	49
Figura 11. Función objetivo F5 con la restricción respectiva de la tabla 3.	49
Figura 12. Función objetivo F6 (Goldstein & Price)	50
Figura 13. Función objetivo F7	50
Figura 14. Función objetivo F8	51
Figura 15. Decrecimiento lineal del factor de inercia vs. Cantidad de generaciones para F2	57
Figura 16. Decrecimiento lineal del factor de inercia vs. Cantidad de generaciones para F3	57
Figura 17. Media aritmética de fitness poblacional vs. Generación actual para F1	61
Figura 18. Desviación estándar de fitness poblacional vs. Generación actual para F1	62
Figura 19. Error medio cuadrático de fitness poblacional vs. Generación actual para F1	63
Figura. 20. Distribución del mejor fitness en la última generación.	63
Figura 21. Distribución de las posiciones de las partículas en el enjambre desde las posiciones iniciales hasta las posiciones de los valores óptimos (actuales).	64
Figura 22. Media aritmética de fitness poblacional vs. Generación actual para F7	64
Figura 23. Desviación estándar de fitness poblacional vs. Generación actual para F7	65
Figura 24. Error medio cuadrático de fitness poblacional vs. Generación actual para F7	66
Figura 25. Distribucion del mejor fitness en la generación final para F7.	66

Figura 26. Media aritmética de fitness poblacional vs. Generación actual para F8	67
Figura 27. Desviación estándar de fitness poblacional vs. Generación actual para F8	67
Figura 28. Error medio cuadrático de fitness poblacional vs. Generación actual para F8	68
Figura 29. División del dominio original mediante las pruebas 1 y 2 para F3.	75
Figura 30. Valores estadísticos medios para F3.	75
Figura 31. División del dominio original mediante las pruebas 1 y 2 para F4.	76
Figura 32. Valores estadísticos medios para F4.	76
Figura 33. División del dominio original mediante las pruebas 1 y 2 para F5.	77
Figura 34. Valores estadísticos medios para F5	77
Figura 35. División del dominio original mediante las pruebas 1 y 2 para F6	78
Figura 36. División del dominio original mediante las pruebas 1 y 2 para F7.	78
Figura 37. Valores estadísticos medios para F7	79

LISTA DE ANEXOS

	Pág.
ANEXO A. Gráficas de valores medios, desviación estándar y error medio cuadrático de las funciones objetivo.	90
ANEXO B. GUIA DE USO INTERFACES	100

RESUMEN

TITULO: REAL POTENCIAL DEL USO DEL MÉTODO DE ANÁLISIS DE INTERVALOS PARA LA OPTIMIZACIÓN CON RESTRICCIONES FRENTE AL PSO DE CONVERGENCIA GARANTIZADA*

AUTORES: DIEGO ARMANDO VANEGAS ARDILA, KAROLSEBASTIAN BARRAGAN NINO**

PALABRAS CLAVES: optimización global, restricciones, optimización por enjambre de partículas, aritmética de intervalos, análisis de intervalos.

El presente documento describe los resultados obtenidos de la comparación del algoritmo de optimización mediante enjambre de Partículas (PSO) y el método de optimización por análisis de intervalos para resolver problemas de optimización en funciones no lineales con restricciones de igualdad y/o desigualdad.

El principal objetivo fue dejar una fundamentación apropiada para futuros trabajos de grado en este campo a través de la implementación y comparación adecuada de los métodos. La importancia de la realización del documento radicó principalmente en la expansión de los conceptos de optimización a diversas ramas de la ingeniería. Parámetros en PSO como la cantidad de partículas, el factor de inercia y el factor de estrechamiento fueron empleados en diversos estudios comparativos en este documento para verificar los principios y el comportamiento propio del método de optimización por enjambre de partículas (PSO) frente al método de optimización por análisis de intervalos. El problema que se abordó fue la optimización global con restricciones. Para el manejo de restricciones en PSO se utilizó un mecanismo simple basado en reglas de factibilidad. El método de optimización por análisis de intervalos estudiado en el documento actual fue propuesto por K. Ichida. Este método se utilizó para encontrar el máximo global de una función multimodal de máximo tres variables sujeta a restricciones de igualdad y/o desigualdad. Al descartar subregiones donde la solución global puede no existir, se pudo encontrar la solución con un límite de error riguroso.

Así, en la fase experimental se utilizaron nueve funciones de prueba con restricciones para realizar el estudio estadístico de los métodos de optimización. Para evaluar dicho rendimiento, se mostraron las estadísticas de 33 ejecuciones independientes de cada variante en las funciones de prueba. Una revisión comparativa de ventajas y desventajas de cada método es planteada al final del documento como base para trabajos futuros sobre el tema.

* Trabajo de grado

** Facultad de Ingenierías Físico-mecánicas, Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Rodrigo Correa

ABSTRACT

TITLE: REAL POTENTIAL OF USING THE METHOD OF INTERVAL ANALYSIS FOR CONSTRAINED OPTIMIZATION AGAINST THE GUARANTEED CONVERGENCE PSO*

AUTHORS: DIEGO ARMANDO VANEGAS ARDILA, KAROLSEBASTIAN BARRAGAN NINO**

KEYWORDS: global optimization, constraints, particle swarm optimization, interval arithmetic, interval analysis.

This paper describes the results of the comparison algorithm using particle swarm optimization (PSO) and the optimization method using interval analysis for solving nonlinear optimization functions with equality and / or inequality constraints.

The main objective was to make a proper foundation for future work in this field through the implementation and comparison of methods. The importance of completing the document is mainly the expansion of optimization concepts to various branches of engineering. PSO parameters as the number of particles, the inertia factor and narrowing factor were used in comparative studies in this document to verify the principles and proper behavior of the method of particle swarm optimization (PSO) versus method interval analysis optimization. The problem addressed was global optimization with constraints. For constraint handling, PSO used a simple mechanism based on feasibility rules. The optimization method for interval analysis studied in the current document was proposed by K. Ichida. This method was used to find the global maximum of a multimodal function of at most three variables subject to equality and / or inequality constraints. By discarding subregions where the global solution may not exist, the solution could be found with a rigorous error bound.

Thus, in the experimental phase using nine test functions with restrictions to perform the statistical study of optimization methods. To evaluate this performance, showed the statistics of 33 independent executions of each variant in the test functions. A comparative review of advantages and disadvantages of each method is put to the end of the document as a basis for future work on the subject.

* Degree work

** Facultad de Ingenierías Físico-mecánicas, Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Rodrigo Correa.

INTRODUCCIÓN

En la mayoría de los usos de ingeniería los problemas de optimización se presentan muy seguidos, lo que le da mucha importancia a este tipo de técnicas. Un aporte enorme se ha realizado para crear técnicas de optimización, analíticas y numéricas que sean aplicables a diversos campos con la mayor eficiencia posible. Por ello los distintos métodos de optimización juegan un papel importante en la actualidad. Aunque ahora existen muchas técnicas, todavía hay una innumerable cantidad de funciones que están más allá de los métodos analíticos y que presentan dificultades significativas para los procesos numéricos. Como consecuencia, hay una continua búsqueda de nuevas y más robustas técnicas de optimización, que puedan solucionar tales problemas.

En la literatura especializada se encuentran gran variedad de trabajos referentes al problema de optimización, utilizando distintas técnicas que van desde técnicas exactas como es el caso de partición por análisis de intervalos, hasta métodos aproximados como es el caso de los meta-heurísticos optimización por enjambre de partículas (PSO) y algoritmos genéticos. Algunos de estos trabajos sirven como base para obtener nuevos métodos derivados, pero sin tener la certeza de que realmente provean un mejor desempeño. Debido a que se aborda la optimización global con restricciones, contrastar el método de análisis de intervalos frente a métodos tradicionalistas como PSO, representa una información necesaria a la hora de obtener un mejor desempeño en sistemas con restricciones. Una razón importante para optar por estos mecanismos de optimización es debido a que ambos fueron publicados aproximadamente en la misma época (después de mitad del siglo xx) pero que rápidamente por su fácil implementación algorítmica se ha aumentado el número de trabajos referentes a PSO y por otro lado no ha sido tan profunda la investigación referente al método analítico de intervalos, el cual es eficiente para el manejo de restricciones ya que emplea los principios del método de bisección para encontrar óptimos globales.

En esta época donde es cada vez más y más importante lograr la mayor eficiencia en el funcionamiento de los sistemas, resulta valioso contar con herramientas óptimas en el manejo de la información que permitan ahorrar costos, tiempo y manejo de datos. Con esta premisa como motor principal se creyó importante realizar una validación de los métodos propuestos de PSO y método de optimización por análisis de intervalos para establecer un criterio básico de escogencia a la hora de analizar problemas de ingeniería cuya solución necesita de un sistema de optimización. El principal objetivo entonces, fue dejar una fundamentación apropiada para futuros trabajos de grado y proyectos de investigación en este campo a través de la implementación y comparación adecuada de métodos de amplio uso en las ciencias aplicadas.

1. OPTIMIZACIÓN GLOBAL

En la vida cotidiana, continuamente se presentan problemas de optimización que se resuelven utilizando la lógica y pequeñas fórmulas matemáticas. Pero si los problemas presentan alta complejidad, se recurre al uso de computadores y máquinas de alta tecnología. La optimización global continua y los métodos de búsqueda globales asociados siempre han sido una parte importante del vivir exitosamente.

La optimización global es el proceso de encontrar el mejor conjunto de condiciones aceptables que logren un objetivo bajo ciertas restricciones, asumiendo que tanto el objetivo a conseguir como las restricciones se encuentran ambos en términos matemáticos.

Una referencia básica en la mayoría de los aspectos de la optimización global es The Handbook of Global Optimization [1]. Un punto de análisis complejo se puede ver en [2] y [3].

Los algoritmos presentes en optimización se pueden ser clasificar de acuerdo al grado de rigor con el cual se aproximan al objetivo [1]:

- Un método incompleto utiliza heurística inteligente intuitiva para realizar la búsqueda pero garantiza si la búsqueda queda atrapada en un óptimo local y pueda no encontrar el óptimo deseado.
- Un método asintóticamente completo alcanza un óptimo global con certeza o al menos con una probabilidad si se le permite correr por un tiempo largo, pero no tiene medios para saber cuándo un mínimo global ha sido encontrado
- Un método completo alcanza un óptimo global con certeza asumiendo cálculos exactos y tiempo de corrida indefinido y conociendo después de un tiempo finito que un óptimo global aproximado ha sido encontrado con las tolerancias establecidas.
- Un método riguroso alcanza un óptimo global con seguridad y dentro de las tolerancias dadas, incluso en la presencia de errores de redondeo.

El método completo más simple para problemas con límites de restricciones es la búsqueda por regiones, donde todos los puntos sobre las regiones cada vez más finas son examinados y el mejor punto sobre cada región es usado como punto de inicio para ubicar óptimos locales. Ya que el número de puntos sobre la región crece de manera exponencial con las cantidad de dimensiones, la búsqueda por regiones solo es eficiente para una o dos dimensiones. Métodos completos más eficientes combinan técnicas de ramificaciones con una o varias técnicas de optimización local, análisis de intervalos y programación con restricciones.

Generalmente los métodos completos son más confiables que los métodos incompletos puesto que la forma en que ellos trabajan (dificultad del problema) constituyen garantías de

buenos resultados. Los métodos completos con terminación finita requieren acceso detallado a la información global del problema. En la mayoría de los códigos esto se logra mediante aritmética de intervalos.

Una solución básica a casi todos los algoritmos de optimización global completa es el principio de ramificación. Esta técnica consiste en dividir (ramificar) el problema original recursivamente en sub problemas que tarde o temprano, son fáciles de resolver. En los métodos de ramificación pura, las ramas más probables se dividen con mayor frecuencia.

Uno de los principales objetivos de cualquier Ingeniería y ciencia en general debe ser el uso de métodos exactos y heurísticos para optimizar funciones objetivo y generar mayores beneficios en costos y eficiencia en los procesos. La optimización de estos problemas parte de un conjunto de datos y una serie de condiciones y limitaciones que dificultan en algunos casos la utilización de métodos exactos. La dificultad se presenta principalmente por la alta complejidad de los cálculos y la duración de éstos, ya que en algunas ocasiones el tiempo de resolución puede ser limitado.

Para la optimización de problemas y cálculos de alta complejidad se han desarrollado múltiples técnicas y métodos, que se pueden clasificar en técnicas exactas y aproximadas.

Las técnicas exactas garantizan encontrar la solución óptima de cualquier problema pero en una cantidad de tiempo no deseable en materia de gasto computacional.

Las técnicas aproximadas sacrifican la garantía de encontrar el resultado óptimo a cambio de obtener una buena solución en un tiempo razonable. Estas técnicas se han venido desarrollando durante los últimos 30 años y se distinguen tres tipos: métodos constructivos, métodos de búsqueda local y las técnicas metaheurísticas [1].

- Los métodos constructivos suelen ser los más rápidos y sus soluciones de muy baja calidad. Su planteamiento depende en gran parte del tipo de problema. Es muy difícil encontrar métodos de esta clase que produzcan buenas soluciones.
- Los métodos de búsqueda local usan el concepto de vecindario y se inician con una solución completa recorriendo parte del espacio de búsqueda hasta encontrar un óptimo local. Estos métodos parten de una solución inicial, examinan su vecindario y eligen el mejor vecino continuando el proceso hasta que encuentran un óptimo local. En función del operador de

movimiento utilizado, el vecindario cambia y el modo de explorar el espacio de búsqueda también, pudiendo la búsqueda complicarse o simplificarse.

- Las técnicas metaheurísticas son algoritmos no exactos y se fundamentan en la combinación de diferentes métodos heurísticos a un nivel más alto para conseguir una exploración del espacio de búsqueda más eficiente.

Hay diferentes formas de clasificar las técnicas metaheurísticas: basadas en la naturaleza (algoritmos bio-inspirados) o no basadas en la naturaleza, basadas en memoria o sin memoria, con función objetivo estática o dinámica, etc.

La clasificación más empleada es la que se basa en si la técnica utiliza un único punto del espacio de búsqueda o trabaja sobre un conjunto o población. Según esta clasificación las técnicas metaheurísticas se dividen en las basadas en trayectoria y las basadas en población.

Algunas de las técnicas metaheurísticas basadas en trayectoria son: el enfriamiento simulado (ES) [4], la Búsqueda Tabú (BT) [5], [6] la Búsqueda en Vecindario Variable (BVV) [7], la Búsqueda Local Iterada (BLI) [8] . Algunas técnicas metaheurísticas basadas en población son: los Algoritmos Evolutivos (AE) [9], [10],[11], la Búsqueda Dispersa (BD), los sistemas basados en Colonias de Hormigas, (ACO) [12] y [13] y los algoritmos basados en nubes de partículas o Particle Swarm Optimization (PSO) [14].

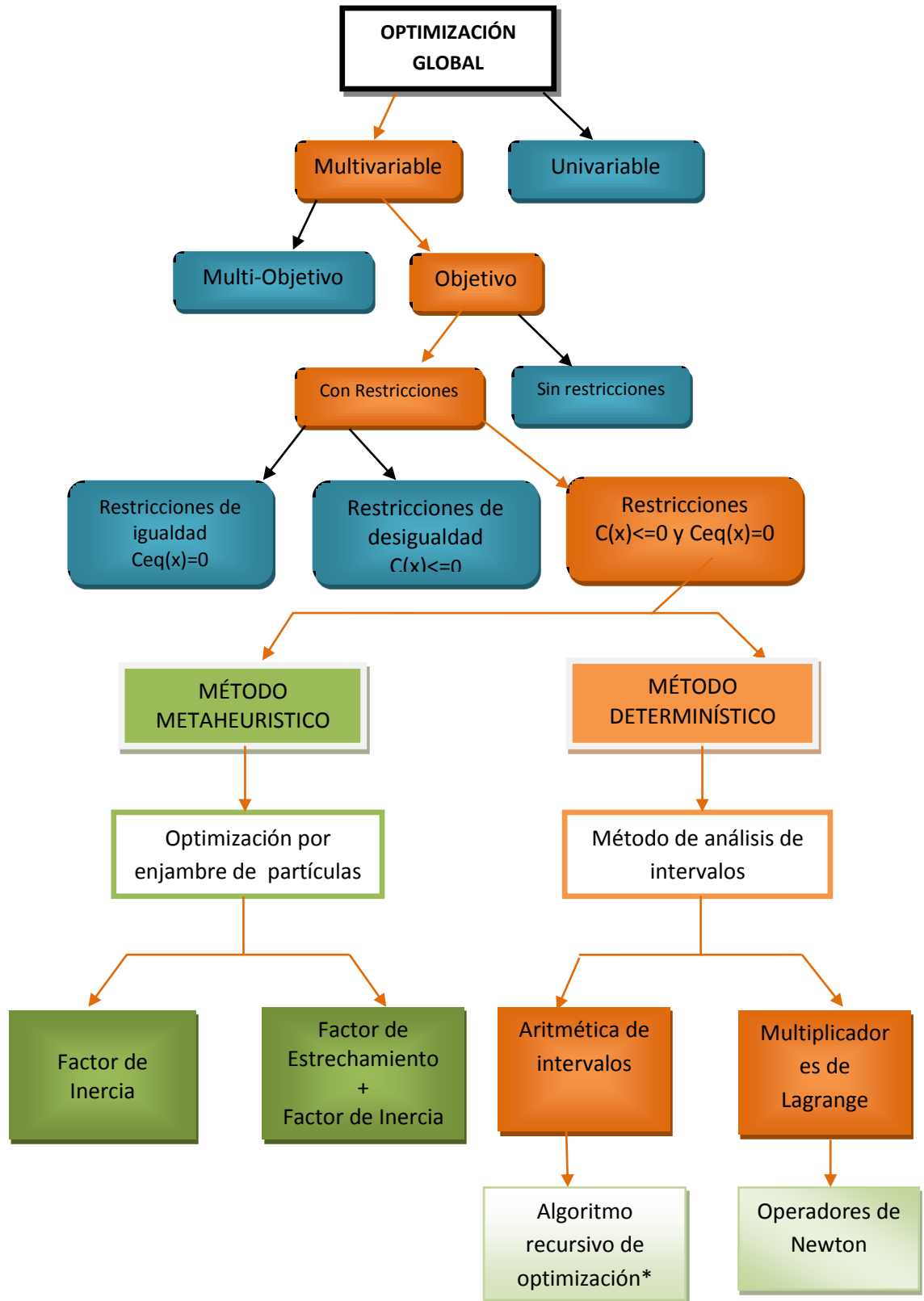
Los algoritmos evolutivos se pueden clasificar en las siguientes tres categorías: Programación Evolutiva (PE) [11] , Estrategias Evolutivas (EE) [15], [16] , [17] y los Algoritmos Genéticos (AG), que constituyen una de las técnicas más conocidas [18], [19] y [20] que fueron introducidos por Holland [21].

Las técnicas metaheurísticas se caracterizan por las siguientes propiedades:

- Son estrategias generales que guían el proceso de búsqueda.
- El objetivo es una búsqueda eficiente que encuentre soluciones casi-óptimas.
- Pueden incorporar mecanismos para evitar la exploración en regiones del espacio de búsqueda no óptimas.
- El procedimiento de cualquier metaheurística no depende del problema a resolver.
- Las metaheurísticas utilizan métodos heurísticos específicos que son controlados por una estrategia de alto nivel.
- Las metaheurísticas cuantifican el grado de adecuación de una determinada solución.

La Figura 1 ilustra el problema de optimización planteado en este documento junto con la respectiva clasificación de los métodos de optimización global.

Figura 1. Esquema de optimización global



2. OPTIMIZACIÓN POR ENJAMBRE DE PARTICULAS (PSO)

2.1 HISTORIA Y DESARROLLO DE PSO

La Optimización mediante enjambres de Partículas es una heurística desarrollada por Kennedy y Eberhart [22], y ha sido empleada para resolver problemas de optimización con y sin restricciones. Denominada PSO por sus siglas en inglés (Particle Swarm Optimization), PSO tiene sus orígenes en la simulación de comportamientos sociales usando herramientas graficas computacionales y expandiendo estos resultados al campo de la psicología social.

Dentro del campo de los gráficos por ordenador, los primeros antecedentes de la optimización por enjambre de partículas se atribuye a Reeves [23], quien propuso sistemas de partículas para objetos de modelos dinámicos que no podían ser fácilmente representados por figuras o superficies, como por ejemplo el agua y las nubes. En estos sistemas, las partículas son independientes unas de otras y rigen su movimiento por un conjunto de reglas. Algunos años después, Reynolds [24], utilizó un sistema de partículas para representar el comportamiento social de una bandada de pájaros. En el mismo tipo de simulación, Heppner y Grenander [25] incluyeron un gallinero como sistema de simulación con aves. Ambos modelos inspiraron el conjunto de reglas que serían utilizadas más adelante en el algoritmo original de optimización por enjambre de partículas (PSO).

La investigación social en psicología [26] fue otra fuente de inspiración en el desarrollo del primer algoritmo de optimización por enjambre de partículas [27]. Las reglas que rigen el movimiento de las partículas en el espacio de búsqueda de un problema también pueden ser vistas como un modelo de comportamiento social humano en el que los individuos ajustan sus actitudes conforme a las de sus compañeros [22].

La primera aplicación práctica de PSO fue en el campo de la formación de redes neuronales y fue reportada con el mismo algoritmo original [28].

Al ser PSO una técnica metaheurística que busca soluciones buenas (óptimas o casi óptimas) a un coste computacional razonable, no garantiza la factibilidad de las mismas. En algunos casos ni siquiera puede determinar la cercanía al óptimo de una solución factible.

En resumen, una metaheurística es una estrategia general de alto nivel que usa diferentes métodos heurísticos para explorar en busca de una solución óptima o

casi óptima en espacios de búsqueda de tamaño considerable. La metaheurística debe identificar rápidamente las regiones satisfactorias del espacio de búsqueda global y no gastar tiempo en regiones que hayan sido exploradas o no contengan soluciones de alta calidad.

PSO es una técnica fácil de implementar y de funcionamiento sencillo que puede ser incluida dentro del área de inteligencia en cúmulos. PSO es ideado como un algoritmo bio-inspirado por tener sus orígenes en la idea básica del comportamiento social de las bandadas de pájaros o bancos de peces (fig. 2). En estos grupos animales se establecen relaciones sociales entre los individuos del grupo existiendo un líder reconocido el cual es seguido por los individuos del grupo. El líder es un individuo que tiene destrezas que le permiten mantener un control sobre los demás en el grupo y guiar las diferentes actividades de búsqueda de alimento y mejores condiciones.

Figura 2. Ejemplo del comportamiento en una bandada de aves.



Figura editada de la tesis de doctorado 'Sistema de generación eléctrica con pila de combustible de óxido sólido alimentado con residuos forestales y su optimización mediante algoritmos basados en nubes de partículas'. Capítulo 3.

Los individuos confían en la capacidad del líder para dirigirlos, pero este líder puede cambiar muy fácilmente si surge un individuo con mejores habilidades que el líder anterior, reemplazándolo en la tarea de guiar al grupo. Así, cada individuo

dentro del grupo ve influenciado su comportamiento por dos factores: El primero es el conocimiento y habilidades propias obtenidas durante su vida y el segundo es la influencia del líder del grupo. Este comportamiento es ajustado para resolver problemas de optimización numérica.

En PSO se tiene un enjambre de partículas donde cada una de ellas representa una posible solución al problema de optimización. Cada partícula tiene una memoria que le permite recordar la mejor posición por donde ha pasado. Además, cada una de ellas tiene asociada una velocidad que le sirve para definir hacia dónde se moverá en cada vuelo. La actualización de la velocidad se calcula de acuerdo a 4 factores: la velocidad actual, la posición actual de la partícula, su valor en memoria y la posición del líder del grupo. De este modelo se obtienen variantes que se diferencian en la forma de actualizar la velocidad de las partículas y en la forma de comunicación entre ellas.

Las principales características del algoritmo PSO son las siguientes:

- En PSO las partículas (agentes de búsqueda) intercambian información. Estas modifican su dirección en función de las direcciones de las partículas de su vecindario (PSO local) o el líder del grupo (PSO global).
- PSO almacena la información propia de cada partícula. La partícula decide su nueva dirección en función de la mejor posición por la que pasó anteriormente.
- Suele tener una convergencia rápida a buenas soluciones.
- La población del algoritmo se inicia de manera aleatoria.
- La población evoluciona con cada iteración.
- La búsqueda gira en torno a la solución más óptima posible.
- La búsqueda se basa exclusivamente en los valores de la función objetivo.
- PSO no crea nuevas partículas en el proceso de búsqueda del óptimo, ya que las partículas iniciales son las que se modifican durante todo el proceso.

2.2 CLASIFICACION Y VARIANTES DE PSO

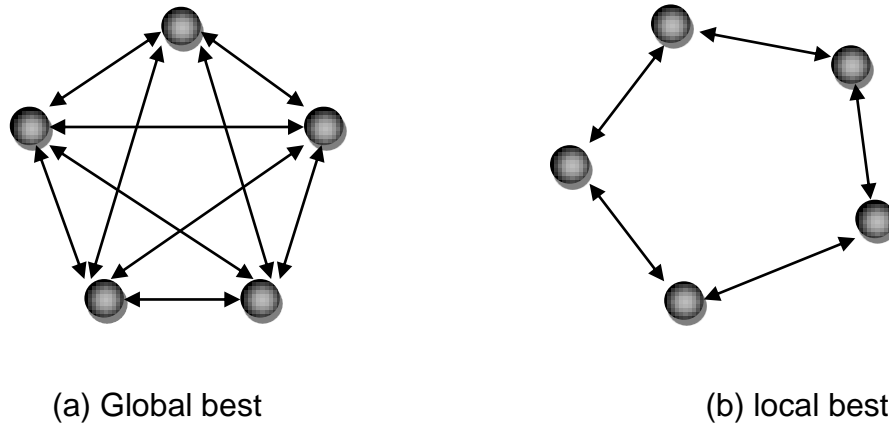
Los problemas de optimización de funciones no-lineales con restricciones constan de tres componentes básicos: un conjunto de variables, una función objetivo a ser optimizada (minimizar o maximizar) y un conjunto de restricciones que especifican los espacios factibles de las variables. Dentro del problema de optimización con restricciones mediante PSO se pueden presentar ciertas variantes presentadas a continuación:

Variantes de acuerdo con la comunicación entre partículas: según el tipo de vecindad utilizada, es decir, la cantidad y posición de las partículas que intervienen en el cálculo de la distancia en la componente social (fig. 3):

- PSO Local, cuyo entorno consiste en las partículas inmediatamente cercanas.
- PSO Global, cuyo entorno consiste en todas las partículas del enjambre.

En el caso particular estudiado en este documento solamente se trabajó con PSO global con restricciones.

Figura 3. Modelos de comunicación básicos en PSO.



Variantes de acuerdo al cálculo de velocidad

De acuerdo a la forma en la cual se calcula la velocidad en PSO se pueden obtener dos clases, una variante que depende del factor de inercia y una variante que depende del factor de estrechamiento.

PSO con valor de inercia: en esta variante PSO determina la nueva posición de cada partícula a partir una función de vuelo, actualizando su posición después de haber calculado la nueva velocidad. La velocidad representa en sí una dirección de búsqueda. En su concepción original, PSO no contaba con el valor de inercia [28], posteriormente se introdujo el valor de inercia [29] para reducir la influencia de la dirección de búsqueda que trae la partícula. La fórmula para calcular la velocidad de la partícula es la siguiente:

$$v(t+1) = w * v(t) + c_1 * rand() * (pBest - x) + c_2 * Rand() * (pG - x) \quad (1)$$

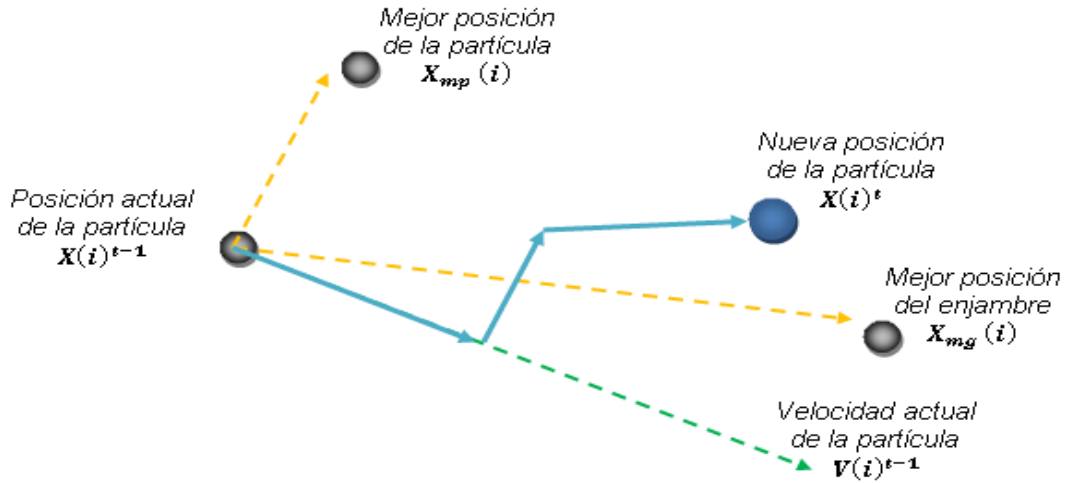
La nueva velocidad ($v(t + 1)$) de la partícula es determinada en su forma general por los siguientes parámetros:

- Factor de inercia “ w ”. Regula la influencia de la velocidad anterior de la partícula $v(t)$ en el cálculo de la nueva velocidad $v(t + 1)$. El factor de inercia beneficia la convergencia del enjambre.
- $v(t)$: velocidad actual (dirección de vuelo de la partícula). La velocidad inicial de cada partícula es cero.
- Coeficientes de aceleración ($c_1 * rand()$ y $c_2 * Rand()$). $Rand()$ y $rand()$ representan un número real aleatorio con distribución uniforme entre 0 y 1. El coeficiente c_1 regula la influencia del conocimiento cognitivo de la partícula (influencia que tiene la mejor posición alcanzada por la partícula ($pBest$) para guiar su nueva dirección) y el coeficiente c_2 regula la influencia del conocimiento social (influencia del líder del grupo (pG) en la dirección de búsqueda de la partícula).
- $pBest$. Representa la memoria de la partícula (mejor posición alcanzada hasta esa iteración)
- pG . Representa la mejor posición del grupo (la posición del líder).
- x . Representa la posición actual de la partícula.
- $v(t + 1)$. Velocidad actualizada de la partícula que determinará la nueva dirección de búsqueda de la partícula en la iteración siguiente. Con base a esto la nueva posición de la partícula x se calcula mediante:

$$x = x + v(t + 1). \quad (2)$$

La Figura 4 representa la forma en que la partícula interactúa para calcular su nueva posición dentro del enjambre.

Figura 4. Representación gráfica del comportamiento de una partícula



PSO con factor de estrechamiento. Propuesto por Parsopoulos y Vrahatis en [30]. La variante con el factor de estrechamiento representado por K ("constriction factor") afecta a la velocidad actualizada calculada en lugar de afectar solo la velocidad actual. La fórmula para la velocidad modificada es la siguiente:

$$v(t+1) = K * [v(t) + c_1 * rand() * (pBest - x) + c_2 * Rand() * (pG - x)], \quad (3)$$

donde el factor de estrechamiento (K) es calculado mediante los factores de aceleración c_1 y c_2 . La variante con factor de estrechamiento ofrece un mejor control de la velocidad y asegura la convergencia para problemas de optimización numérica con y sin restricciones. El trabajo hecho por Clerk [31] indica que el uso de un factor de estrechamiento puede ser necesario para asegurar la convergencia de un algoritmo de enjambre de partículas. Un método simplificado de incorporar este factor se muestra en la siguiente ecuación:

$$K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \quad \text{donde } \varphi = c_1 + c_2, \quad \varphi > 4 \quad (4)$$

Típicamente cuando el método de factor de estrechamiento de Clerk es usado, φ es ajustado a 4.1 y la constante de estrechamiento K es igual a 0.729.

Optimización global con restricciones

Muchos problemas de optimización, con importancia tanto teórica como práctica, consisten en la búsqueda de la mejor configuración de un conjunto de variables para conseguir algunos objetivos. Más formalmente, la optimización con

restricciones en su forma más general es conocida como el problema general de programación no-lineal que puede definirse como sigue:

optimizar $f(\mathbf{x})$, sujeto a:

$$g(\mathbf{x}) \leq 0; i = 1, 2, \dots, m \quad (5)$$

$$h(\mathbf{x}) \leq 0; i = 1, 2, \dots, l \quad (6)$$

Donde \mathbf{x} es el vector de variables de decisión ($[x_1, x_2, \dots, x_d]$) las cuales tienen un rango inferior (L_j) y superior (U_j). La zona factible (F) representa el vector solución que cumple todas las restricciones, y se encuentra en el espacio de búsqueda (S) que pertenece a un espacio de continuo de n dimensiones dentro del rango de variables.

Dentro de la literatura especializada cabe resaltar que PSO ha sido estudiado a cabalidad por muchos autores para los problemas de optimización de funciones no-lineales. En [4] Toscano y Coello plantean un operador de turbulencia y utilizan el valor de la función objetivo y las restricciones para determinar la calidad de la solución. Liang y Sugathan [32] utilizan un esquema de PSO multi-enjambres, cada subdivisión del enjambre total se encarga de una restricción que es asignada dinámicamente. Parsopoulos y Vrahatis [33] utilizan la variante de PSO que utiliza una versión mezclada de global best y local best con factor de estrechamiento utilizando parámetros que varían dinámicamente. Li, Tian y Kong [34] proponen un algoritmo de PSO que utiliza mutación como una alteración del vuelo de la partícula. Cagnina, Esquivel y Coello [35] estudian una variación de PSO que utiliza una versión total de global best y local best con valor de inercia. Lu y Chen [36] proponen un algoritmo que optimiza dos objetivos, primero se buscan soluciones factibles para luego optimizar la función objetivo. Krohling y dos Santos Coelho [37] plantean un enfoque basado co-evolutivo, dos sub-enjambres interactúan para optimizar la función objetivo y cumplir las restricciones. Finalmente He, Prempain y Wu [38] investigan la conservación de soluciones factibles, mediante un mecanismo de retroceso del vuelo de las partículas.

Muchas áreas de aplicación en ingeniería y ciencias afines han sido estudiadas mediante PSO, incluidas los sistemas de control, las telecomunicaciones, los sistemas de energía y el procesamiento de señales. Hasta el día de hoy, gran cantidad de publicaciones detallan las aplicaciones de los algoritmos de PSO. Para más información, véase [39]. Ejemplos de aplicaciones prácticas en ingeniería y en sistemas de energía pueden verse en [40], [41], [42] y [43]. Aunque PSO se ha utilizado principalmente para resolver problemas de optimización de una sola función objetivo sin restricciones, los algoritmos de PSO se han desarrollado para resolver problemas de optimización multi-objetivo con restricciones, problemas aplicados a sistemas cambiando dinámicamente, y para encontrar soluciones múltiples (véase Engelbrecht [29]).

Una serie de líneas de investigación son estudiadas actualmente en PSO, tales como la selección de parámetros en el desarrollo de algoritmos y la generación de nuevas variantes.

3. OPTIMIZACIÓN GLOBAL POR ANÁLISIS DE INTERVALOS

3.1 HISTORIA E IMPORTANCIA DEL ANÁLISIS DE INTERVALOS

En matemáticas básicas un problema se resuelve de forma exacta cuando se conoce la solución exacta. En pocas formas de enseñanza se acostumbra a determinar que un problema está resuelto con un resultado dentro de un intervalo de solución. En álgebra, para muchos problemas de un grado de dificultad alta, la solución exacta puede ser algo difícil de concebir y con un gasto de tiempo demasiado alto, para ello en algunas ocasiones, se tiene un buen nivel de eficiencia al encontrar un rango en el cual se pueda ubicar la respuesta del problema planteado. Es allí donde surge la aplicación de las técnicas de análisis de intervalos.

La idea de presentar cálculos como conjuntos de números y definir el álgebra sobre tales conjuntos sigue siendo relativamente nueva en algunos campos. Fue solo cuando los computadores se introdujeron que tales cálculos llegaron a ser prácticos y también de interés para propósitos de control de errores. Así, el análisis de intervalos surge como método para implementar mediante algoritmos computacionales.

La delimitación de errores de redondeo usando intervalos fue introducido en los años 50 [44] y [45]. Sin embargo el análisis de intervalos fue propuesto en la literatura por R. Moore en 1959 [46] y más formalmente en 1966 [47]. Desde entonces miles de artículos han aparecido y numerosos libros han sido publicados sobre el tema. Ejemplos de esto pueden verse en [48], [49], [50], [51] y [52]. Del planteamiento inicial de Moore se han derivado múltiples avances como los de Hansen en [53], [54], [55], [56] y [57]. Entre lo más destacado de sus trabajos presenta la lógica para realizar optimización global con aritmética de intervalos adaptando métodos que se utilizan en aritmética de punto flotante. Igualmente en [58] se muestran nuevos avances del método de análisis de intervalos, entre los cuales se encuentran: manejo de matrices con intervalos, método intervalos de Newton, integración de funciones de intervalos, ecuaciones diferenciales e integrales y aplicaciones e implementación en INTLAB (el cual es una herramienta computacional para análisis de intervalos desarrollado por el profesor Siegfried M. Rump de Hamburg University of Technology en Alemania)

El propósito del análisis de intervalos es proveer límites superiores e inferiores sobre el efecto que los errores computacionales producen en las cantidades computacionales. Es deseable hacer límites de intervalos tan angostos como sea posible. Un mejor enfoque sobre el análisis de intervalos es desarrollar algoritmos de intervalos que produzcan límites definidos (o cercanamente definidos) en la solución de problemas de computación numérica.

En la teoría del análisis de intervalos, se expresan sentencias de desigualdad en términos de intervalos cerrados sobre la recta real. Se piensa acerca de un intervalo como un conjunto de números, el cual comúnmente se representa como un par ordenado.

Para el análisis de intervalos, el estudio de la teoría e implementación de algoritmos para cálculos, puede verse más detallado en [59] y [60] .

Toda la posible aritmética real utilizada para punto flotante ahora es más detallada para análisis de intervalos, una muestra de ello es la aplicación de algoritmos para resolver un sistema de ecuaciones no lineales en una región inicial; métodos como el método de intervalos de Newton multi-variable, así como operadores derivados de Newton que muestran una mejor velocidad de convergencia son mencionados y explicados en [58] y [61]. Entre los más importantes operadores derivados de Newton se encuentran el operador de Krawczyk, Hansen-Sengupta y Hansen-Greenberg. Algunas modificaciones son realizadas para obtener cada operador, pero en general manejan el mismo concepto recursivo para converger a la solución del sistema de ecuaciones no-lineales.

Uno de los objetivos de este trabajo de grado es describir algoritmos que usen análisis de intervalos para resolver problemas de optimización no-lineales con restricciones. Los algoritmos que se tratan usan algebra lineal de intervalos y algoritmos de Newton para resolver el tipo de sistemas planteados. Por lo dicho anteriormente es necesario conocer los fundamentos y principios del análisis de intervalos para poder llevar a cabo la implementación requerida en el desarrollo de este documento.

En el ámbito de la optimización global el método de análisis de intervalos es poco utilizado por su difícil implementación, sin embargo la utilización de la aritmética de intervalos para la búsqueda recursiva de la solución de una función multi-variable sujeta a múltiples restricciones tiene una rápida velocidad de convergencia para condiciones establecidas (operadores de Newton).

La importancia del análisis de intervalos para la optimización global se debe a varios hechos como:

- El análisis de intervalo da fácilmente los límites necesarios en el rango de expresiones.
- El análisis de intervalos permite controlar no-linealidades de una forma sencilla.
- El análisis de intervalos extiende el análisis clásico en su capacidad de proporcionar existencia local y condiciones de optimalidad, válida dentro de una región local en torno a cierto punto, mientras que el análisis clásico en

general solo afirma la existencia de tales vecindarios sin proveer una forma sencilla de encontrarlos.

Referencias generales sobre métodos de intervalos para optimización global son incluidas en [62], [63], [64], [65].

3.2 FUNDAMENTOS Y PRINCIPIOS DEL ANÁLISIS DE INTERVALOS

La aritmética de intervalos es una generalización o una extensión de la aritmética real. Sea un intervalo real x definido como un conjunto no vacío de números reales

$$x = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} : \underline{x} \leq x \leq \bar{x}\}, \quad (7)$$

Donde \underline{x} es llamado el ínfimo y \bar{x} es llamado el supremo. La Tabla 1 ilustra algunos conceptos importantes en el desarrollo de la aritmética de intervalos. Así, el conjunto de todos los intervalos sobre \mathbb{R} es denotado por $I\mathbb{R}$, donde:

$$I\mathbb{R} = \{[\underline{x}, \bar{x}] : \underline{x}, \bar{x} \in \mathbb{R}, \underline{x} \leq \bar{x}\} \quad (8)$$

Tabla 1. Conceptos básicos en aritmética de intervalos

Aritmética real de intervalos	
Punto medio de x	$mid(x) = \check{x} = \frac{1}{2}(\underline{x} + \bar{x})$
Radio de x	$rad(x) = \frac{1}{2}(\bar{x} - \underline{x})$
Valor absoluto o la magnitud de x	$ x = mag(x) = \max\{ \check{x} : \check{x} \in x\}$, o $mag(x) = \max\{ \underline{x} , \bar{x} \}$
Mignitud de x	$mig(x) = \min\{ \tilde{x} : \tilde{x} \in x\}$ $mig(x) = \begin{cases} \min(\underline{x} , \bar{x}) & \text{si } 0 \notin x \\ 0 & \text{de otra forma} \end{cases}$
Punto intervalo o intervalo delgado	$[\underline{x}, \bar{x}] \equiv x$

Un intervalo \mathbf{x} es un subconjunto de un intervalo \mathbf{y} , denotado por $\mathbf{x} \subseteq \mathbf{y}$, si y solo si $\underline{y} \leq \underline{x}$ y $\overline{y} \geq \overline{x}$. La relación $\mathbf{x} < \mathbf{y}$ significa que $\overline{x} < \underline{y}$ y que otras desigualdades son definidas de forma similar.

[66]. Las operaciones en aritmética de intervalos son definidas en $\mathbb{I}\mathbb{R}$ tal que el intervalo resultante encierre todos los posibles resultados reales. Dado que sea $\mathbf{x} = [\underline{x}, \overline{x}]$ y $\mathbf{y} = [\underline{y}, \overline{y}]$, las operaciones elementales son definidas como:

$$\mathbf{x} \text{ op } \mathbf{y} = \{x \text{ op } y : x \in \mathbf{x}, y \in \mathbf{y}\} \text{ for } \text{op} \in \{+, -, \times, \div\} \quad (9)$$

La Tabla 2 ilustra las operaciones básicas en aritmética de intervalos. La aritmética extendida de intervalos puede revisarse en [58].

Tabla 2. Operaciones básicas en aritmética de intervalos

Operación matemática	Expresión
Suma	$\mathbf{x} + \mathbf{y} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$
Resta	$\mathbf{x} - \mathbf{y} = [\underline{x} - \overline{y}, \overline{x} - \underline{y}]$
Multiplicación	$\mathbf{x} \times \mathbf{y} = [\min\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\}, \max\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\}]$
División	$\frac{1}{\mathbf{y}} = \left[\frac{1}{\overline{y}}, \frac{1}{\underline{y}}\right]$ $\frac{\mathbf{x}}{\mathbf{y}} = \mathbf{x} \times (1/\mathbf{y})$ <p style="text-align: center;">Excluyendo la división por cero</p>
Potencia (enteros no negativos n)	$\mathbf{x}^n = \begin{cases} [1,1] & \text{si } n = 0 \\ [\underline{x}^n, \overline{x}^n] & \text{si } \underline{x} \geq 0 \text{ o si } n \text{ es impar} \\ [\overline{x}^n, \underline{x}^n] & \text{si } \overline{x} \leq 0 \text{ o si } n \text{ es par} \\ [0, \max(\underline{x}^n, \overline{x}^n)] & \text{si } \underline{x} \leq 0 \leq \overline{x} \text{ y } n > 0 \text{ es par} \end{cases}$

Una herramienta importante que surgió para el análisis de la aritmética de intervalos fue INTLAB [66]. Esta herramienta computacional permite operaciones básicas que se realizan en intervalos escalares reales y complejos, vectores y matrices. Estas operaciones se registran de forma similar a la aritmética real y compleja en MATLAB. Para una revisión general de los principios de la aritmética de intervalos en INTLAB véase [67]

Aparte de la evaluación de los intervalos y las formas centradas, también se emplean los métodos de intervalos de Newton para solucionar sistemas de ecuaciones lineales y no lineales y proveer convergencia rápida a las soluciones al reducir la cantidad de subregiones que se presentan en los métodos de particiones para optimización con restricciones. Dichos sistemas se plantean a continuación.

3.3 MÉTODO DE INTERVALOS DE NEWTON MULTIVARIABLE.

El método de intervalos de Newton fue derivado por Moore en 1966 y permite encontrar las raíces de una ecuación lineal o no lineal, de una o múltiples variables. En comparación a otros métodos numéricos este es muy veloz aunque no está exento de fallas. Tales como el caso donde el valor de partida o al que se llegue después de algunas sucesiones genere una derivada nula, es decir, la función en ese punto presente una tangente horizontal o el algoritmo explote al no estar definido en una derivada primera de valor cero.

Otro caso no deseable se genera cuando se presentan cambios bruscos y simétricos de convexidad y el algoritmo puede entrar en un círculo vicioso infinito, como ocurre con las funciones cúbicas donde el algoritmo no converge a la raíz sino que oscila permanente.

Para resolver sistemas de ecuaciones no-lineales tales como las presentadas en este documento, se utiliza el método de Newton mediante 3 operadores los cuales son modificaciones del operador básico planteado más adelante.

El problema es encontrar límites sobre la solución de una función continua no lineal $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ en una región dada $x^{(0)} \in \mathbb{I}\mathbb{R}^n$ debido al poder de la aritmética de intervalos para limitar rangos de funciones, la solución de sistemas de ecuaciones no-lineales y la optimización global, por métodos similares, son los de mayor uso del análisis de intervalos, como ya se nombró anteriormente.

Hay muchas aplicaciones donde nos gustaría resolver una ecuación lineal mediante una técnica que pueda encontrar, con certeza matemática, todas las raíces en una región determinada.

Usando un método de no-intervalos es difícil encontrar todas las soluciones y frecuentemente imposible revisar donde todas las soluciones han sido encontradas. Los métodos de intervalos pueden ser usados para contar el número de raíces en una región dada y proveer límites estrictos para cada raíz individual.

Usando el teorema de valor medio se tiene que para alguna x^*

$$f(x^*) \in f(\tilde{x}) + J(x)(x^* - \tilde{x}), \quad (10)$$

Donde $J(x)$ es la matriz Jacobiana de intervalos con

$$J_{ij} = \frac{d^2f}{dx_i dx_j} \quad i, j = 1, \dots, n, \quad (11)$$

la cual puede ser formada usando diferenciación automática y $\tilde{x} \in x$. Si x es un cero de f , $f(x^*) = 0$ y así $-f(\tilde{x}) \in J(x)(x^* - \tilde{x})$.

El sistema lineal propuesto anteriormente puede ser resuelto en x para obtener un límite externo sobre el ajuste de la solución, $N(\tilde{x}, x)$. La notación incluye tanto como para mostrar la dependencia sobre ambos términos. Esto da:

$$0 \in f(\tilde{x}) + J(x)(N(\tilde{x}, x) - \tilde{x}),$$

lo cual sugiere la iteración:

$$\begin{aligned} \text{resolver para } N \quad f(\tilde{x}^{(k)}) + J(x^{(k)})(N(\tilde{x}^{(k)}, x^{(k)}) - \tilde{x}^{(k)}) = 0, \\ x^{(k+1)} = x^{(k)} \cap N(\tilde{x}^{(k)}, x^{(k)}), \end{aligned} \quad (12)$$

Para $k=0, 1, \dots$ y $x^{(k)} \in x^{(k)}$. una elección razonable para $\tilde{x}^{(k)}$ es el centro $\tilde{x}^{(k)}$ de x , sin embargo mejores elecciones son disponibles; un ejemplo de determinar $\tilde{x}^{(k)}$ es usar la versión sin intervalos del método de Newton para encontrar una aproximación de la solución en x . Varios métodos de intervalos de Newton son determinados por como el operador $N(\tilde{x}^{(k)}, x^{(k)})$ es definido. Variaciones esenciales sobre los siguientes tres operadores son comúnmente utilizadas.

Operadores de Newton

Un sistema lineal puede ser resuelto usando métodos de Newton, empleando el operador:

$$N(\tilde{x}, x) = \tilde{x} - J(x)^{-1}f(\tilde{x}). \quad (13)$$

En la práctica este operador no se usa ya que un sistema de intervalos lineales tiene que ser resuelto para cada iteración. También $J(x)$ es comúnmente dado a contener matrices singulares, a menos que el ancho de x sea pequeño.

Una alternativa es seguir Krawczyk y usar lo que ahora se llama el operador de Krawczyk.

La ventaja de este operador es que ecuaciones lineales sin intervalos han sido resueltas en alguna etapa, incrementando así la velocidad y confiabilidad. Sustituyendo los términos de la ecuación lineal en el operador de Krawczyk lleva a su versión para sistemas no lineales $K: \in \mathbb{IR}^n \times \mathbb{R}^n \rightarrow \mathbb{IR}^n$ definida por:

$$K(\tilde{x}, x) = \tilde{x} - Cf(\tilde{x}) - (I - CJ(x))(\tilde{x} - x), \quad (14)$$

donde C es la matriz de preconditionamiento o inversa del punto medio de $J(x)$; $\tilde{x} \in x$. Aunque una inversa es aun requerida, es la inversa de una matriz real más que de una matriz de intervalos.

Reemplazando $N(\tilde{x}, x)$ por $K(\tilde{x}, x)$ en la ecuación se obtiene la iteración de Krawczyk:

$$\mathbf{x}^{(k+1)} = K(\tilde{x}^{(k)}, \mathbf{x}^{(k)}) \cap \mathbf{x}^{(k)}, \quad (15)$$

Para $k=0, 1, \dots$ y $\tilde{x}^{(k)} \in \mathbf{x}^{(k)}$.

Un tercer operador desarrollado por Hansen y Sengupta, usa un procedimiento de Gauss-Seidel. Precondicionando (15) con la inversa del punto medio C de $J(x)$ da:

$$CJ(x)(N(\tilde{x}, x) - \tilde{x}) = -Cf(\tilde{x}) \quad (16)$$

Cambiando la notación $N(\tilde{x}, x)$ a $H(\tilde{x}, x)$ y definiendo:

$$M = CJ(x), \quad b = Cf(\tilde{x}), \quad (17)$$

El procedimiento de intervalos de Gauss-Seidel procede componente por componente a dar la iteración:

$$H(\tilde{x}^{(k)}, \mathbf{x}^{(k)})_i = \tilde{x}_i^{(k)} - \frac{b_i + \sum_{j=1}^{i-1} M_{ij}(x_j^{(k+1)} - \tilde{x}_j^{(k+1)}) + \sum_{j=i+1}^n M_{ij}(x_j^{(k)} - \tilde{x}_j^{(k)})}{M_{ii}} \quad (18)$$

$$\mathbf{x}_i^{(k+1)} = H(\tilde{x}^{(k)}, \mathbf{x}^{(k)})_i \cap \mathbf{x}_i^{(k)},$$

Para $k=0, 1, \dots$ y $\tilde{x}^{(k)} \in \mathbf{x}^{(k)}$

Un beneficio de usar el análisis de intervalos para resolver ecuaciones no-lineales, es que los operadores de Hansen-Sengupta pueden ser usados para probar la existencia y unicidad de un cero en el intervalo x [55].

Como parte de la resolución del método de análisis de intervalos para optimización de funciones no-lineales con restricciones, se utilizó también el método de Newton propuesto por Hansen-Greenberg en [61], para proporcionar menor tiempo de computo en algunas funciones, este método soluciona este tipo de sistemas mediante el empleo de tres subalgoritmos. El primero es un paso tipo Gauss-Seidel. El segundo es una iteración de Newton real, sin intervalos y el tercero resuelve la ecuación linealizada por el método de eliminación

4. DESARROLLO DE ALGORITMOS PARA PSO Y ANALISIS DE INTERVALOS.

4.1 DESARROLLO DEL ALGORITMO PSO

En el desarrollo del trabajo de grado presentado en este documento se utilizó como herramienta básica de simulación el software MATLAB en la versión 7.1 bajo la cual posee licencia la Universidad Industrial de Santander. Mediante este software los métodos de optimización fueron plasmados y evaluados en las diversas funciones objetivo propuestas.

Dos métodos con sus respectivos algoritmos fueron necesarios para la utilización y evaluación de las funciones objetivo en la optimización no-lineal con restricciones. El primero de ellos fue el método de optimización por enjambre de partículas. Este método pudo ser simulado mediante la implementación del algoritmo de PSO al cual se le realizaron algunas modificaciones para poder llevar a cabo la respectiva comparación. El algoritmo propuesto posee dos herramientas dentro de su funcionamiento, la primera es el algoritmo básico de PSO para funciones no-lineales sujetas a restricciones, el segundo es la utilización de la función *fmincon* la cual es una herramienta propia de MATLAB para llevar a cabo la solución de problemas de optimización con restricciones y esta ajena al contenido de esta tesis ya que no proporciona las características ni los principios bajo los cuales está caracterizado PSO. Dicho esto solo se utilizó de este algoritmo la herramienta que empleaba y evaluaba la optimización por enjambre de partículas (PSO).

Descripción del algoritmo general PSO con restricciones.

El algoritmo general de PSO con restricciones empleado en este trabajo de grado, se pudo entender mucho mejor estudiando el algoritmo de PSO básico sin restricciones.

Para empezar, la posición de una partícula i se expresa por x_i , donde x_i es un vector que almacena cada una de las posiciones que tiene la partícula en cada una de las dimensiones que comprende el espacio de búsqueda. Además, se expresa como v_i a la velocidad de la partícula i , la cual también es un vector que contiene cada una de las velocidades que tiene la partícula en cada dimensión. Esta velocidad es añadida a la posición de la partícula, para mover la partícula desde un tiempo $t - 1$ a un tiempo t .

La descripción del algoritmo PSO empezó estudiando cómo está determinada cada partícula [68]. Una partícula está compuesta de tres vectores y dos valores de aptitud (también conocida como bondad, adaptación, capacidad, adecuación o "fitness") con respecto al problema considerado:

1) Tres vectores:

- El vector $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,N})$ almacena la posición actual de la partícula.
- El vector $pBest_i = (pBest_{i,1}, pBest_{i,2}, \dots, pBest_{i,N})$ almacena la posición de la mejor solución encontrada por la partícula hasta el momento.
- El vector de velocidad $v_i = (v_{i,1}, v_{i,2}, \dots, v_{i,N})$ almacena la dirección según la cual se moverá la partícula.

2) Dos valores de aptitud:

- El valor de $aptitud_xi$, almacena el valor de adaptación o adecuación de la posición actual correspondiente al vector x_i .
- El valor de $aptitud_mejorposi$, almacena el valor de adecuación de la partícula con mejor solución local encontrada hasta el momento, correspondiente al vector $mejorposi$.

La descripción del proceso algorítmico es la siguiente:

- 1) La nube se inicializa generando las posiciones de forma aleatoria.
- 2) Se generan las velocidades aleatoriamente en un intervalo establecido $[-Vmax, Vmax]$, no es conveniente fijarlas a cero.
- 3) Se calcula la aptitud de cada partícula y se actualizan los valores de $aptitud_xi$ y $aptitud_mejorposi$.
- 4) Las partículas se mueven en cada iteración desde una posición del espacio de búsqueda hasta otra. Al vector de posición x_i se le añade el vector velocidad v_i para obtener un nuevo vector x_i .
- 5) Con la nueva posición de la partícula se calcula y actualiza $aptitud_xi$.
- 6) Si el nuevo valor de aptitud es el mejor encontrado por la partícula i hasta el momento, se actualizan los valores de $pBest_i$ y $aptitud_mejorposi$.
- 7) Si el nuevo valor de $aptitud_mejorposi$ es el mejor encontrado por la nube de partículas hasta el momento, se actualizan el valor de la mejor posición de la nube $pBest_i$ y su $aptitud_mejorposi$.
- 8) El vector velocidad de cada partícula es modificado en cada iteración utilizando la velocidad anterior, un componente cognitivo y un componente social. El modelo matemático resultante que representa la esencia del algoritmo PSO, viene representado por las siguientes ecuaciones:

$$v(t+1)_i = w^t * v(t)_i + c1 * rand() * (pBest_i - x(t)_i) + c2 * Rand() * (pG - x(t)_i) \quad (19)$$

$$x(t+1)_i = x(t)_i + v(t+1)_i. \quad (20)$$

Para $i = 1, 2, \dots, P$

dónde:

$x(t + 1)_i$ \equiv Vector posición de la partícula i en la iteración t .

$v(t + 1)_i$ \equiv Vector velocidad de la partícula i en la iteración t .

w^t \equiv Factor de inercia en la iteración t .

φ_1, φ_2 \equiv Son pesos que controlan los componentes cognitivo y social.

rand_1 \equiv Número aleatorio entre 0 y 1.

rand_2 \equiv Número aleatorio entre 0 y 1.

$pBest_i$ \equiv Mejor posición encontrada por la partícula i hasta el momento que posee la mejor solución.

pG . \equiv Representa la posición de la partícula con la mejor solución o aptitud.

P \equiv número de partículas que componen la nube.

La ecuación (19) actualiza el vector velocidad de cada partícula i en la iteración t . Igualmente, la ecuación (20) actualiza el vector de posición de la partícula i para cada iteración.

De las formulas planteadas en (19) para el algoritmo de optimización por enjambre de partículas, se puede observar que la velocidad es una función que está compuesta por la suma de tres partes. La primera parte es la velocidad anterior de la partícula. A esta parte se le conoce como inercia. La segunda parte es la diferencia entre la mejor posición encontrada por la partícula, con la actual posición. Esta es la parte cognitiva que representa el aprendizaje de su propia experiencia. Y la última parte es la diferencia entre la mejor posición alcanzada por alguna partícula en la totalidad del grupo (global best), con la posición actual de la partícula. Esta parte es la social, que representa el aprendizaje del grupo [68].

El componente cognitivo indica la decisión que tomará la partícula y depende de su propia experiencia, dicho de otra manera, representa la distancia entre la posición actual y la mejor conocida por esa partícula. El componente cognitivo en la ecuación (19) es el factor: $c_1 * \text{rand}() * (pBest - x)$.

El componente social apunta la decisión que tomará la partícula en base a la influencia del resto de partículas que componen la nube, es decir, representa la distancia entre la posición actual y la mejor posición encontrada por vecindario. El componente social se modela en la ecuación (19) como el factor: $c_2 * \text{Rand}() * (pG - x)$ [68].

El algoritmo mostrado anteriormente es un indicio básico de cómo se trabajó la programación para PSO en este trabajo de grado. Para el manejo de restricciones se utilizó un mecanismo simple encontrado en la literatura especializada basado en reglas de factibilidad. Como ya se dijo, para la realización de este documento

se usó una implementación del algoritmo de optimización por enjambre de partículas, este tiene la misma sintaxis que la herramienta computacional de algoritmos genéticos empleada en MATLAB, con algunas opciones adicionales específicas a PSO. Así, permite la reutilización del código cuando se trata de diferentes algoritmos de optimización basados en la población. Este código fue utilizado para nuestro propósito con algunas variaciones para demostrar la eficiencia del método PSO en algunas condiciones.

4.2 DESARROLLO DEL ALGORITMO DE OPTIMIZACIÓN POR ANÁLISIS DE INTERVALOS

El método de análisis de intervalos utiliza las herramientas básicas de MATLAB más una herramienta computacional denominada INTLAB, esta aplicación de MATLAB permite trabajar con diversas operaciones y funciones aplicadas a intervalos específicamente, en lugar de puntos como suelen ser los cálculos comunes, es decir, se introducen intervalos para su evaluación directamente. La introducción del software rápido y eficiente para aritmética de intervalos INTLAB ha dado como resultado un incremento en la popularidad del análisis de intervalos. En [67] se da una introducción a la aritmética de intervalos y se explica cómo se implementa la herramienta computacional INTLAB. Un método de análisis de intervalos fue descrito para encontrar el máximo global de una función de hasta 3 variables multimodal sujeta a restricciones de igualdad y/o desigualdad. Al descartar subregiones donde la solución global puede no existir mediante la evaluación de unas pruebas de comparación y aplicando de manera opcional el método de intervalos de Newton para resolver la ecuación de Lagrange, se puede encontrar la solución con un límite de error riguroso. Para ello se realiza el respectivo estudio del método con las funciones objetivos propuestas inicialmente en esta tesis. Este método propuesto por Ichida en [69] es una técnica poco conocida en la literatura especializada.

El algoritmo implementado para la optimización de funciones no-lineales con restricciones se compone de dos partes: la primera de ellas utiliza los principios del método de bisección para descartar regiones, realizar la comparación de las funciones y encontrar los puntos óptimos globales. La segunda parte como ya se dijo antes, utiliza el método de intervalos de Newton para reducir la cantidad de subregiones y garantizar una convergencia rápida (en el caso tratado en este documento se hace énfasis en la primera parte del algoritmo). En este trabajo de grado se utilizó una modificación propuesta por Hansen-Greenberg que utiliza las mejores ventajas de cada operador de Newton.

El algoritmo general propuesto en [69] contiene los siguientes pasos (ver Figura 7):

Paso 1: Sea $A = [x_{11}, x_{12}] \times [x_{21}, x_{22}] \times \dots \times [x_{n1}, x_{n2}]$ el dominio original de las variables que pueda ser suficientemente grande para contener la solución.

Paso 2: Se divide el dominio original en dos subregiones en el punto medio de su lado más grande. Se Examina si las subregiones divididas pueden ser descartadas o no por las pruebas 1 y prueba 2.

Prueba 1. Comparación entre valores de la función objetivo. Sean $F(A_{izq}) = [\underline{f_{izq}}, \overline{f_{izq}}]$ y $F(A_{der}) = [\underline{f_{der}}, \overline{f_{der}}]$ valores de intervalo de $f(x)$ en dos subregiones A_{izq} y A_{der} que no violen las restricciones, respectivamente. Si $\overline{f_{izq}} < \underline{f_{der}}$, entonces se puede descartar A_{izq} ya que es evidente que el punto máximo global no puede existir en A_{izq} (fig.5). El caso en que $\overline{f_{der}} < \underline{f_{izq}}$ se trata de manera similar. Si se puede calcular $f_m = f(x_m)$, donde $x_m \in A_L$ es un punto que satisface las restricciones, el siguiente mecanismo es útil para descartar subregiones innecesarias. Si $\overline{f_{der}} < f_m$, se puede descartar A_{der} y el punto máximo global no se pierde (fig. 6).

Figura 5. Valores de intervalos de $F(X)$ en 2 subregiones

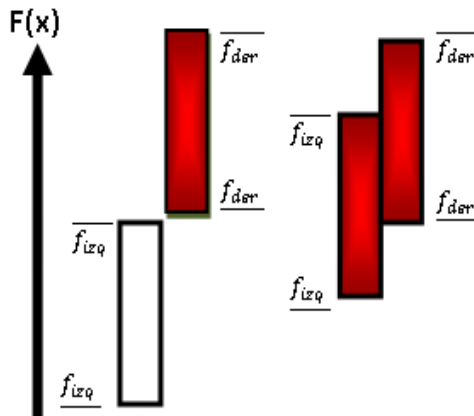
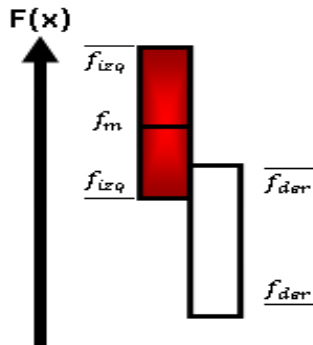


Figura 6. Comparación entre f_m y $\overline{f_{der}}$



Prueba 2: Control sobre restricciones. Considere una restricción de igualdad $g(x) = 0$. Sea $G(X) = 0$ la extensión del intervalo. Para una subregión A, si:

$$(17) \quad G(A) = [\underline{g}, \bar{g}] \not\subseteq 0$$

es decir, si $\underline{g} > 0$ o $\bar{g} < 0$, entonces A puede ser descartada ya que no satisface la restricción de igualdad.

Para una restricción de desigualdad $h(x) \leq 0$ se requiere que:

$$(18) \quad H(A) = [\underline{h}, \bar{h}] > 0$$

Si este es el caso ($\underline{h} > 0$), la región A puede ser descartada.

Al revisar detalladamente el artículo propuesto por Ichida en [69] se pudo concluir que para la realización de las pruebas 1 y 2 en el algoritmo era necesario primero realizar la evaluación de la prueba 2 y luego la realización de la prueba 1 ya que una condición necesaria en la prueba 1 es que las regiones evaluadas no violen las restricciones y esta condición es precisamente el principio de funcionamiento de la prueba 2, así que para la creación del algoritmo que realizara estas pruebas se siguió el orden mencionado.

Paso 3: Si el ancho máximo de las subregiones reservadas se encuentra dentro de la tolerancia, el algoritmo encontró la respuesta. De lo contrario se elige una subregión entre las subregiones reservadas y se calcula U_i dada en las ecuaciones de Lagrange para igualdad o desigualdad. En caso de que L no sea diferenciable y que la U_i no se pueda calcular, se va al paso 2.

Paso 4: Si $0 \notin U_i$ para algún valor i , se va al paso 2 ya que las ecuaciones de Lagrange para igualdad o desigualdad no tienen soluciones reales. De lo contrario se calcula $D = \det \left\{ \frac{dU_i}{dx_j} \right\}$.

Paso 5: Si $0 \in D$, vaya al paso 2, ya que el Jacobiano de la matriz puede no tener inversa. De lo contrario se aplica el método de intervalos de Newton y se guarda el resultado. Se recurre nuevamente al paso 2.

La primera parte del algoritmo (Pasos 1, 2 y 3), la cual descarta regiones mediante la comparación de la evaluación de las funciones y las restricciones, se

implementó mediante un desarrollo propio de los autores de este documento utilizando el lenguaje de programación MATLAB y la herramienta computacional INTLAB [66]. La realización de esta comparación se implementa mediante las pruebas 1 y 2. Mediante una interfaz gráfica se pudo corroborar el cumplimiento de las pruebas (ver anexo 2).

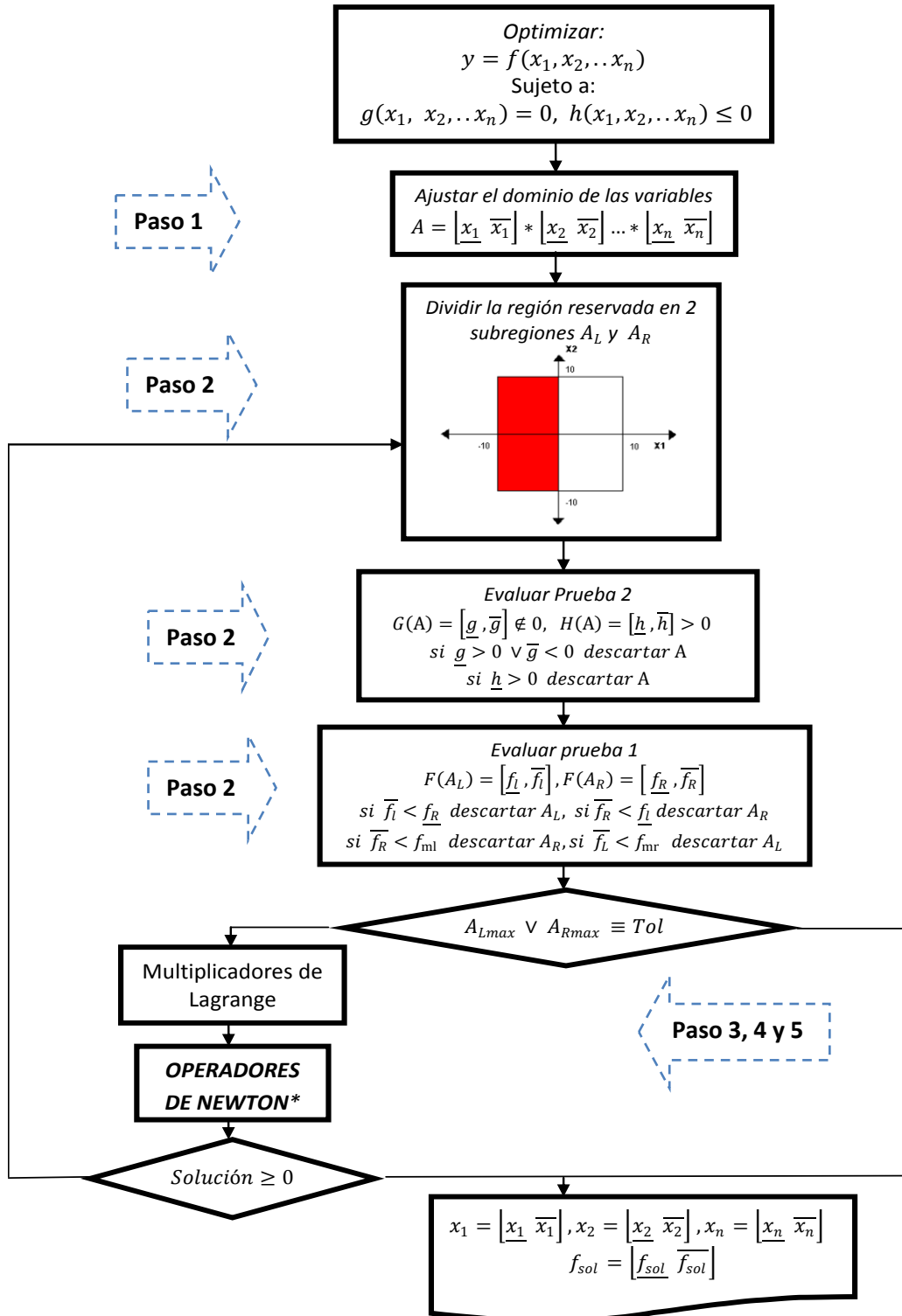
Por otra parte el algoritmo de Hansen-Greenberg se puede resumir mediante los siguientes pasos [61]:

1. Se calcula $J(X)$ (Jacobiano de f).
2. Se calcula B , la cual es aproximadamente la inversa de J .
3. Se calcula $M = BJ(x)$.
4. Se ajusta x al valor central de X .
5. Se realiza un paso de Hansen-Sengupta para todos los i tales que $0 \notin M_{ii}$.
6. Se realiza un paso de Hansen-Sengupta para todos los i tales que $0 \in M_{ii}$ y se guarda el intervalo más grande.
7. Se realiza la iteración real para mejorar x , partiendo del centro de X .
8. Si $\|f(x)\|$ no es lo suficientemente pequeño, se recurre al paso 14.
9. Se realiza la descomposición LU de M si es posible, de otra forma se va al paso 14.
10. Se calcula $Bf(x)$ y se resuelve Z por sustitución hacia atrás y adelante.
11. Se reemplaza X por $X \cap Z$.
12. Si el ancho de X se mejora significativamente, se ajusta x al nuevo valor central de X , y se regresa al paso 10.
13. Se recurre al paso 16.
14. Se realiza un paso de Hansen-Sengupta para los componentes tales que $0 \notin M_{ii}$.
15. Si X se mejora significativamente, se ajusta x igual al centro de X , y se regresa al paso 14.
16. Si se tiene un intervalo almacenado del paso 6, se usa. Si la región se divide, se almacena una subregión y la otra parte se mantiene como el nuevo X . Se regresa al paso 1.
17. Si la región no mejoró significativamente en los pasos 5 y 6, en el paso 11 o en el paso 14, entonces se divide esta región por la mitad del componente más largo, se almacena una subregión y la otra se usa como el nuevo X .
18. Se regresa al paso 1.

En los pasos 5 y 6, 11 o 14, la región es reducida al valor del tamaño de la tolerancia. Si no se encuentra una solución, entonces la región almacenada más nueva, se convierte en el nuevo X y se regresa al paso 1. Una región suficientemente pequeña puede ser sacada y retirada internamente. Cuando no hay regiones almacenadas, se ha terminado el proceso.

Para más detalles e información importante sobre el método de Hansen-Greenberg véase [61]. A continuación en la Figura 7 se puede observar el funcionamiento general del algoritmo de optimización por análisis de intervalos.

Figura 7. Algoritmo de optimización con restricciones por análisis de intervalos



5. ESTUDIO DEL COMPORTAMIENTO DE PSO CON RESTRICCIONES

Para la realización de los experimentos con PSO se utilizaron 9 funciones de prueba establecidas previamente, con restricciones asignadas a cada función donde se quiso determinar el comportamiento del algoritmo estudiado al variar algunos parámetros básicos.

5.1 FUNCIONES DE PRUEBA

Se seleccionaron 9 funciones de prueba o funciones objetivo las cuales son las que se pretendían optimizar, estas funciones son no-lineales y mediante su estudio se buscó verificar el comportamiento de PSO. Estas funciones objetivo también fueron utilizadas para el estudio comparativo entre PSO y el análisis de intervalos. En la Tabla 3 se ilustran estas funciones junto con las restricciones y los límites propuestos.

Tabla 3. Funciones objetivo y restricciones

	Función objetivo	Restricciones	Límites de restricciones
F1	$f_1 = -(x_1 + x_2 - x_3 - 1)^2 - (x_1 + x_2)^2 - 5x_1^2$	$2x_1 + x_3 = 0$	$-10 \leq x_i \leq 10$
F2	$f_2 = -2x_1^2 + 2x_1x_2 - 2x_2^2 + 6x_1$	$3x_1 + 4x_2 - 6 \leq 0$	$-10 \leq x_i \leq 10$
F3	$f_3 = -x_1 - x_2 - x_2^2$	$x_2^2 - x_1^3 = 0$	$-10 \leq x_i \leq 10$
F4	$f_4 = -x_1 - x_2$	$x_2 - x_1^3 \leq 0$ $-x_2 \leq 0$	$-10 \leq x_i \leq 10$
F5	$f_5 = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$ (Rosenbrock's Valley)	$3x_1 + 4x_2 - 6 \leq 0$	$-10 \leq x_i \leq 10$
F6	$f_6 = (1 + (x_1 + x_2 + 1)^2 * (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)) * (30 + (2x_1 - 3x_2)^2 * (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2))$ (Goldstein and Price)	Quadrifolium $(x_1^2 + x_2^2)^3 - 8x_1^2x_2^2 \leq 0$	$-2 \leq x_i \leq 2$

Tabla 3. (Continuación)

F7	$f_7 = \sum_{i=1}^N (x_i^2 - 10 \cos(2\pi x_i) + 10)$ <p>(Rastrigin)</p>	<p>Heart</p> $(x_1^2 + x_2^2 - 1)^3 - x_1^2 x_2^3 \leq 0$	$-5.12 \leq x_i \leq 5.12$
F8	$f_8 = \sum_{i=1}^N (0.25x_i^3 + 0.75x_i^2 - 1.5x_i - 2)$	<p>Quadrifolium</p> $(x_1^2 + x_2^2)^3 - 8x_1^2 x_2^2 \leq 0$	$-2 \leq x_i \leq 6$
F9	$f_9 = \sum_{i=1}^N x_i^2$ <p>(Jong)</p>	$x_1 + x_3 = 0$ $x_1 - 1 \leq 0$	$-5.12 \leq x_i \leq 5.12$

Para todo el desarrollo del presente documento los valores óptimos calculados para las funciones F1, F2, F3 y F4 fueron valores máximos globales de las funciones sujetos a restricciones. En el caso de F5, F6, F7, F8 y F9 los valores óptimos globales calculados fueron valores mínimos de las funciones sujetos a restricciones.

- **GRÁFICAS DE LAS FUNCIONES OBJETIVO CON RESTRICCIONES**

A continuación se muestra una representación gráfica de cada una de las funciones objetivo propuestas y las restricciones que se le asignaron a cada una.

Figura 8. Función objetivo F2 con la restricción respectiva de la tabla 3.

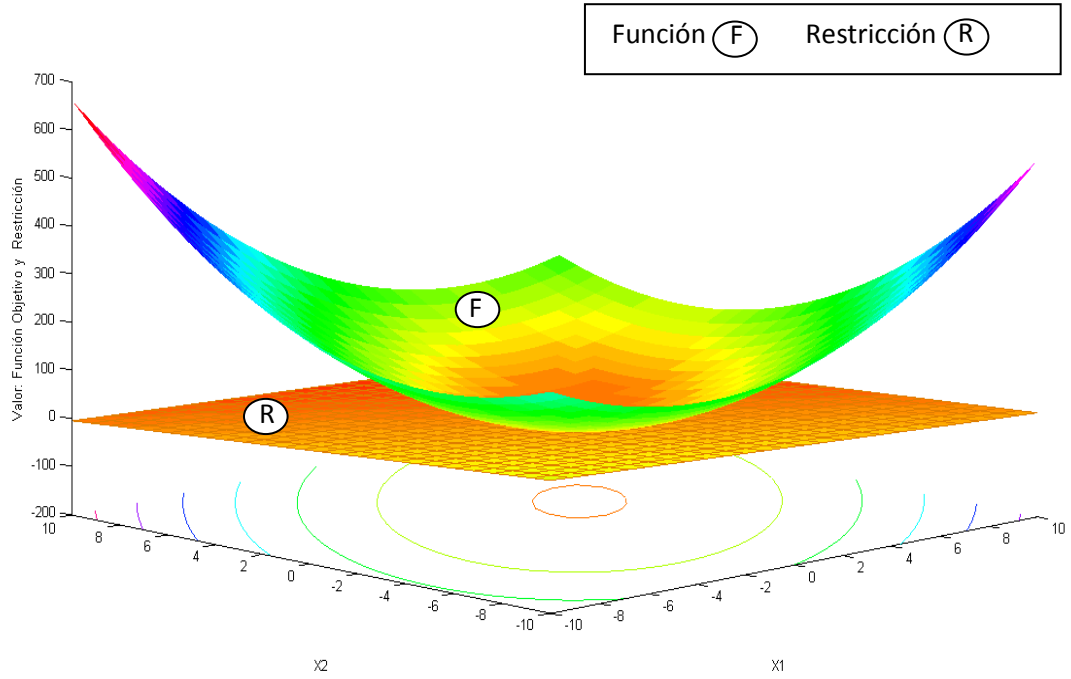


Figura 9. Función objetivo F3 con la restricción respectiva de la tabla 3.

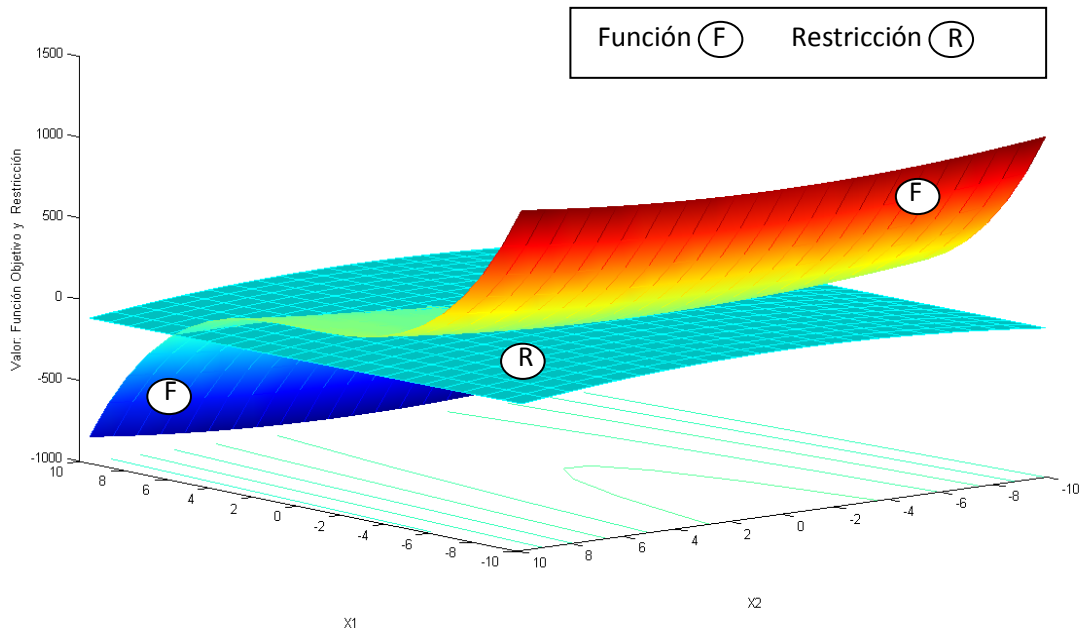


Figura 10. Función objetivo F4 con la restricción respectiva de la tabla 3.

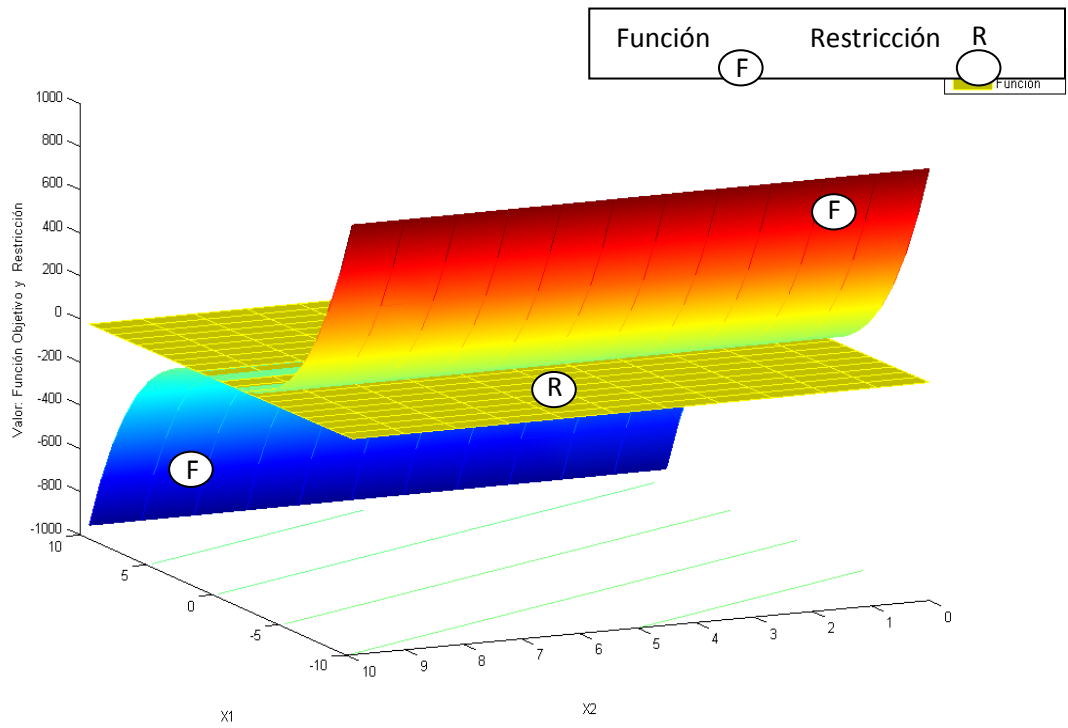


Figura 11. Función objetivo F5 con la restricción respectiva de la tabla 3.

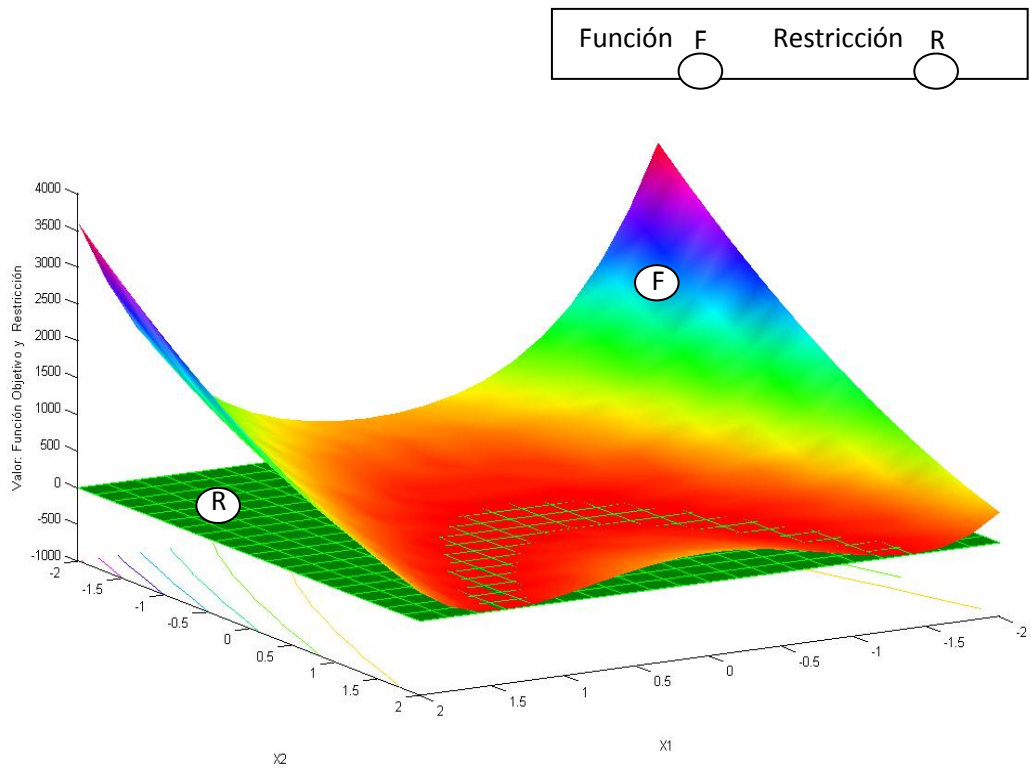


Figura 12. Función objetivo F6 (Goldstein & Price)

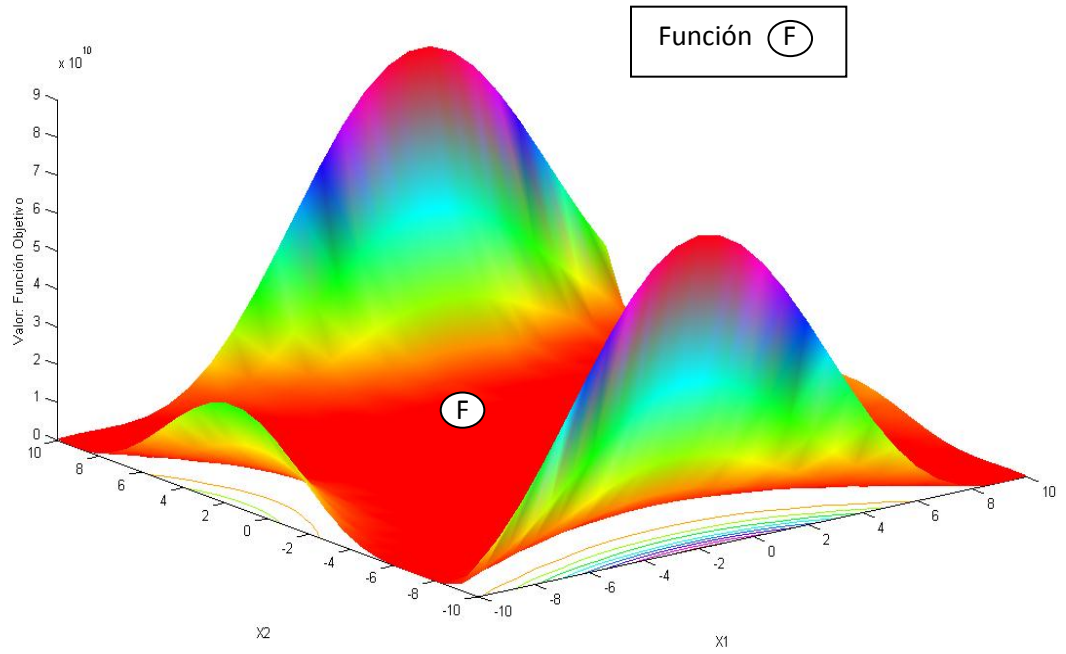


Figura 13. Función objetivo F7

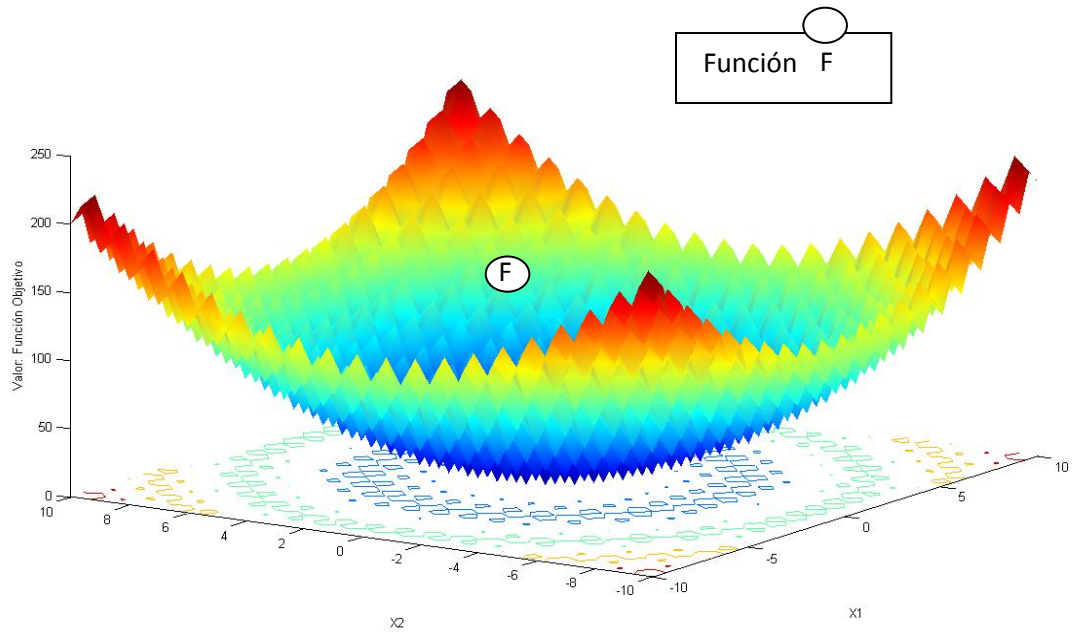
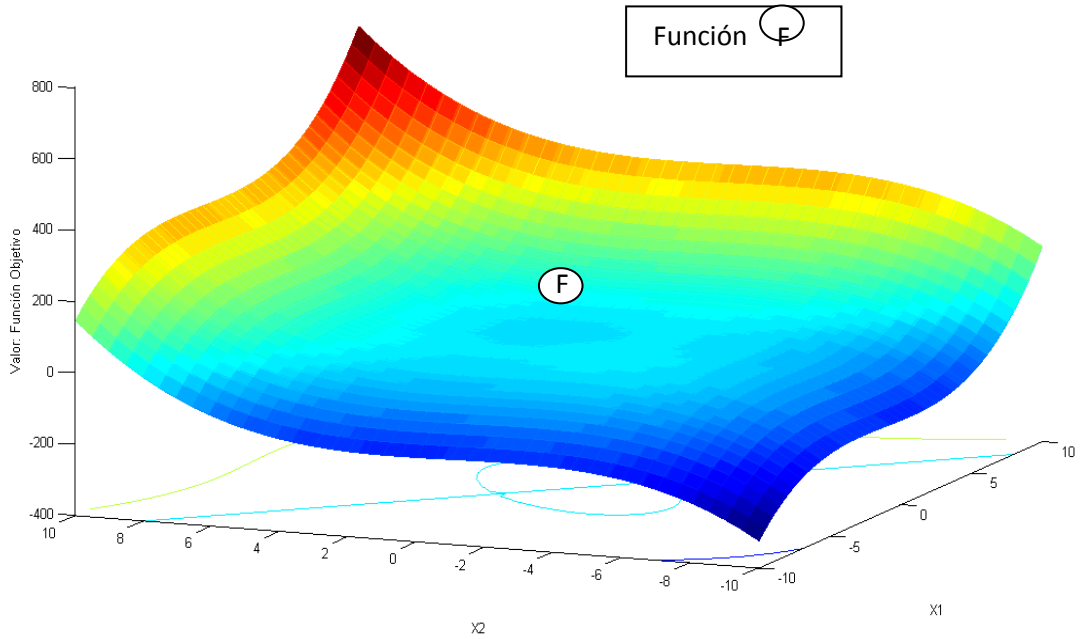


Figura 14. Función objetivo F8



5.2 EXPERIMENTO 1 PSO: VARIACION CANTIDAD DE PARTICULAS.

El tamaño de la cantidad de partículas juega un papel importante, ya que determina el equilibrio entre la calidad de las soluciones obtenidas y el número de iteraciones necesarias hasta llegar a una buena solución (tiempo computacional). La Tabla 4 muestra un estudio comparativo de las funciones de prueba al variar la cantidad de partículas. Para el caso de todas las funciones se calcularon mínimos globales.

Tabla 4. Variación de la cantidad de partículas para las funciones de prueba y resultados obtenidos con 200 iteraciones máximas.

		Cantidad de partículas N (población)						
Función objetivo		$N=50$	$N=100$	$N=150$	$N=200$	$N=250$	$N=300$	$N=400$
F1	Mejor	-7.5515870	-2.8511744	-0.39033	-0.36887	-0.00364	-0.35863	-0.35850
	Media	-14.323386	-5.5191062	-0.77836	-0.00365	-0.56874	-0.35747	-0.00358
	Desv. estándar	1.19577099	1.08165092	0.609929	0.605195	0.564738	0.568049	0.568185

Tabla 4. (Continuación)

F2	Mejor Media Desv. estándar % partículas acertadas	5.3509 5.3509 5.2264e- 006 76	5.3514 5.3513 4.0476e- 005 59	5.3514 5.3513 2.3102e- 004 57	5.3514 5.3513 1.0846e- 004 57	5.3514 5.3513 2.3557e- 005 72	5.3514 5.3513 1.8350e- 004 51	5.3514 5.3513 1.3120e- 004 51
F3	Mejor Media Desv. estándar	0.21900499 0.83358386 0.53247932	0.18082176 0.24577602 0.53247932	0.163875 0.236507 0.532479	0.161022 0.243520 0.532479	0.167237 0.242479 0.248434	0.169720 0.222069 0.532479	0.160218 0.243694 0.532479
F4	Mejor Media Desviación estándar % partículas acertadas	0.0126 0.0126 1.0900e- 006 72	0.0126 0.0110 0.0104 38	0.0126 0.0115 0.0060 41	0.0126 -0.0112 0.0058 37	0.0126 0.0126 1.6957e- 004 30	0.0126 0.0118 0.0042 34	0.0126 0.0108 0.0091 33
F5	Mejor Media Desv. estándar	0.0084 0.0172 0.0013	0.0088 0.0177 0.0023	0.0088 0.0176 0.0021	0.0088 0.0176 0.0021	0.0088 0.0176 0.0028	0.0086 0.0174 0.0014	0.0087 0.0176 0.0017
F6	Mejor Media Desv. estándar % partículas acertadas	26.0388 26.0388 3.2128e- 005 78	20.5060 20.5060 0.0027 56	26.0388 26.0388 4.9315e- 005 60	20.5113 20.5113 0.0573 55	20.5059 20.5059 7.0655e- 004 51	20.5056 20.5069 0.0133 51	20.5056 20.5073 0.0216 54
F7	Mejor Media Desviación estándar	0 -2.84e-016 9.736e-016	0 -1.26e-010 4.730e-010	0 -0.1260 1.5431	0 -2.6e-008 3.7e-007	0 -0.1220 1.9292	0 -0.0482 0.8341	0 4.8e-014 9.4e-013
F8	Mejor Media Desv. estándar	5.0270287 4.90919633 0.04399603	5.02895070 4.96684232 0.03826004	5.023604 4.936204 0.041308	5.029130 4.917614 0.042633	5.028922 4.880217 0.044401	5.029125 4.885339 0.046369	5.028713 4.927028 0.042221

En el cálculo de los valores de la tabla anterior, en el algoritmo en MATLAB se utilizó la función *soft* [70] para limitar los valores óptimos con restricciones de desigualdad. Este tipo de limitación permite a las partículas dejar los límites del

problema, pero ajusta su valor óptimo a infinito cuando ellas lo hacen. Este tipo de limitación es indicada cuando se utilizan restricciones de desigualdad. En el caso de la tabla quedó registrada la cantidad de partículas acertadas, es decir aquellas que no dejaron los límites del problema y se quedaron en los rangos adecuados. En el caso de restricciones de igualdad se utilizó en MATLAB la función *penalize* [70] que brinda la mejor combinación entre buena presentación y versatilidad. Otro tipo de opciones de limitación son *absorb* y *reflect* [70]. Estas opciones permitan a las partículas viajar por fuera de los límites del problema pero en algunos casos pueden sufrir de pobre desempeño si se utilizan restricciones de igualdad no lineales.

Resultados experimento 1. El tamaño de la población es importante en PSO, para poblaciones pequeñas, el algoritmo tiene un comportamiento pobre, pero a medida que se aumenta el tamaño, se va mejorando el comportamiento de la técnica.

Para PSO, una población grande permite acercarse más al valor óptimo, pero la diferencia no es sustancial comparado a una población de tamaño medio.

El algoritmo no se comporta bien para valores pequeños de población. Valores pequeños proveen muestras insuficientes, lo que hace que el algoritmo actúe de forma pobre. Al contrario, grandes tamaños de población, mejorarán el comportamiento del algoritmo, debido a una búsqueda informada y dirigida.

Sin embargo, una población grande, requiere más evaluaciones por iteración, lo que podría demorar encontrar el valor óptimo y no representa ganancia comparado con un valor de población de término medio por lo cual es más recomendable utilizar este tipo de tamaño de población. Se deduce que una población de tamaño entre 200 y 250 partículas serían los valores con los cuales se obtendría el mejor desempeño del algoritmo, así se llegaría a una respuesta óptima a un costo computacional bajo.

5.3 EXPERIMENTO 2 PSO: VARIACIÓN CANTIDAD DE ITERACIONES.

Tabla 5. Variación de la cantidad de iteraciones para las funciones de prueba y resultados obtenidos una población de N=200 partículas.

		Numero de iteraciones máximas						
Función objetivo		Iter=50	Iter =100	Iter =150	Iter =200	Iter =300	Iter =500	Iter =1000
F1	Mejor	-0.6771784	-0.453189	-0.39795	-0.36675	-0.35799	-0.35840	-0.359111
	Media	-0.6639325	-0.450200	-0.39824	-0.36369	-0.35741	-0.35545	-0.003663
	Desv. estándar	0.59854308	0.5772759	0.568125	0.571653	0.569038	0.567847	0.5689322

Tabla 5. (Continuación)

F2	Mejor Media Desv. estándar	5.3213 5.3512 1.9344e-004	5.3513 5.3513 5.0687e-005	5.3514 5.3513 9.5085e-005	5.3514 5.3513 1.9158e-005	5.3514 5.3513 3.5934e-004	5.3514 5.3513 1.4374e-004	5.3514 5.3513 4.5815e-005
F3	Mejor Media Desvia estándar	0.17486337 0.23233542 0.50936108	0.1617715 0.2390977 0.5086882	0.164798 0.246647 0.507936	0.153038 0.238990 0.508698	0.158818 0.252951 0.507309	0.135222 0.242466 0.508352	0.138452 0.243567 0.504598
F4	Mejor Media Desv. estándar	0.0059 -0.0525 0.2730	0.0126 0.0109 0.0095	0.0126 0.0125 8.2076e-004	0.0126 0.0056 0.0202	0.0126 0.0093 0.0167	0.0126 0.0122 0.0019	0.0126 0.0119 0.0022
F5	Mejor Media Desv. estándar	0.0088 0.0177 0.0017	0.0087 0.0175 0.0016	0.0088 0.0176 0.0030	0.0088 0.0176 0.0017	0.0088 0.0176 0.0023	0.0088 0.0176 0.0018	0.0088 0.0176 0.0018
F6	Mejor Media Desv estándar	20.5059 21.1305 5.5776	20.5056 20.5059 2.1256e-004	20.5056 20.5056 2.7251e-004	20.5058 20.5058 2.9527e-004	20.5056 21.3122 6.6067	20.5057 20.5057 2.9921e-009	20.5056 20.5065 0.0079
F7	Mejor Media Desv. estándar	4.440e-013 0.63910414 4.53297280	0 -9.4739e-017 4.7168074	0 -0.1373e-016 1.681145	0 -1.4211e-016 1.4e-015	0 -0.1156e-016 1.4157	0 -3.0790e-016 1.0e-015	0 -2.0269e-006 2.46e-005 0.8708
F8	Mejor Media Desv. estándar	5.19615242 5.19615242 0.00000000 0004033	5.1961524 5.1961524 0	5.196152 5.196152 0.000000 0010797	5.196152 5.196152 0.000000 0024104	5.196152 5.196152 0.000000 0017081	5.196152 5.196152 0.000000 0018928	5.1961524 5.1961524 0.0000000 06832787

Resultados experimento 2. Se puede apreciar en la Tabla 5 que a mayor número de iteraciones, el algoritmo más se acerca al valor óptimo. Esto se debe a que cada partícula tiene más oportunidades de ajustar su velocidad y posición, pero el realizar más iteraciones tiene un costo computacional mayor, este es un punto muy importante que se debe tener en cuenta en el análisis de este tipo de funciones ya que se relaciona con la eficiencia en la búsqueda de la solución del problema. Algo importante para resaltar es que las funciones de prueba con tres variables necesitan más iteraciones para llegar al óptimo que en caso de 2 variables (ya que el algoritmo necesita interactuar más veces para encontrar el valor deseado).

Al observar el experimento 2 también se pudo observar que el cambio en el promedio de los valores óptimos de cada función en las evaluaciones, presentan un límite por encima del cual su valor no cambia considerablemente de acuerdo a tolerancias establecidas en las funciones y las restricciones. Estas tolerancias fueron de $1e-006$ para la evaluación de las funciones y las restricciones. La Tabla 6 ilustra esta condición.

Tabla 6. Promedio límite de iteraciones en la evaluación de la función y las restricciones para cada función objetivo.

Función objetivo	Promedio límite de iteraciones en la evaluación de la función sin presentar cambio alguno.	Promedio límite de iteraciones en la evaluación de la restricción sin presentar cambio alguno.
F1	50	73
F2	23	82
F3	35	60
F4	60	121
F5	50	80
F6	30	89
F7	50	111
F8	25	51

Se pudo apreciar que a mayor número de iteraciones, el algoritmo más se acercaba al valor óptimo. Esto se debe a que cada partícula tiene más oportunidades de ajustar su velocidad y posición, pero el realizar más iteraciones tiene un costo computacional mayor, este es un punto muy importante que se debe tener en cuenta en el análisis de este tipo de funciones ya que se relaciona con la eficiencia en la búsqueda de la solución del problema.

5.4 EXPERIMENTO 3 PSO: COMPORTAMIENTO DEL FACTOR INERCIA Y EL FACTOR DE ESTRECHAMIENTO.

La velocidad máxima (v_{max}) sirve como una restricción para controlar la habilidad de exploración global de un enjambre de partículas. Un valor grande de v_{max} facilita la exploración global. El concepto de factor de inercia se desarrolló

para mejorar el control de la exploración de las partículas. La inclusión del factor de inercia pudo verse primero reportado en [11].

Este experimento realizó un análisis de PSO para corroborar su comportamiento al modificar otro de sus parámetros. Lo importante de este análisis fue que para que el rendimiento de PSO fuera el deseado, se debía seleccionar aparte de la cantidad de partículas y el número de iteraciones, el valor de la velocidad máxima (V_{max}) que puede adquirir una partícula, ya que con este parámetro, se aseguraba que las partículas se acercaran más al óptimo. Lo complejo de la selección de este parámetro es que, para el caso de aplicar PSO a la búsqueda de óptimos en funciones no-lineales, se debe conocer dicha función, ya que V_{max} depende directamente de la función. Como solución a este inconveniente surgió un coeficiente, llamado coeficiente de inercia [8], nombrado ya anteriormente, el cual se multiplica en la fórmula de la velocidad con el valor de la velocidad anterior para solucionar el problema de la convergencia. En los experimentos a realizar se observó como el coeficiente de inercia puede depender de la cantidad de iteraciones o generaciones.

- **Relación lineal del factor de inercia variable con la cantidad de iteraciones**

Para probar la relación lineal entre el valor del factor de inercia y la cantidad de iteraciones presente en el método de PSO, se analizaron las funciones objetivo F2 y F3 con una población de 200 partículas. Las Figuras 15 y 16 muestran el comportamiento lineal en el transcurso de las iteraciones hasta que el algoritmo encuentra el valor óptimo para cada caso.

Figura 15. Decrecimiento lineal del factor de inercia vs. Cantidad de generaciones para F2

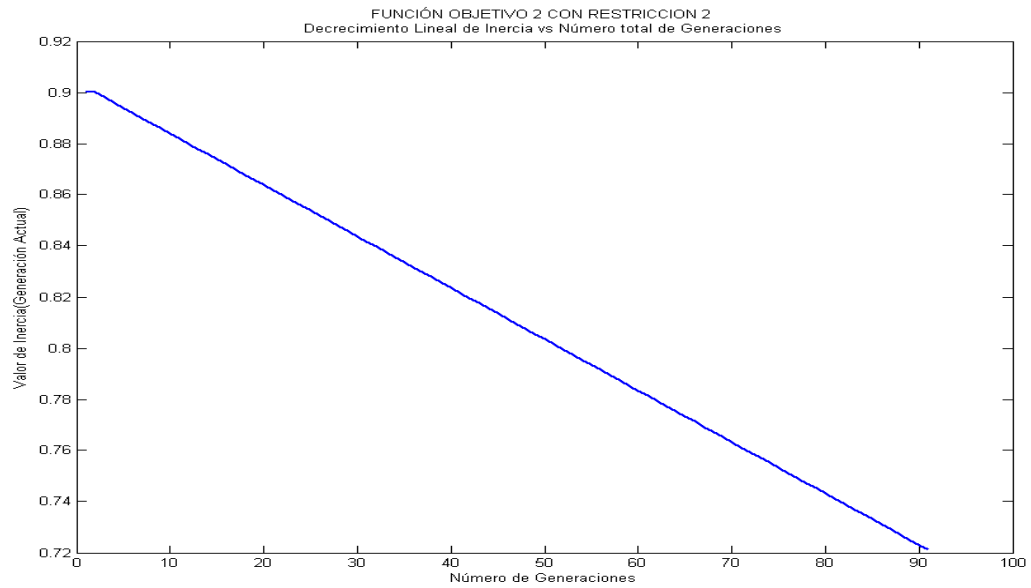
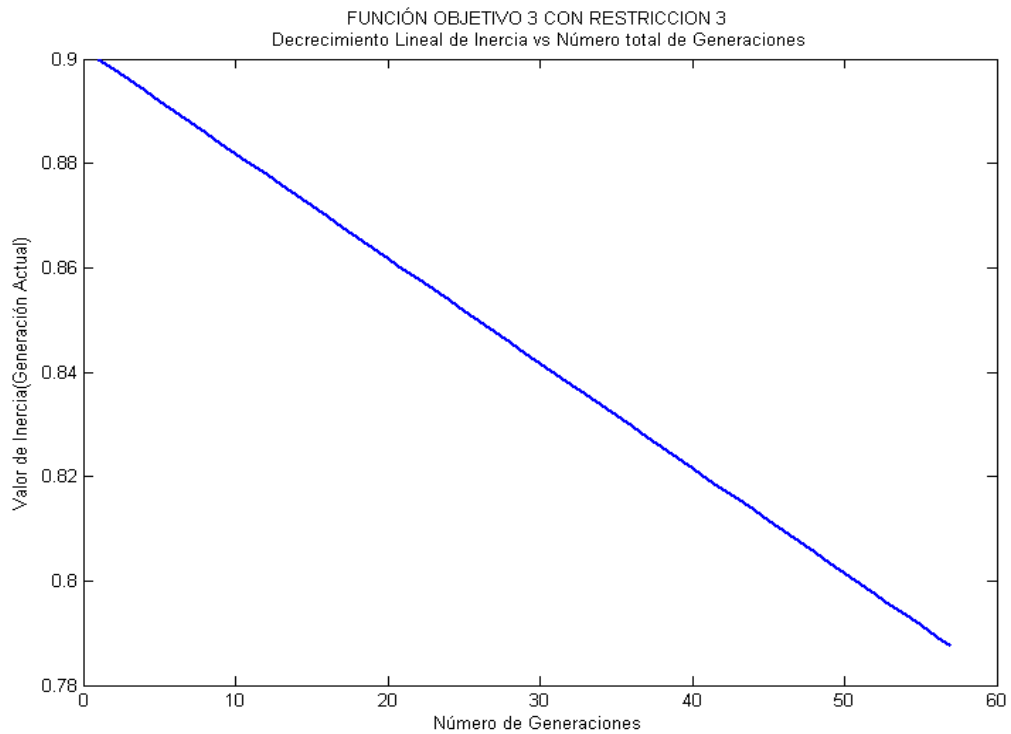


Figura 16. Decrecimiento lineal del factor de inercia vs. Cantidad de generaciones para F3



La ecuación que representa este comportamiento lineal es:

$$Factor\ Inercia\ (actual) = Fact.Inercia_max - Fact.Inercia_min \frac{(Iteración(actual) - 1)}{(Iteraciones\ totales - 1)}$$

$$Factor\ Inercia\ (actual) = 0.9 - 0.4 \frac{(Iteración(actual) - 1)}{(Iteraciones\ totales - 1)}$$

La Tabla 7 muestra un estudio comparativo entre PSO con restricciones para variantes de factor de inercia y factor de inercia + factor de estrechamiento, en 33 ejecuciones independientes de las funciones de prueba. Los resultados obtenidos se calcularon con N= 200 partículas y lter=200 iteraciones máximas.

Tabla 7. Variantes de PSO con factor de inercia y factor de estrechamiento. Estudio estadístico de 33 ejecuciones independientes.

Función Objetivo	Parámetros estadísticos	Variantes	
		PSO con factor de inercia variable	PSO con factor de inercia variable + factor de estrechamiento
F1	Mejor Media Desviación estándar	0.368701358146784 0.641767396014524 0.458483046417239	0.357240386247340 0.559527966582097 0.190645572417188
F2	Mejor Media Desviación estándar	5.35135133559053 5.35134941289651 4.1296726672e-006	5.35135137004329 5.35134490573591 1.1591992734e-006
F3	Mejor Media Desviación estándar	0.166935632419330 0.529748237273121 0.244577016679588	0.130021316201593 0.169531098806406 0.025372963820431
F4	Mejor Media Desviación estándar	0.01260017438982 0.01557295104185 0.01655168119222	0.01260017428457 0.01260021003583 8.6619506395e-010
F5	Mejor Media Desviación estándar	0.005674930292873 0.041043079217641 2.2659485538e-008	0.003675950378268 0.041043065269892 1.6950816597e-008

F6	Mejor Media Desviación estándar	20.24356476876556 20.79331946147473 0.899974329195106	20.12985746398 23.35792013579649 4.01405576094658
F7	Mejor Media Desviación estándar	0.537485739109 0.54270494045131 0.89997432919510	0.46735647837932 0.48240439131796 0.86600070215346
F8	Mejor Media Desviación estándar	5.196152422594165 5.028800772065689 0.002473727998142	5.196152420450249 5.028017557320797 4.835923992414693e-004

Resultados experimento 3. Al comparar el valor de PSO con el factor inercia solamente y luego comparar los valores de PSO implementando el factor de estrechamiento en el algoritmo, se pudo observar que sin factor de estrechamiento el algoritmo llega a una respuesta aproximada pero al agregar el factor de estrechamiento, la solución de los problemas converge de una mejor forma, esto comprobó lo ya revisado en la literatura especializada referente a este tipo de variantes.

5.5 EXPERIMENTO 4 PSO: COMPORTAMIENTO DEL ALGORITMO AL AUMENTAR EL NÚMERO DE VARIABLES.

Este experimento se realizó solamente para la función de Rosenbrock's Valley (F5), en el cual se pretendió determinar qué tan eficiente era PSO al trabajar con un número de variables elevado y que variación debía ocurrir en los parámetros de la población y cantidad de iteraciones para la solución de funciones no-lineales con restricciones. Se establece como parámetro de operación fijo 5000 iteraciones máximas.

Tabla 8. Variación de los valores óptimos de PSO en F5 al cambiar el número de variables.

Cantidad de partículas	Cantidad de variables				
	NVAR=2	NVAR=3	NVAR=5	NVAR=7	NVAR=10
200	0.0088	0.0410	0.5188	2.1829	21.4834
300	0.0088	0.0410	0.4299	2.1409	5.4689
500	0.0088	0.0410	0.4299	1.8563	5.1295
1000	0.0088	0.0410	0.4299	1.8563	4.8156
5000	0.0088	0.0410	0.4299	1.8563	4.8095
10000	0.0088	0.0410	0.4299	1.8563	4.8095

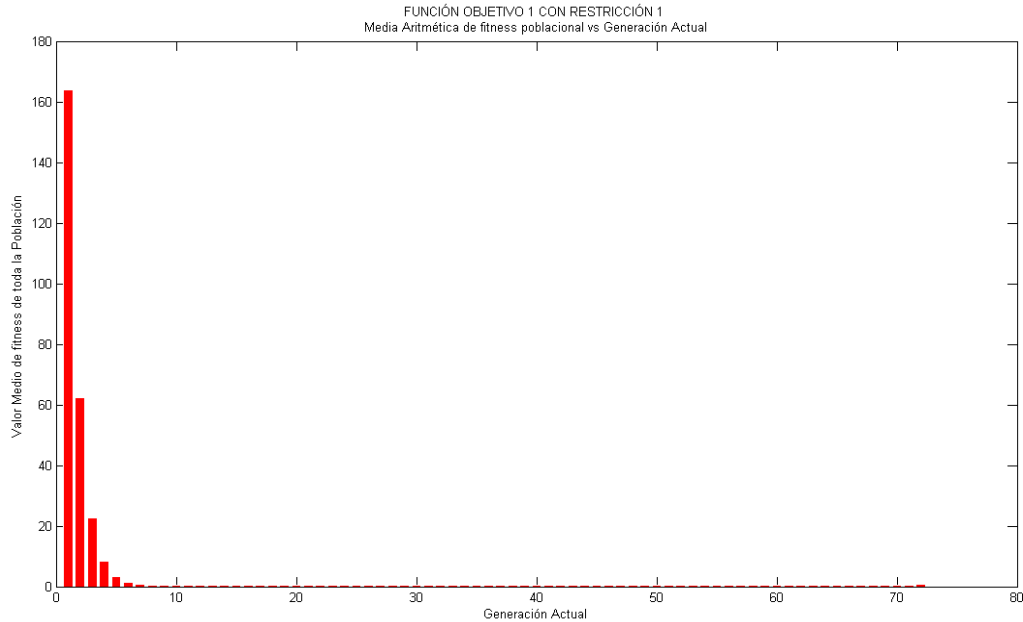
Resultados experimento 4. Se puede concluir del experimento realizado y de los resultados obtenidos en la Tabla 8, que al aumentar el número de variables en PSO hace falta aumentar la cantidad de partículas para que el algoritmo pueda converger al valor óptimo y reduzca su error, ya que se necesitan más partículas que puedan realizar en el espacio de búsqueda indicado la interacción para llegar a una mejor aproximación de la respuesta, mediante la utilización del conocimiento global y su conocimiento propio de la misma. También se observa que para el caso de más de 3 variables, el algoritmo PSO no llega a la respuesta óptima sino a un aproximado a pesar de aumentar la cantidad de partículas y de iteraciones a un valor muy grande. Esto es porque la optimización por enjambre de partículas no nos asegura la convergencia al valor deseado, pero cuando llega a un valor aproximado lo hace de manera muy rápida. En el caso del uso de dos y tres variables, el algoritmo convergió con una cantidad de iteraciones menor a 200.

5.6 ESTUDIO ESTADÍSTICO PSO

A continuación se realizó un estudio estadístico de las funciones de prueba F1, F7 y F8 mediante criterios tales como el valor medio y la desviación estándar en el transcurso de las generaciones para la población establecida. También se llevó a cabo el cálculo del error medio cuadrático de los valores óptimos de cada partícula en el transcurso de todas las iteraciones hasta llegar al valor óptimo. Las gráficas de valores medios, desviación estándar y error medio cuadrático para las demás funciones se presentan en el anexo A. En las gráficas el término fitness hace referencia al valor óptimo de la solución del problema.

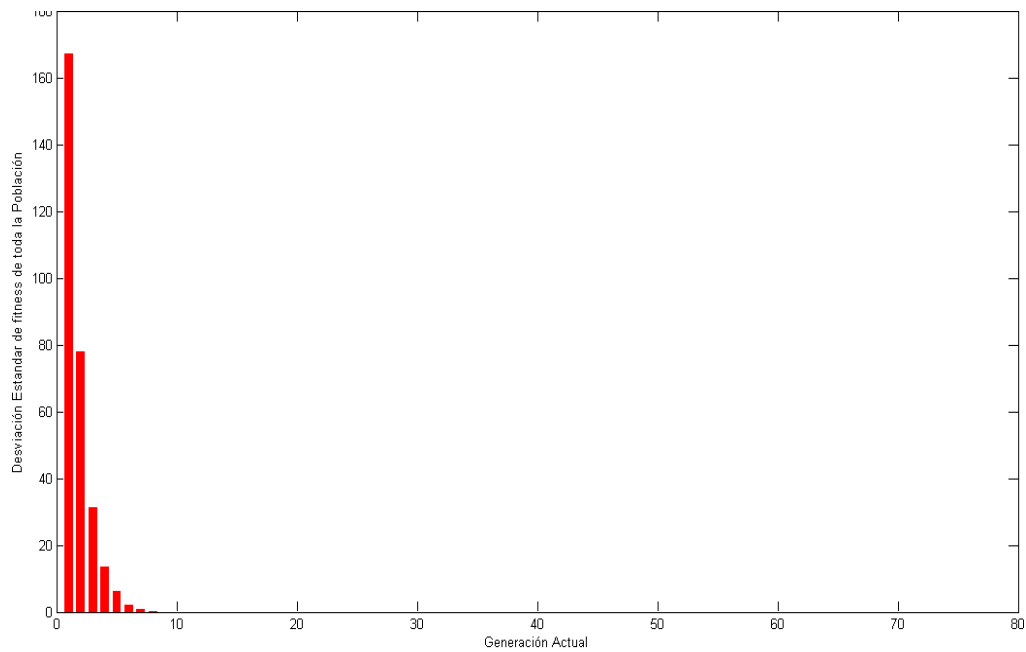
VALORES MEDIOS, DESVIACIÓN ESTÁNDAR Y ERROR MEDIO CUADRÁTICO PARA LAS FUNCIONES OBJETIVO

Figura 17. Media aritmética de fitness poblacional vs. Generación actual para F1



La Figura 17 mostró el comportamiento de los valores medios del óptimo de la función objetivo para cada iteración, con una población de 500 partículas. Ya que en la primera iteración de las partículas no se ven involucrados los factores de inercia y estrechamiento por falta de condiciones en un estado anterior, los valores obtenidos en estos pasos suelen ser muy erróneos. Pero pasadas las generaciones debido a que se ha involucrado en la ecuación de velocidad el factor de estrechamiento, el algoritmo llega de forma muy rápida a la respuesta en un rango de aproximadamente 20 iteraciones debido a que este factor ofrece un mejor control de la velocidad y asegura la convergencia garantizada como lo propuso Clerc en [31] para este tipo de problemas de optimización. Para la función objetivo 1 el valor teórico es de 0.3571 y el algoritmo alcanzó este valor de forma satisfactoria y rápida con variaciones mínimas entre la iteración 25 y 70 y con una tolerancia de $1e^{-6}$ en los valores de la evaluación de la función y las restricciones. El algoritmo no presentó cambios significativos en evaluaciones de la función óptima después de 20 generaciones ni violación en la evaluación de las restricciones después de 73 generaciones.

Figura 18. Desviación estándar de fitness poblacional vs. Generación actual para F1



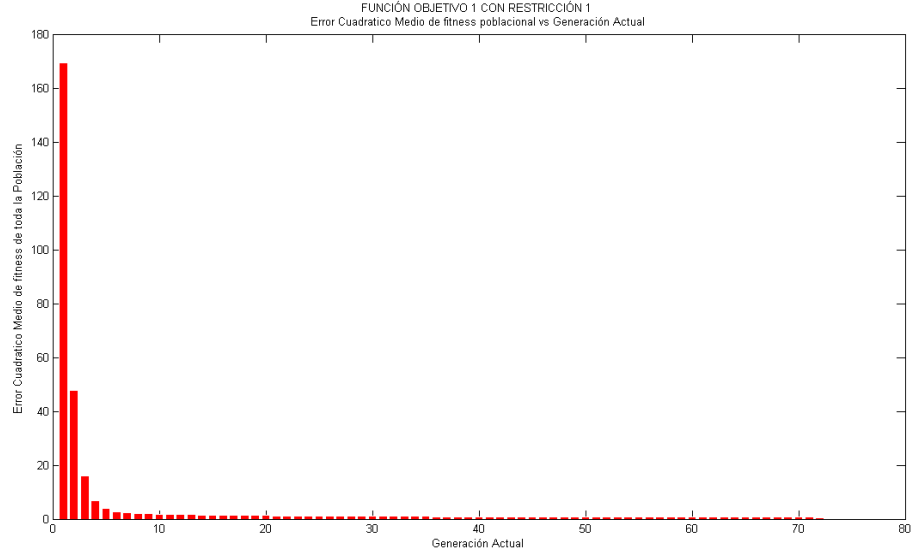
Debido a que la desviación estándar es simplemente el "promedio" o variación esperada con respecto a la media aritmética, se puede observar en la Figura 18 una desviación significativa para las primeras 10 generaciones, lo cual confirma como los datos para este periodo se mantienen dispersos con respecto al valor central de las partículas.

El error cuadrático medio se calculó mediante la ecuación:

$$EMC = \sqrt{\frac{1}{Ni} \sum_{i=1}^N (yi - \bar{Y}_{teorico})^2}$$

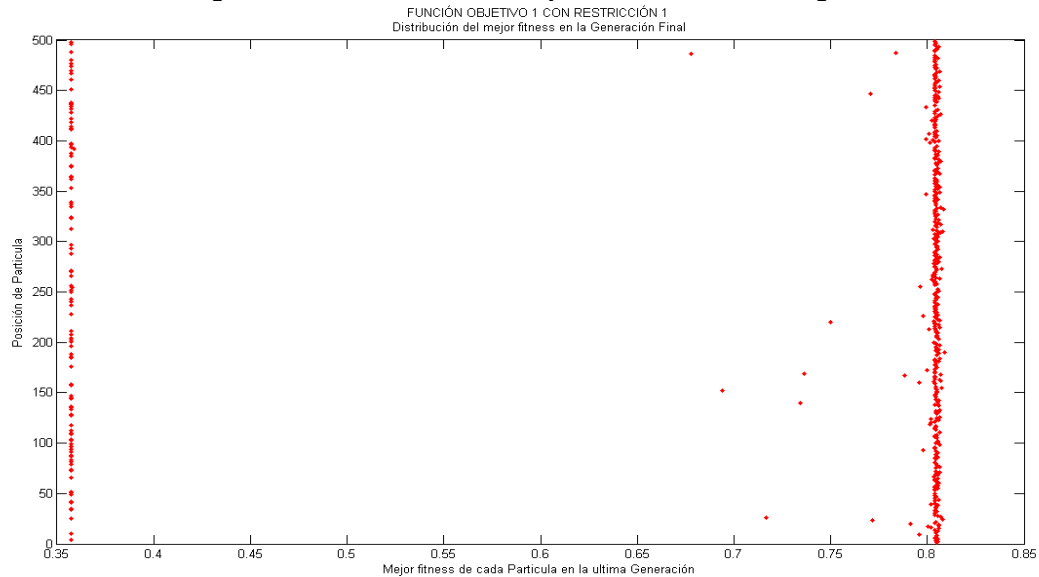
Donde Ni determina el valor de la iteración actual. Este error se calculó así debido a que el algoritmo PSO no finaliza siempre con la cantidad de iteraciones establecidas inicialmente, sino que lo hace cuando el algoritmo haya encontrado la mejor respuesta con la tolerancia establecida para la evaluación de las funciones y las restricciones.

Figura 19. Error medio cuadrático de fitness poblacional vs. Generación actual para F1



Esta medida es un método práctico y sencillo para conocer la precisión del conjunto de valores de la media de los errores cuadráticos. Se toma una aproximación de la Figura 19 que el EMC para las generaciones superiores a la generación 20 es cercano a $0.36e^{-3}$. Esto concuerda con los datos analizados anteriormente donde se observa que el algoritmo converge a la respuesta para esta función objetivo de una manera muy rápida después de 20 generaciones.

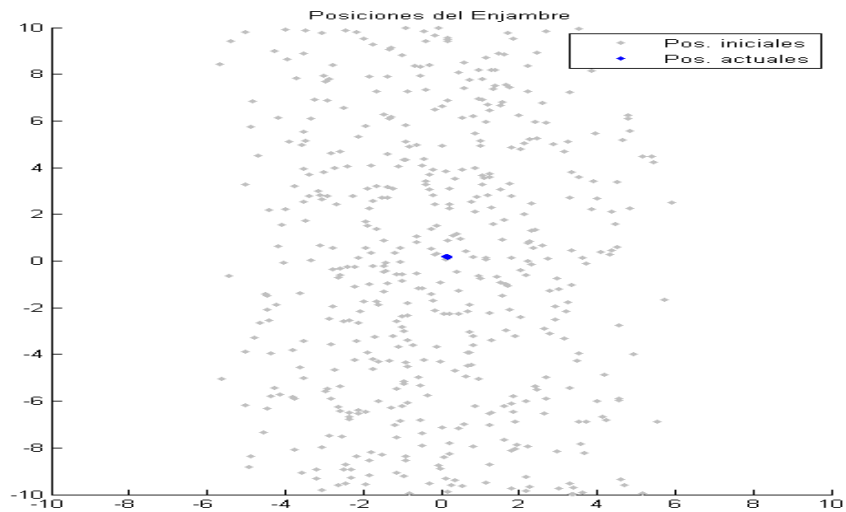
Figura. 20. Distribución del mejor fitness en la última generación.



La Figura 20 representa la distribución de los mejores valores fitness de las 500 partículas en la última generación (iteración). Como se pudo observar hay una

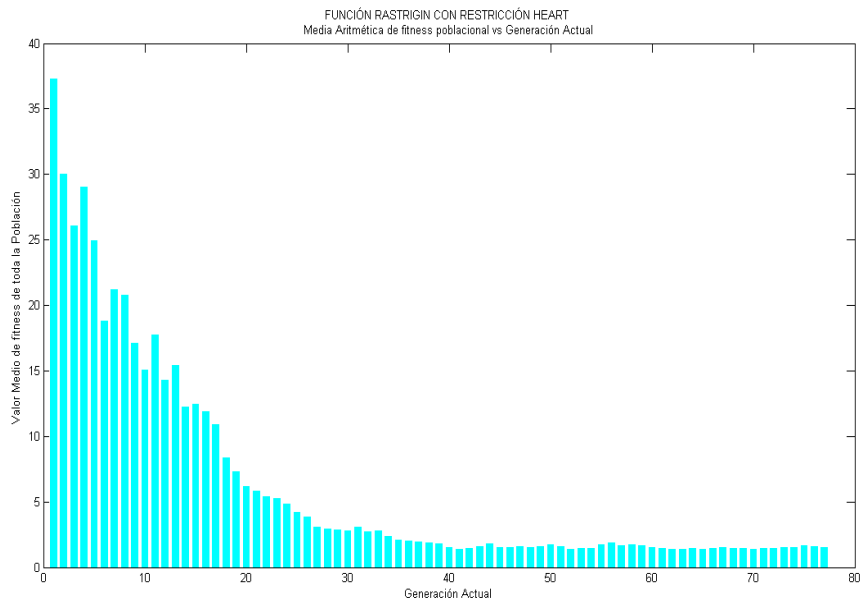
cantidad dominante de partículas que no llegaron al valor fitness, esta es una situación no deseada y una desventaja de PSO en algunos casos, sin embargo el algoritmo de PSO pudo encontrar la respuesta con una cantidad de partículas significativas debido a que tomo como respuesta el mejor global.

Figura 21. Distribución de las posiciones de las partículas en el enjambre desde las posiciones iniciales hasta las posiciones de los valores óptimos (actuales).



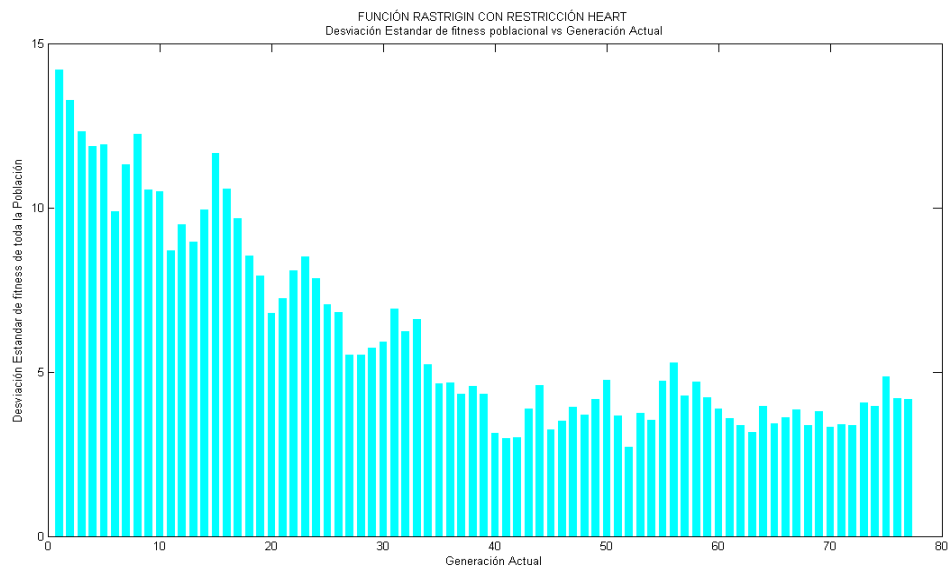
La Figura 21 muestra el camino de las partículas para llegar al mejor valor de evaluación de la función objetivo y las restricciones desde las posiciones iniciales hasta las posiciones de los valores óptimos (actuales).

Figura 22. Media aritmética de fitness poblacional vs. Generación actual para F7



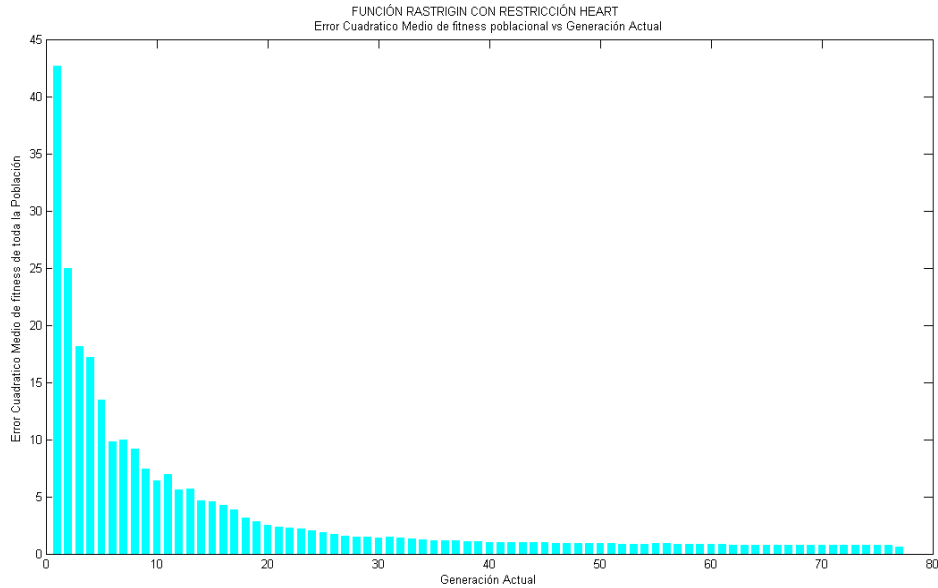
La función Rastrigin (F7) es una función con la que el método meta-heurístico PSO falla aproximadamente en el 30% del total de simulaciones. Sin embargo para el 70% por ciento restante el algoritmo alcanza a entregar una solución que está en un rango aceptable. Para este y otros casos en que la función es tan dinámica, el método no es viable, debido a que tiene que redistribuir partículas y cambiar drásticamente su velocidad para poder converger rápidamente. En la Figura 22 se muestra el caso específico en el que una variable de la función objetivo no pudo llegar al óptimo, por tal motivo el valor medio final no es cercano al valor teórico (cero).

Figura 23. Desviación estándar de fitness poblacional vs. Generación actual para F7



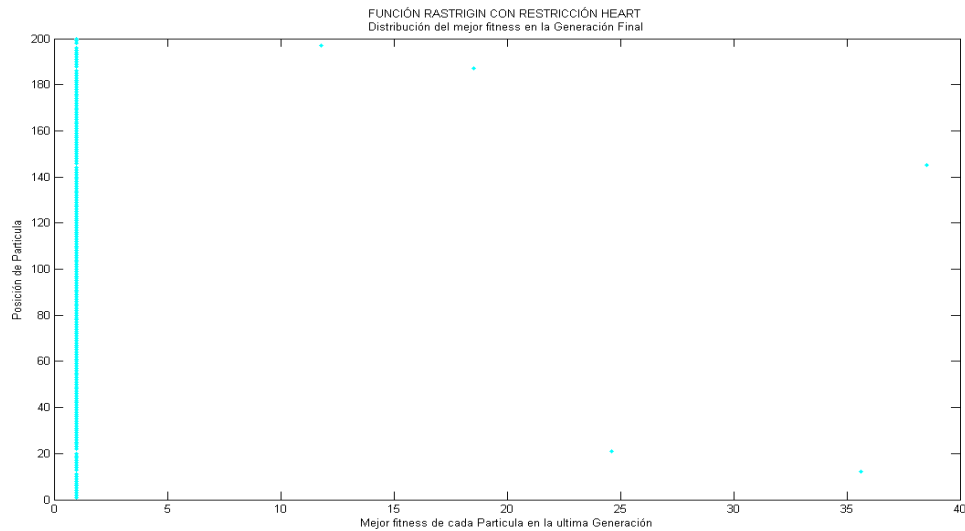
La curva marcada de desviación estándar de la solución detalla aún más la dinámica de la función Rastrigin. En el transcurso de las generaciones las partículas tienen que variar su velocidad drásticamente cada ocho iteraciones en promedio, debido a las propiedades de la función cosenoidal que dan como resultado una desviación muy alta con respecto al valor medio.

Figura 24. Error medio cuadrático de fitness poblacional vs. Generación actual para F7



Aunque el error cuadrático tiene una caída de energía en el transcurso de las generaciones, no alcanzó a ser lo suficientemente mínimo como para considerar que la solución de PSO este cercana al valor teórico que es igual a cero.

Figura 25. Distribucion del mejor fitness en la generación final para F7.



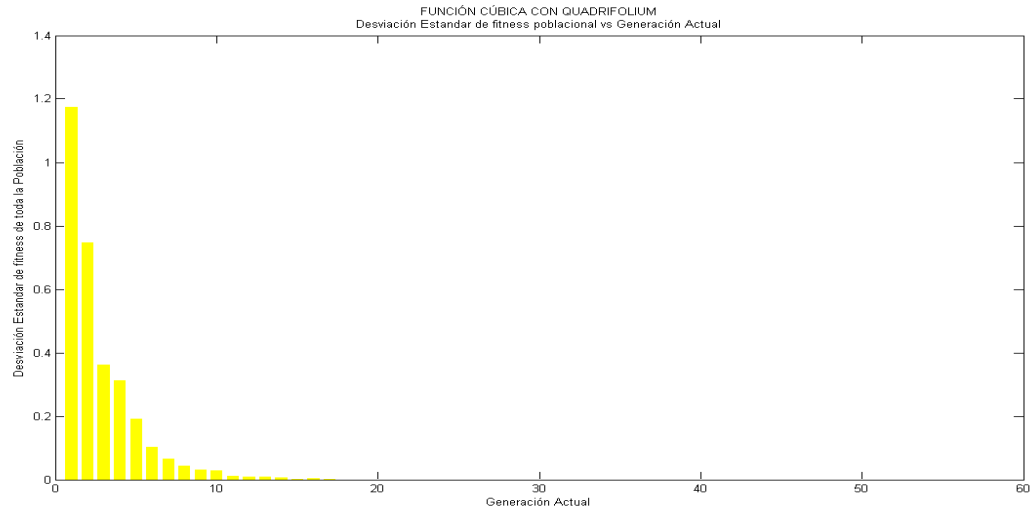
Como se observa en la Figura 25, un estancamiento de la función por encima de 0.90 es debido a la variable x_1 en el transcurso de todas las generaciones, al no poder redireccionar la posición de las partículas para modificar su velocidad y tratar de buscar el punto más óptimo posible.

Figura 26. Media aritmética de fitness poblacional vs. Generación actual para F8



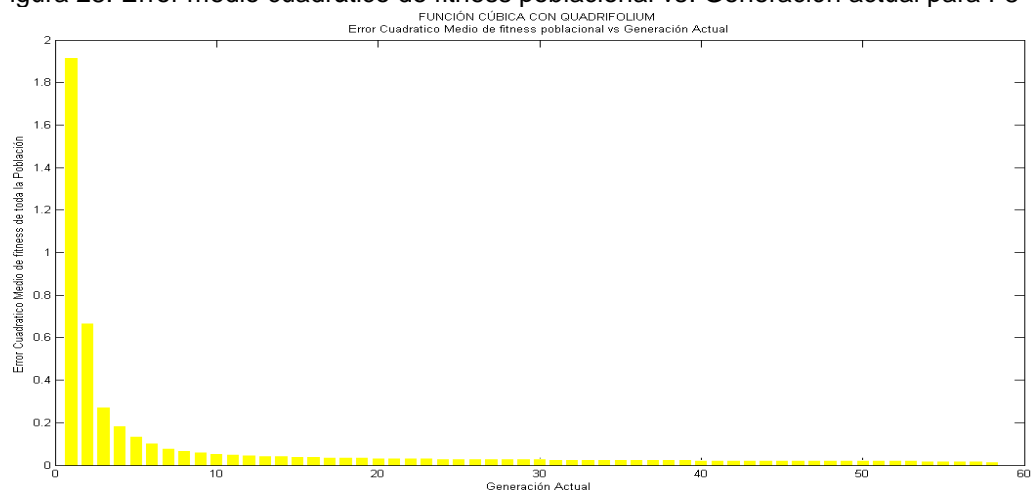
Para el caso de la función cubica se pudo observar que el algoritmo convergió rápidamente al valor medio en el transcurso de las primeras generaciones sin presentar cambio alguno, ya que la dispersión de las partículas fue mínimo a partir de la generación 12.

Figura 27. Desviación estándar de fitness poblacional vs. Generación actual para F8



De acuerdo a los resultados obtenidos en la gráfica de valor medio se puede observar que para el caso de la desviación estándar los valores se encuentran cercanos a cero en generaciones muy tempranas, debido al establecimiento de óptimos en un valor cercano a cinco.

Figura 28. Error medio cuadrático de fitness poblacional vs. Generación actual para F8



Para F8 el error cuadrático medio se estabilizó a un valor muy bajo como resultado de una solución muy cercana a la esperada.

6. ESTUDIO COMPARATIVO ENTRE EL METODO DE OPTIMIZACION POR ENJAMBRE DE PARTÍCULAS (PSO) Y EL METODO DE OPTIMIZACION DE ANÁLISIS DE INTERVALOS.

Recordando que PSO es una heurística inspirada en estudios y comportamientos de la naturaleza mediante la interacción de grupos de individuos que satisfacen unos objetivos propuestos, partiendo de forma aleatoria y basándose en el conocimiento individual y grupal; destacando también que el método de análisis de intervalos para optimización global es un método determinístico que utiliza los principios del método de bisección para encontrar soluciones mediante el descarte de regiones donde pueda no hallarse un óptimo y con base en la documentación y observación de estos comportamientos, se propuso hacer el estudio correspondiente en este documento.

Así, se hizo necesario realizar y evaluar mediante criterios estadísticos certeros la eficiencia de cada método y las ventajas y desventajas que cada uno proveía en la solución de problemas de optimización de funciones no lineales de 1, 2 y 3 variables sujetas a restricciones de igualdad y /o desigualdad. Para ello en la Tabla 9 se plantearon inicialmente una serie de ventajas y desventajas en base a la documentación realizada [42], [58] y como punto de comparación para el estudio estadístico de este capítulo.

Tabla 9. Estudio característico de ventajas y desventajas de PSO y la optimización por análisis de intervalos.

PSO		Análisis de intervalos	
Ventajas	Desventajas	Ventajas	Desventajas
Tiene buen desempeño, baja complejidad computacional y un mínimo de parámetros. El procedimiento es genérico, no depende del tipo de problema.	Son algoritmos aproximados de optimización, no exactos.	Es una técnica exacta que garantiza encontrar la solución óptima de cualquier problema.	Sería un método idóneo si no tuviera el inconveniente de la cantidad de tiempo necesario para la resolución. El tiempo crece exponencialmente con el tamaño del problema.

Tabla 9 (continuación)

Suele tener una convergencia rápida a buenas soluciones.	No necesariamente encuentran la solución óptima pero si la más aproximada.	Presenta mejor calidad en la solución de problemas para cualquier tolerancia establecida.	Se corre un alto riesgo cuando se utiliza experimentalmente en algunos problemas reales que requieren rápida reacción.
Combina de forma inteligente distintos conceptos para explorar y explotar adecuadamente el espacio de búsqueda.	En algunos casos ni siquiera puede determinar la cercanía al óptimo de una solución factible.	Tiene su propio análisis matemático en la literatura especializada que lo respalda.	La mayoría de las personas no están aún familiarizadas con el método.
Es capaz de proporcionar muy buenas soluciones en tiempo y con recursos razonables. Integra de diversas maneras, procedimientos de mejora local y estrategias de alto nivel para crear un proceso capaz de escapar de óptimos locales y realizar una búsqueda robusta en el espacio de búsqueda.	Sigue teniendo problemas de dependencia de las condiciones iniciales.	Puede trabajar paralelamente con múltiples métodos de optimización para intervalos.	Los paquetes de aritmética de intervalos, actualmente tienen implementación muy lenta.

El estudio comparativo de los métodos evaluados en las 9 funciones objetivo se distribuyó realizando un análisis estadístico para funciones de 1, 2 y 3 variables. Los resultados del estudio de los métodos generaron una serie de discusiones que serán tratadas más adelante. Para el caso de las funciones F1, F2, F3, F4 y F6 los valores óptimos calculados fueron valores máximos globales de las funciones sujetos a restricciones. En el caso de F5, F7, F8 y F9 los valores óptimos

calculados fueron valores mínimos globales de las funciones sujetos a restricciones. Los datos obtenidos fueron consignados y para el caso de 2 variables los resultados gráficos fueron plasmados en este documento.

Estudio comparativo para 1 variable. En Tabla 10 se consignaron los resultados del mejor valor y tiempo promedio de cómputo para la funciones F7 y F8. Los resultados obtenidos se calcularon con una población de 200 partículas y 300 iteraciones máximas en el caso de PSO. Los tiempos de cómputo se calcularon mediante las funciones *tic-toc* de MATLAB. *Tic* guarda el tiempo actual que *toc* usa más adelante para medir el tiempo transcurrido entre los dos. La ubicación de *tic* y *toc* se hizo al principio y final del algoritmo principal para el cálculo de óptimos por el método de análisis de intervalos.

Tabla 10. Resultados de la comparación de los métodos de PSO y análisis de intervalos para 1 variable.

Función Objetivo	Criterios Estadísticos	PSO	Análisis de Intervalos
F7 (restricción $x_1 - 1 \leq 0$)	Mejor valor Tiempo computo promedio	$5.329070518200 \times 10^{-15}$ 2.843846 seg.	$[-3.5527136788005 \times 10^{-15}, 5.3290705182007 \times 10^{-15}]$ 2.526689 seg,
F8 (restricción $x_1 - 2 \leq 0$)	Mejor valor Tiempo computo promedio	-2.59807621135332 2.418248 seg.	$[-2.596235931152478, -2.599207641556859]$ 87.871776 seg.

Discusión y análisis de los resultados. Se pudo observar de los resultados obtenidos, que tanto los valores de PSO como de análisis de intervalos llegaron a los óptimos deseados, con la diferencia que PSO llegó en un tiempo menor y este es un punto a favor importante en el manejo de la eficiencia en optimización global con restricciones. El error con respecto al valor exacto (el cual es -2.59 para el caso de F8) es menor para el método de análisis de intervalos que para PSO. La tolerancia que se maneja para la evaluación de las funciones y las restricciones en PSO es de $1e^{-6}$. Para el método de análisis de intervalos se trabajó con una tolerancia de $1e^{-3}$ para la evaluación de la función, lo que condujo a un resultado exitoso. En el caso de F7 el método de análisis de intervalo llegó a la respuesta

deseada en un tiempo menor que PSO, lo cual confirma su importancia en la solución de problemas de optimización.

Estudio comparativo para 2 variables. En la Tabla 11 se consignaron los resultados de valor medio, desviación estándar y tiempo promedio de cómputo para la funciones F2, F3, F4, F5, F6 y F7. Los valores fueron calculados mediante la utilización de los métodos de PSO y análisis de intervalos en la evaluación de las funciones objetivo sujetas a las restricciones preestablecidas. Los resultados obtenidos se calcularon con una población de 200 partículas y 300 iteraciones máximas en el caso de PSO . Los tiempos de cómputo se calcularon mediante las funciones *tic-toc* de MATLAB.

Tabla 11. Estadísticas de 33 ejecuciones independientes para los métodos de PSO y análisis de intervalos para 2 variables.

Función Objetivo	Criterios Estadísticos	PSO	Análisis de Intervalos
F2	Mejor valor. Tiempo computo	5.3513513700432 9 8.021036 seg.	[5.35135134512176, 5.35135135906026] 7.903387 seg.
F3	Mejor valor Media Desviación estándar Tiempo comp.	0.1300213162015 93 0.1695310988064 06 0.0253729638204 31 5.601491 seg.	[0, 0.193870456067116 × 10 ⁻²⁴] -1.5314425818616 7.931167687811831 12.247175 seg Tol 1 × 10 ⁻²⁴
F4	Mejor Media Desviación estándar Tiempo comp.	0.0126001742845 7 0.0126002100358 3 8.6619506395× 10 ⁻¹⁰ 11.291957 seg.	[-0.129246970711411 × 10 ⁻²⁴], 0.064623485355705 × 10 ⁻²⁴] 0.12931034482758 0.824424873586758 5.619032 seg Tol × 10 ⁻²⁴
F5	Mejor valor Media Desviación estándar Tiempo comp.	0.0088456782364 4 0.0546784386757 3 2.47876856457× 10 ⁻⁸ 7.823015 seg.	[0.008847395800163, 0.008847121191960] 6.12968983558042× 10 ³ 5.561904904716773× 10 ⁴ 305.890770 seg Tol 1 × 10 ⁻⁶

F6	Mejor valor Media Desviación estándar Tiempo comp.	122021.17883062 45 122420.76895047 8 11698.76859 1.564246	[3.162509151662930, 3.166758329056556] 4.385117557952302 × 10 ⁵ 1.457997908076409 × 10 ⁶ 1028.555556 seg.
F7	Mejor valor Media Desviación estándar Tiempo comp.	0 3.55271367880 × 10 ⁻¹⁵ 0.0878564536282 34 9.741469 seg.	[0.2421777760908131. × 10 ⁻⁴ , 0.0484355644125861 × 10 ⁻⁴ ,] 11.475591882909940 12.471612370469391 393.775467 seg. Tol 1 × 10 ⁻³

Discusión y análisis de los resultados. Al analizar los valores obtenidos en la Tabla 11 para las funciones objetivo de dos variables con restricciones, se deben nombrar algunos puntos importantes. Al realizar la comparación entre variantes de los dos métodos se pudieron observar ciertos comportamientos particulares en los resultados. Por ejemplo, en el caso de F3 el método PSO no llega a la respuesta exacta sino a un valor aproximado ya que el valor teórico óptimo para este caso fue el obtenido por el método de análisis de intervalos. Se observó que al aumentar la cantidad de partículas y de iteraciones en PSO el método no llegó a la respuesta deseada, esta es una desventaja importante de este método ya en que algunos casos no convergen a los valores óptimos sino que se establece en valores aproximados en muy corto tiempo. Para este ejemplo en particular el método de intervalos calculó el óptimo en un tiempo muy pequeño, lo que hace idóneo la aplicación y el resultado del mismo. En el caso de F5 (Rosenbrock's Valley) el método de análisis de intervalos llegó al mismo óptimo que el método de PSO con restricciones, con una diferencia de tiempo mucho menor para el caso de PSO, el cual resultó muy eficiente para este ejemplo. Para el caso de F6 el algoritmo de análisis de intervalos proporciona una respuesta más exacta pero en un tiempo de computo mucho más alto en comparación con PSO. En general en la mayoría de los resultados el método de análisis de intervalos para optimización global con restricciones obtuvo un mejor desempeño frente al método de PSO, lo que indica que es más eficiente usar este método para el tipo de problemas propuestos.

Estudio comparativo para 3 variables. En la Tabla 12 se consignaron los resultados de mejor valor y tiempo de cómputo para las funciones F1 y F9. Los resultados obtenidos se calcularon con una población de 200 partículas y 300 iteraciones máximas en el caso de PSO. Los tiempos de cómputo se calcularon mediante las funciones *tic-toc* de MATLAB.

Tabla 12. Estadística de 33 ejecuciones independientes para los métodos de PSO y análisis de intervalos para F1.

Función Objetivo	Criterios Estadísticos	PSO	Análisis de Intervalos
F1	Mejor valor	- 0.357240386247340	[-0.35714285714287, -0.35714285714285] 0.698706 seg.
	Tiempo computo promedio	14.466789 seg.	
F9	Mejor valor	0.03060123662791	[0.000400000000, 0.002400000000] 4.636704983660115
	Media	0.065481546193847	9.939545904980420
	Desviación estándar	0.001254758695840	408.009457 seg.
	Tiempo computo	8.851915 seg	

Discusión y análisis de los resultados. Al analizar los resultados de la comparación de los métodos planteados mediante la evaluación de las funciones de 3 variables para F1 y F9, se observó que los resultados obtenidos son mucho más precisos para el caso de optimización por análisis de intervalos que en el caso de PSO, ya que calculan el óptimo de manera más exacta con la desventaja de la cantidad de tiempo empleado en el proceso. Ya que PSO es un algoritmo que converge muy rápidamente a valores que pueden ser o no óptimos, se debe tener cuidado en la aplicación de este método y los riesgos que puede provocar con respecto a la precisión de las respuestas. El óptimo de F1 se calculó usando el método de intervalos de Newton.

Resultados Gráficos del método de análisis de intervalos para la optimización de funciones de 2 variables sujetas a restricciones.

En las siguientes gráficas se realiza un seguimiento de las pruebas realizadas en el método de análisis de intervalos (ver Figuras 29, 31, 33, 35 y 36). El descarte por prueba 2 se observa como una sola línea en la subregión reservada y el descarte por prueba 1 se observa como una cruz en cada subregión. El eje horizontal representa a x_1 en la gráfica y el eje vertical representa a x_2 . También

se presenta una descripción de los valores medios de cada función objetivo en el proceso de obtención del óptimo (ver Figuras 30, 32,34 y 37).

Figura 29. División del dominio original mediante las pruebas 1 y 2 para F3.

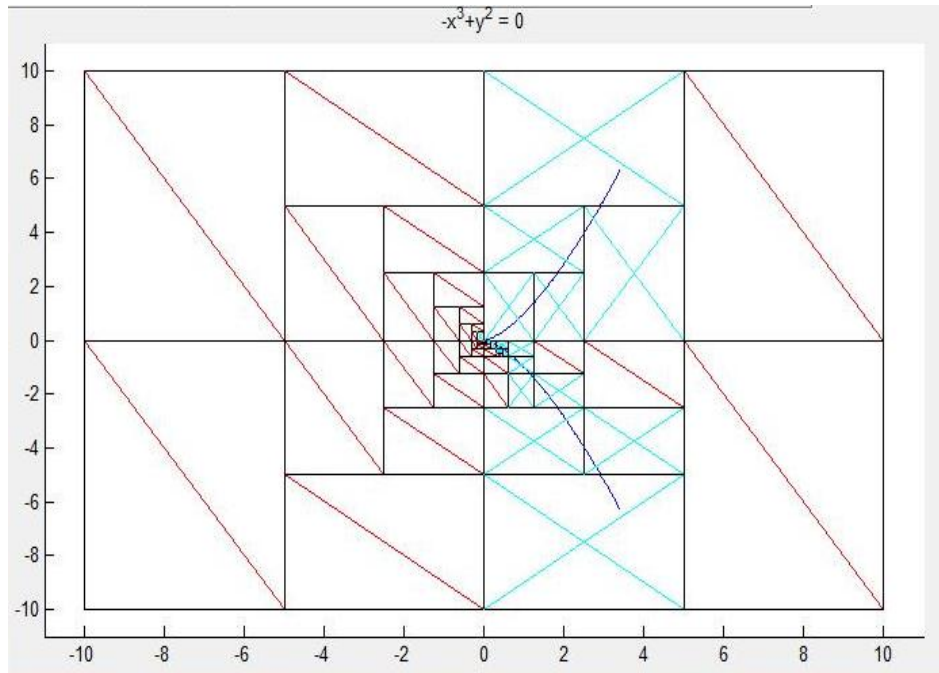


Figura 30. Valores estadísticos medios para F3.

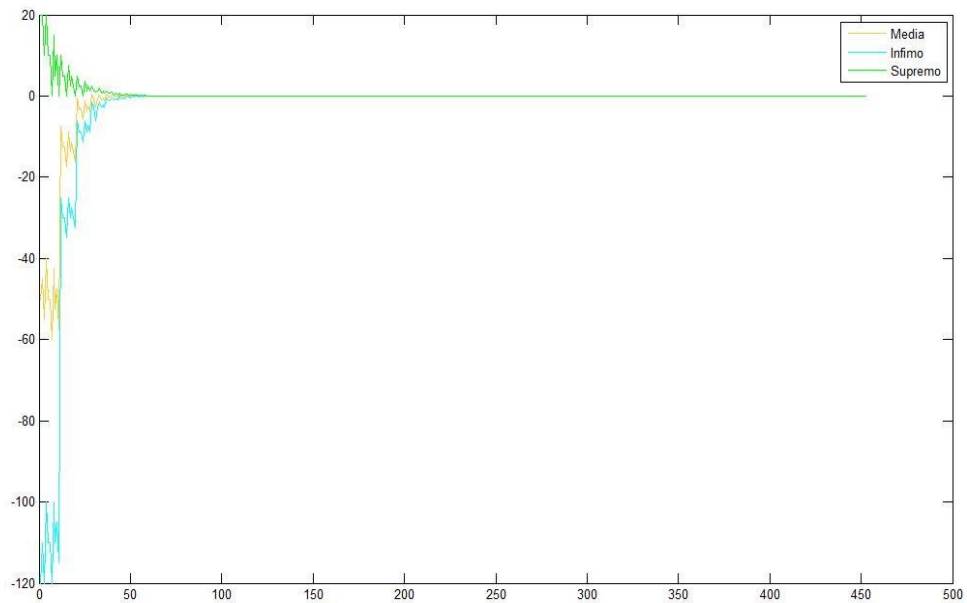


Figura 31. División del dominio original mediante las pruebas 1 y 2 para F4.

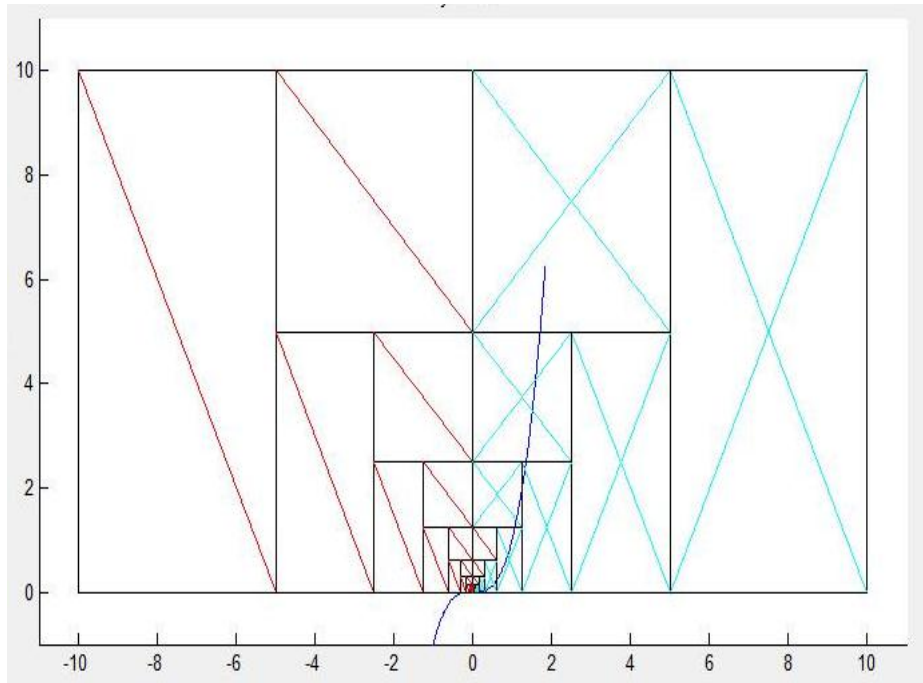


Figura 32. Valores estadísticos medios para F4.

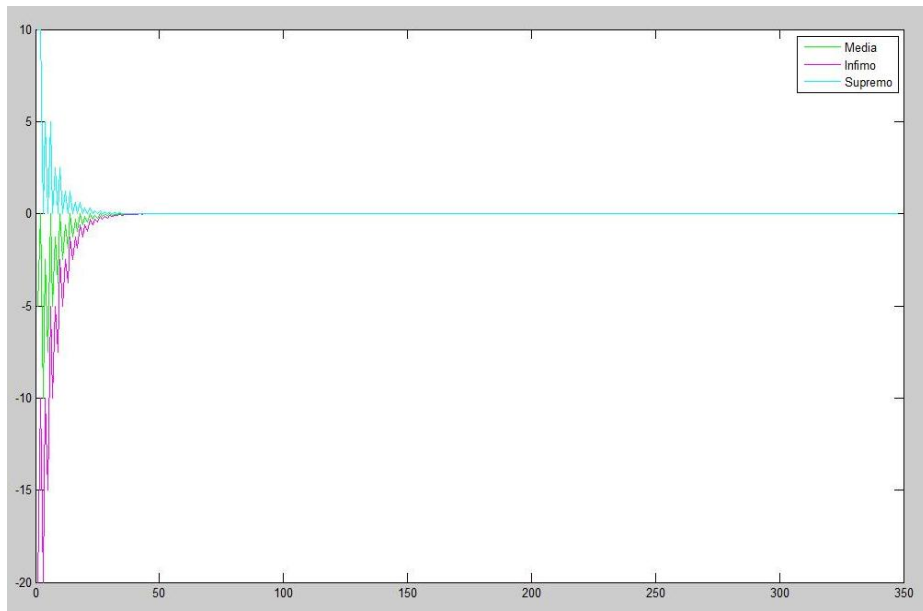


Figura 33. División del dominio original mediante las pruebas 1 y 2 para F5.

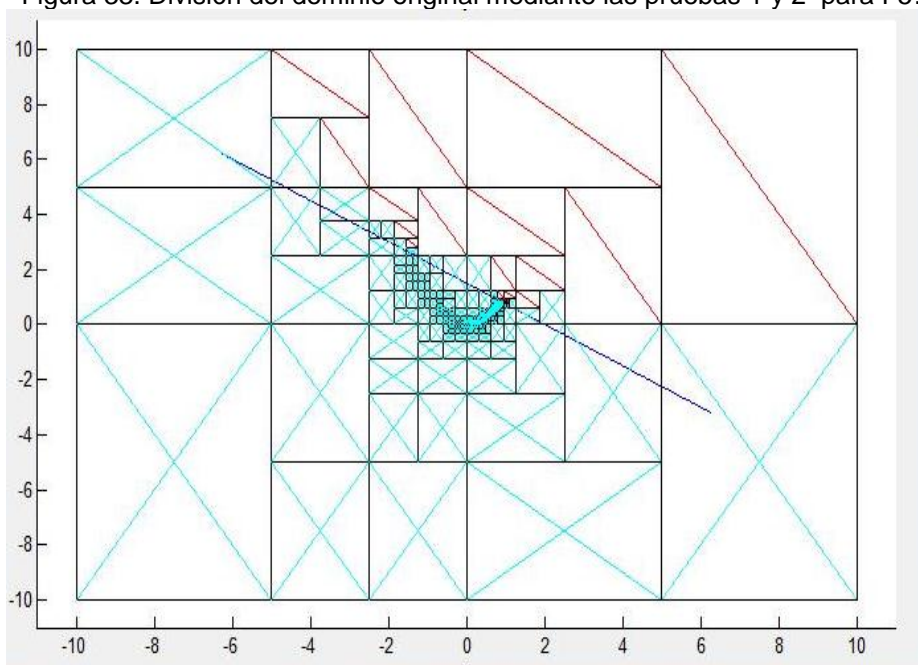


Figura 34. Valores estadísticos medios para F5

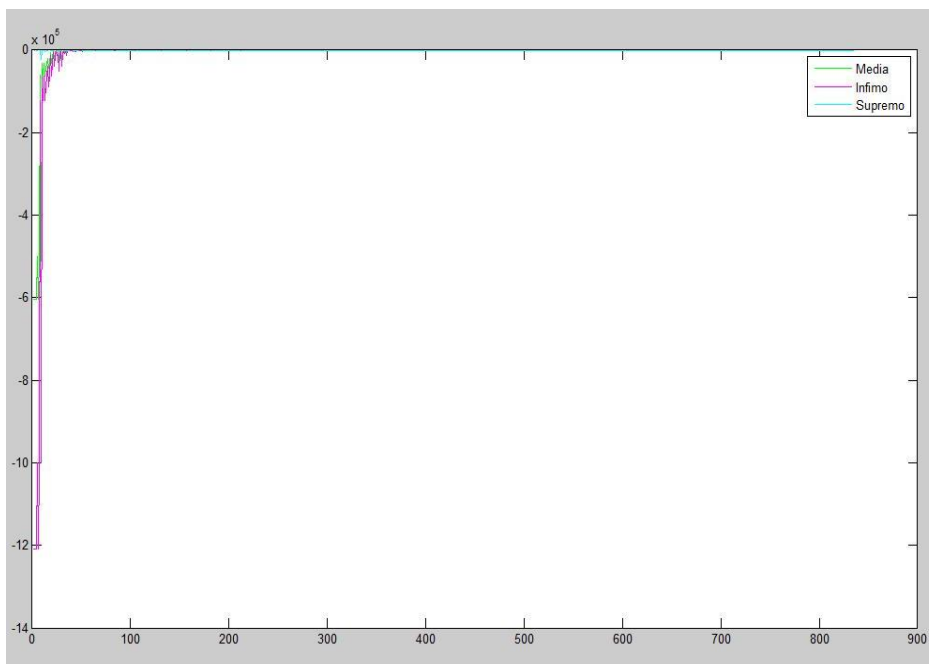


Figura 35. División del dominio original mediante las pruebas 1 y 2 para F6

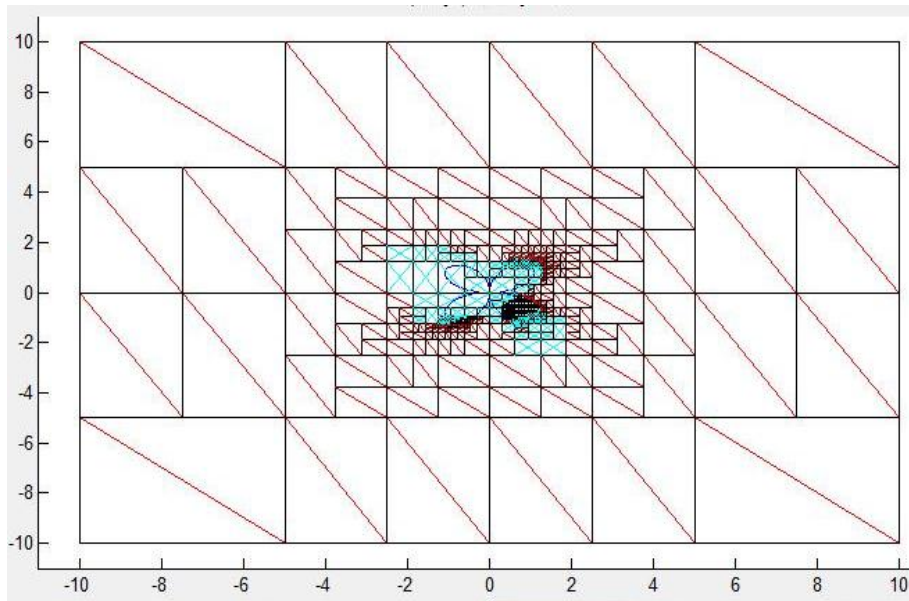


Figura 36. División del dominio original mediante las pruebas 1 y 2 para F7.

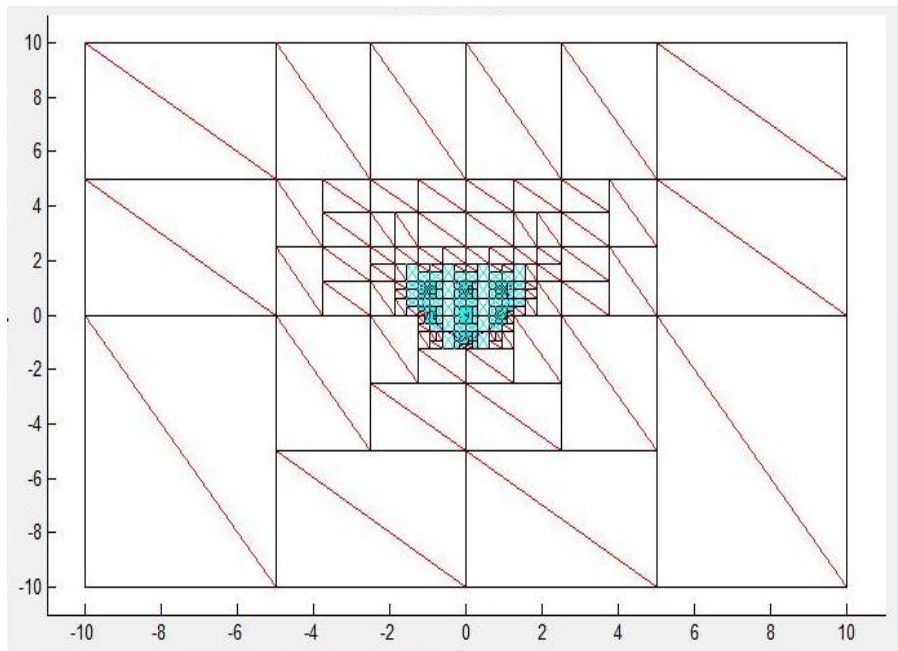
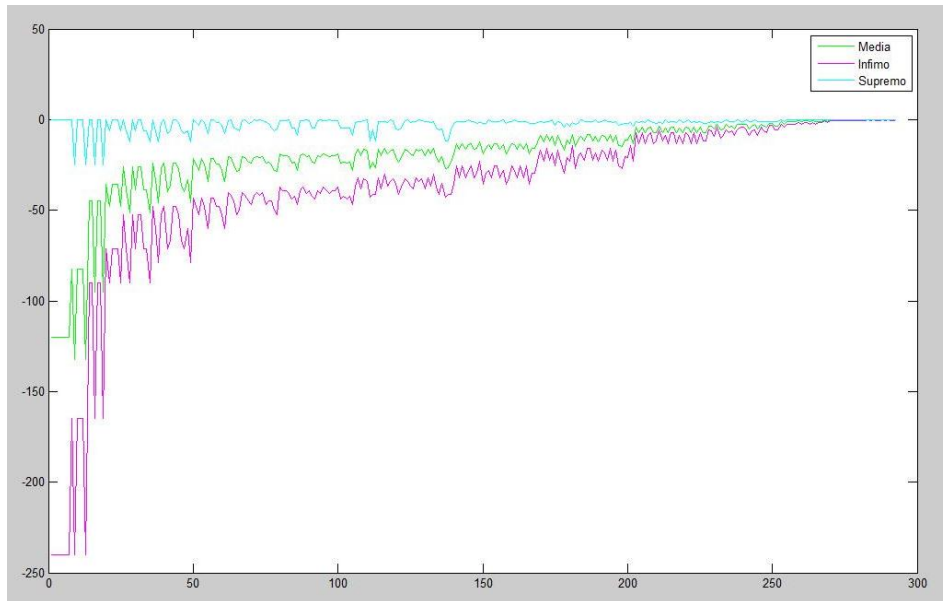


Figura 37. Valores estadísticos medios para F7



Una vez realizado el estudio correspondiente a la comparación de los métodos de optimización por enjambre de partículas (PSO) y el método de optimización por análisis de intervalos, es necesario nombrar algunas aplicaciones en las cuales sea indicado el uso de cada técnica. Como se pudo observar de los resultados obtenidos, el método de PSO se puede implementar en sistemas donde se requiera una respuesta buena pero no tan precisa con un tiempo de respuesta rápida, caso contrario ocurre con el método de optimización por análisis de intervalos, el cual puede ser implementado en sistemas de respuesta lenta pero con una solución rigurosa.

Un ejemplo de una aplicación del método de PSO es la optimización de la confiabilidad en redes de distribución de potencia [42], que ha sido planteada como la minimización de los costos totales de la confiabilidad. Estos costos son la suma de los costos de confiabilidad correspondientes a los costos anuales de operación y mantenimiento y los costos anualizados de inversión. Es así que, mediante la determinación del tipo, cantidad, y ubicación del equipo de protección y maniobra, los costos totales de confiabilidad son minimizados. Esta aplicación como se puede observar, no requiere una precisión rigurosa para que el proceso pueda ser optimizado, pero si requiere de respuestas rápidas en la generación de los datos. Otra aplicación importante que puede ser implementada mediante PSO es una metodología para mejorar la ingeniería de producto/proceso buscando la mejora en aspectos de costos y tiempo de respuesta para conseguir y mantener una posición competitiva en el mercado.

Un ejemplo de una aplicación del método de análisis de intervalos en la ingeniería se ve en el control de procesos químicos tales como la optimización de la detoxificación de aguas contaminadas con plaguicidas en el proceso de oxidación avanzada (AOPs). Este proceso es un proceso que no requiere una respuesta rápida pero si muy precisa.

Otro ejemplo en particular donde pueda ser empleado el método de análisis de intervalos es un procedimiento de optimización de la temperatura de agua helada en sistemas de climatización centralizados a flujo constante, con el objetivo de reducir el consumo de energía bajo determinadas condiciones climatológicas.

CONCLUSIONES

Los objetivos propuestos para el desarrollo de este proyecto de grado fueron cumplidos a cabalidad dejando como prueba y referencia la estadística mostrada al final del mismo.

Mediante la respectiva comparación de los resultados experimentales obtenidos al evaluar unas funciones de prueba sujetas a ciertas restricciones, se evaluó el desempeño de cada método planteado al realizar variaciones en ciertos parámetros.

Se verificó que tanto PSO como la técnica de análisis de intervalos estudiada, tienen el potencial de ser utilizados en la optimización de diversas funciones objetivo no-lineales sujetas a restricciones de igualdad y/o desigualdad. El beneficio de estos métodos radica en poder obtener resultados eficientes debido a la convergencia favorable de los mismos en condiciones sujetas a variaciones de sus parámetros. Aunque en el caso de PSO se puede tener una respuesta acertada en un menor tiempo de cómputo, en otros casos el algoritmo no encuentra el óptimo pero si la solución aproximada teniéndose esta como valor de referencia para la solución del problema. Caso contrario ocurre en la técnica de análisis de intervalos, que al ser un método determinístico, su respuesta siempre converge al valor exacto, pero con la limitación del tiempo de computación que en algunas aplicaciones prácticas puede llegar a ser una desventaja enorme, ya que en la actualidad la optimización global despierta gran interés como sistema de paso eficiente hacia nuevas tecnologías.

De los resultados experimentales de PSO, se puede deducir que los valores de ciertos parámetros afectan fuertemente su comportamiento, tales como la cantidad de población inicial, la cantidad de iteraciones y el factor de inercia y de estrechamiento, lo que puede generar mediante una buena aplicación, reducción en los costos computacionales y generación de sistemas más eficientes en general. Esto sería ideal si se requiere una solución razonable en costos y recursos ya que PSO genera en un número considerable de casos una respuesta aproximada que puede satisfacer las necesidades de optimización y contribuir a solucionar problemas de no-linealidades presentes en la mayoría de las aplicaciones de la ingeniería y las ciencias. Claro está que si lo que se requiere es una respuesta totalmente acertada, sin importarnos los costos requeridos moderados, entonces el empleo del método determinístico de análisis de intervalos sería la solución apropiada a los problemas objetivo planteados en el desarrollo de este trabajo de grado.

Se encontró que PSO con restricciones es potencialmente más rápido en la mayoría de los problemas estudiados frente al método de análisis de intervalos con restricciones. Sin embargo, la optimización entregada para las funciones F1 a

F9 con sus respectivas restricciones por el método de análisis de intervalos, ofreció mejor calidad de solución que el método de PSO con restricciones ya que generó un intervalo solución para la tolerancia indicada, con una precisión deseada. Así este método sería idóneo si se le implementaran algunas mejoras para reducir costos en tiempo computacional.

Cabe notar, que el método por particiones propuesto por K. Ichida se implementó para una, dos y tres variables con restricciones y fue implementado por los autores de este trabajo de grado utilizando INTLAB. Este método fue muy lento para algunas funciones debido a que las subregiones reservadas se incrementaban al aproximarse al intervalo solución y así se determinó que el criterio de descarte de subregiones no fue suficiente para eliminar la mayoría de las subregiones que no contenían la solución, lo cual ocasionó el incremento exagerado en el tiempo de cómputo.

Se concluye finalmente que el método de análisis de intervalos ofrece soluciones supremamente rigurosas a todos los problemas estudiados, comparado con PSO con restricciones, pero con un tiempo de cómputo que en algunos casos puede ser no deseado. Para ello surge la necesidad de implementar formas de reducir las subregiones generadas tales como el método de intervalos de Newton, que generaría un beneficio en la reducción de costos y mayor eficiencia.

TRABAJO FUTURO

El presente trabajo deja algunas propuestas interesantes para trabajos futuros tales como el hecho de expandir la aplicación del método de análisis de intervalos a problemas objetivo de más de 3 variables con restricciones de igualdad y/ o desigualdad. Para ello se dejan los principios del algoritmo y se espera que se realice en futuros proyectos de este tipo una ampliación de las aplicaciones de estos métodos.

Otro punto importante el cual se dejó como resultado de la ejecución del proyecto de grado es el hecho de poder encontrar un equilibrio entre la formulación de los dos métodos.

Para esto se requiere hacer una implementación de PSO que tenga incluido un paso del algoritmo de intervalos y así mediante este método se reduzca el espacio factible a un rango limitado en el cual PSO pueda encontrar la respuesta logrando obtener las ventajas de rapidez y precisión de los métodos de PSO y análisis de intervalos respectivamente y así conseguir una rápida convergencia y rigurosidad en las soluciones.

Como trabajo futuro se debe poder obtener lo mejor de ambos métodos y realizar una vinculación entre las características de desempeño, eficiencia y precisión de cada uno para así lograr un mecanismo de optimización global con restricciones muy poderoso en aplicaciones en diversas ramas de la ingeniería y ciencias en general.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Pardalos, P. M. and Horst, R. "Handbook of Global Optimization.", Journal of Applied Mathematics and Mechanics , vol. 77, 1995.
- [2] Floudas, C.A. Deterministic Global Optimization: Theory, Algorithms and Applications. Kluwer, Dordrecht, Holanda: kluwer academy publishers, 1999.
- [3] Tawarmalani, M and Sahinidis, N. Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications. Dordrecht: Kluwer academy publishers, 2002.
- [4] Kirkpatrick, S. Gelatt, C. D. and Vecchi, M. P. "Optimization by simulated Annealing ", Science, vol. 220, no. 4598, pp. 671-680, 1983.
- [5] Glover, F. "Heuristics for Integer Programming Using Surrogate Constraints", Decision Science, pp. 156-166, 1977.
- [6] Glover, F. "Future Paths for Integer Programming and Links to Artificial", Computers and Operations Research, pp. 533-549, 1986.
- [7] Hansen, P. and Mladenovic, N. "Variable neighborhood search", en Handbook of Metaheuristics. Boston, USA: Kluwer Academy Publishers, 2002, pp. 145-184.
- [8] Lourenco, HR. Martin, O. Stutzle, T. "Iterated local search", en Handbook of Metaheuristics. Boston: USA, 2003, pp. 321-353.
- [9] Back, T. Algoritmos evolutivos en teoría y práctica: Estrategias de la evolución, programación evolutiva, algoritmos genéticos.: Oxford University press, 1996.
- [10] Fogel, LJ.Owens,AJ.Walsh,MJ. Artificial Intelligence Through simulated evolution. New york, 1966.
- [11] Fogel, D.B. Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. Piscataway,New Jersey: IEEE Press, 1995.
- [12] Dorigo, M.Stutzle,T. Ant Colony Optimization.: MIT press, 2004.

- [13] Dorigo, M. Maniezzo, V. The ant system: Optimization by a colony cooperating agents., 1996.
- [14] Guerra Alvarez, N. Crawford Labrin, B. Optimizacion de funciones a traves de optimizacion por enjambre de particulas y algoritmos geneticos.
- [15] Beyer, HG. Schwefel, HP. "Evolution strategies-A comprehensive introduction", Natural computing, no. 1, pp. 3-52, 2002.
- [16] Rechenberg, I. "Evolutionstrategie: Optimierung Technischer Systeme Prinzipien der Biologischen Evolution", 1973.
- [17] Rechenberg, I. "Evolutionsstrategie '94".
- [18] Goldberg, D.E. Genetic Algorithms in search, optimization and machine learning. Boston, MA, USA: Addison-Wesley Longman Publishing Co, 1989.
- [19] Vose, D. L. Battle and Michael D. "Isomorphisms of genetic algorithms", Artificial Intelligence, vol. 60, pp. 155-165, 1993.
- [20] Zhang, Qijuan Chen and Jingyu Liu Dahai. "An Improved Hybrid Genetic Algorithm for Solving Multi-modal Function Global Optimization Problem", en Proceedings of the IEEE International Conference on Automation and Logistics, Jinan, China., 2007.
- [21] Holland, J.H. Adaptation in Natural and Artificial Systems.: The University of Michigan Press, 1975.
- [22] Kennedy, J. Eberhart, R. "Particle swarm optimization", en IEEE international conference on Neural Networks, Piscataway, NJ, 1995, pp. 1942-1948.
- [23] Reeves, W.T. "Particle systems -A technique for modeling a class of fuzzy objects", ACM Transactions on Graphics, pp. 91-108, 1983.
- [24] Reynolds, C.W. "Flocks, herds and schools: A distributed behavioral model", ACM Computers Graphics, pp. 25-34, 1987.
- [25] Heppner, F and Grenander, U. "A stochastic nonlinear model for coordinated bird flocks", en The ubiquity of chaos, Saul Krasner, Ed. Washington, DC: AAAS Publications, 1990, pp. 233-238.

- [26] Nowak, A. Szamrej, J. and Latane, B. "From private attitude to public opinion: A dynamic theory of social impact.", *Psychological Review*, pp. 362-376, 1990.
- [27] Kennedy, J. "Swarm Intelligence", en *Handbook of Nature-Inspired and Innovative Computing: Integrating Classical Models with Emerging Technologies*. Secaucus, NJ, USA, 2006, pp. 187-219.
- [28] Kennedy, J. Eberhart, R. "Particle swarm optimization.", en *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, NJ, 1995, pp. 1942-1948.
- [29] Engelbrecht, A.P. *Fundamentals of computational Swarm Intelligence*. UK, 2005.
- [30] Parsopoulos, K.E. and Vrahatis, M.N. "A unified particle swarm optimization scheme", en *Computational Methods in sciences and Engineering*, 2004, pp. 868-873.
- [31] Clerc, M. "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization", *Evolutionary Computation*, vol. 3, pp. 1951-1957, 1999.
- [32] Liang, J. J. and Suganthan, P.N. "Dynamic multi-Swarm particle optimizer with a novel constrained handling mechanism", en *Proceedings of the IEEE Congress on Evolutionary Computation*, Vancouver, 2006, pp. 316-323.
- [33] Parsopoulos, K.E. and Vrahatis, M.N. "Unified particle swarm optimization for solving constrained engineering optimization problems", *Computer Science*, vol. 3612, pp. 582-591, 2005.
- [34] Kong, P. and Li, X. Tian M., "Novel particle swarm optimization for constrained optimization problems", *Computer Science*, vol. 3809, pp. 1305-1310, 2005.
- [35] Coello-Coello, C. Cagnina, C. A. and Esquivel, L. C. "A particle swarm optimizer for constrained numerical", *Computer Science*, vol. 4193, pp. 910-919, 2006.
- [36] Chen, W. and Lu, Q. "Dynamic-objective particle swarm optimization for constrained optimization problems", *Journal de combinatorial optimization*, pp. 409-419, 2006.

- [37] Krohling, L. and dos Santos Coelho, R. A. "Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems", IEEE Transactions on Systems, Man and Cybernetics, pp. 1407-1416, 2006.
- [38] Wu, E. Prempain, S. He and Q.H. "An improved particle swarm optimizer for mechanical design optimization problems ", Engineer optimization, vol. 36, pp. 585-605, 2004.
- [39] Poli, R. "Analysis of the publications on the applications of particle swarm optimization", Journal of artificial Evolution and applications, 2008.
- [40] Min Li, Xianjia Wang. Weibing Liu, An Improved Particle Swarm Optimization Algorithm for Optimal Power Flow.. China.
- [41] Wang, Qing-ran. "An improved particle swarm optimization algorithm and application in available transfer capability", North China Electric Power University, Beijing,.
- [42] Midence, D. y Vargas, A. "Estudio Comparativo de Algoritmos de Computación Evolutiva en la Optimización de la Confiabilidad en Redes de Distribución de Potencia", en Decimo tercer encuentro regional Iberoamericano de Cigre, Puerto Iguazu, Argentina, 2009.
- [43] Mori, H. y Maeda, Y. "A hybrid method of EPSO and TS for FACTS optimal allocation in power systems", en ICSMC'06 IEEE International Conference on Systems, Man and Cybernetics., 2006, pp. 1831-1836.
- [44] Dwyer, P.S. "Computation with approximate numbers", en Linear computation. New York, 1951, pp. 11-34.
- [45] Sunaga T. "Theory of interval algebra and its application to numerical analysis", RAAG Memoirs, pp. 29-46., 1958.
- [46] Moore, R. E. and Yang, C. T. "Interval analysis I", Technical Report Space Div LMSD285875, 1959.
- [47] Moore, R. E., Interval analysis. USA: Prentice-Hall, 1966.
- [48] Hansen. E. R., "On solving systems of equations using interval arithmetic",

Math. of Comp. , vol. 22, pp. 374-384, 1968.

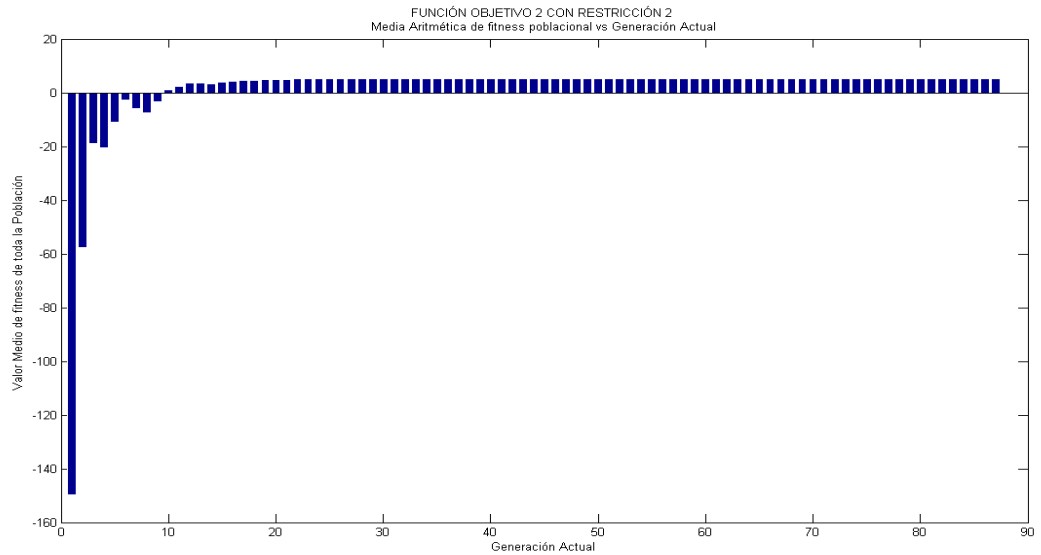
- [49] Moore, R. E. Methods and Applications of Interval Analysis. Philadelphia, 1979.
- [50] Nickel, K. Interval Mathematics. New york: Springer, 1979.
- [51] Markov, S. "Interval Differential Equations", en Interval Mathematics., 1980, pp. 145-164.
- [52] Alefeld, G. and J. Herzberger . Introduction to Interval Computations. New york, USA: Academic Press, 1983.
- [53] Hansen, E.R. "On solving systems of equations using interval arithmetic", en Mathematics of computation., 1968, pp. 374-384.
- [54] Hansen, E. "Sharpness in Interval Computations", Computing, vol. 3, no. 1, pp. 17-29, 1997.
- [55] Hansen, Eldon and Walster,G. William. Global optimization using interval analysis.: CRC Press, 2003.
- [56] Hansen, E. "Sharpening Interval Computations. Reliable ", Computing, pp. 21-34, 2006.
- [57] Hansen, E. "Multidimensional Interval Newton Method.", Computing, vol. 12, no. 4, pp. 253-272, 2006.
- [58] Moore, E. R. Kearfott,R y Cloud, MJ. Introduction to interval analysis.: SIAM, 2009.
- [59] Neumaier, A. "Interval Methods for Systems of Equations", en Encyclopedia of Mathematics and its Applications. Cambridge, Inglaterra: Cambridge University Press, 1990.
- [60] Neumaier, A. Introduction to numerical analysis. Cambridge, UK: Cambridge University Press, 2001.
- [61] Hansen, E.R. and Greenberg,R.I. "An interval newton method".

- [62] Hansen, E.R. "Global optimization using interval analysis: The one-dimensional case ", *Journal of Optimization Theory and Applications*, vol. 29, no. 3, pp. 331-344, 1979.
- [63] Hansen, E.R. "Global optimization using interval analysis — the multi-dimensional case ", *Numerische Mathematik* , vol. 34, no. 3, 1980.
- [64] Ichida, K. and Fujii, Y. "An interval arithmetic method for global optimization", *Computing*, vol. 23, no. 1, pp. 85-97, 1979.
- [65] Jansson, C. "A global optimization method using interval arithmetic", en *Third International IMACS-GAMM symposium on computer arithmetic and scientific computing*, Amsterdam, 1992, pp. 259-268.
- [66] S.M. Rump. INTLAB - INTerval LABoratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77-104. Kluwer Academic Publishers, Dordrecht, 1999
- [67] Hargreaves, G. I. "Interval Analysis in MATLAB", University of Manchester, Manchester, Numerical Analysis Report 2002.
- [68] Gómez González, M. "Técnicas metaheurísticas. algoritmos basados en nubes de partículas", en tesis doctoral: sistema de generación eléctrica con pila de combustible de óxido sólido alimentado con residuos forestales y su optimización mediante algoritmos basados en nubes de partículas, 2008, cap. 3, pp. 51-55.
- [69] Ichida, K. "Constrained optimization using interval analysis", *Computers and Engineering*, vol. 31, no. 3-4, pp. 933-937, 1995.
- [70] Sam. Another Particle Swarm Toolbox. Disponible en: <http://www.mathworks.com/matlabcentral/fileexchange/25986-another-particle-swarm-toolbox>. Consultada: 20 Marzo 2010.

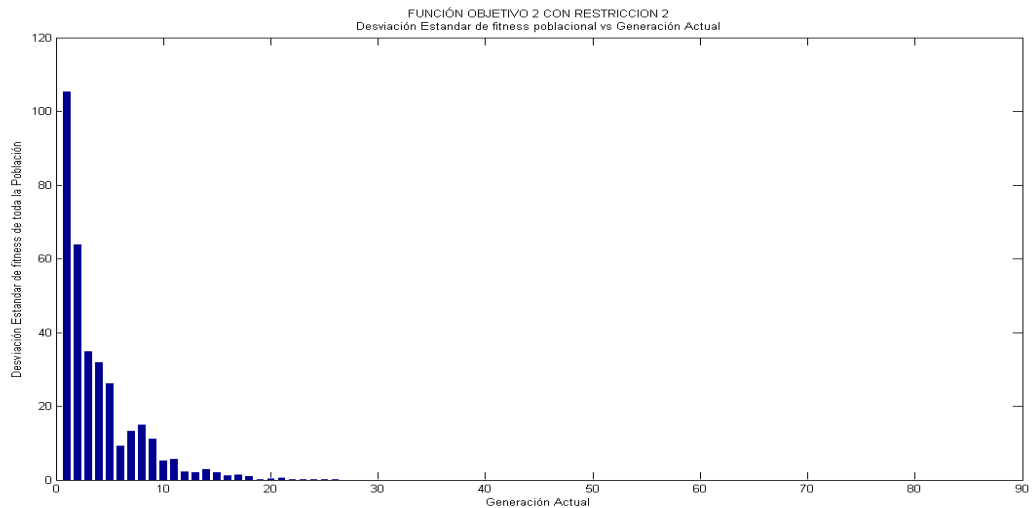
ANEXOS

ANEXO A. Gráficas de valores medios, desviación estándar y error medio cuadrático de las funciones objetivo.

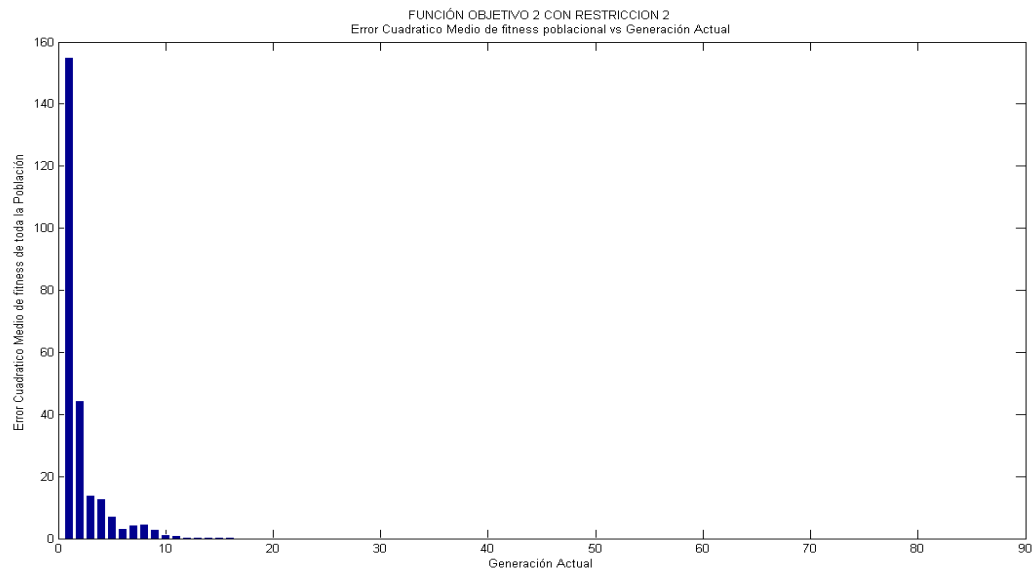
- **Media aritmética de fitness vs. Generación actual para F2.**



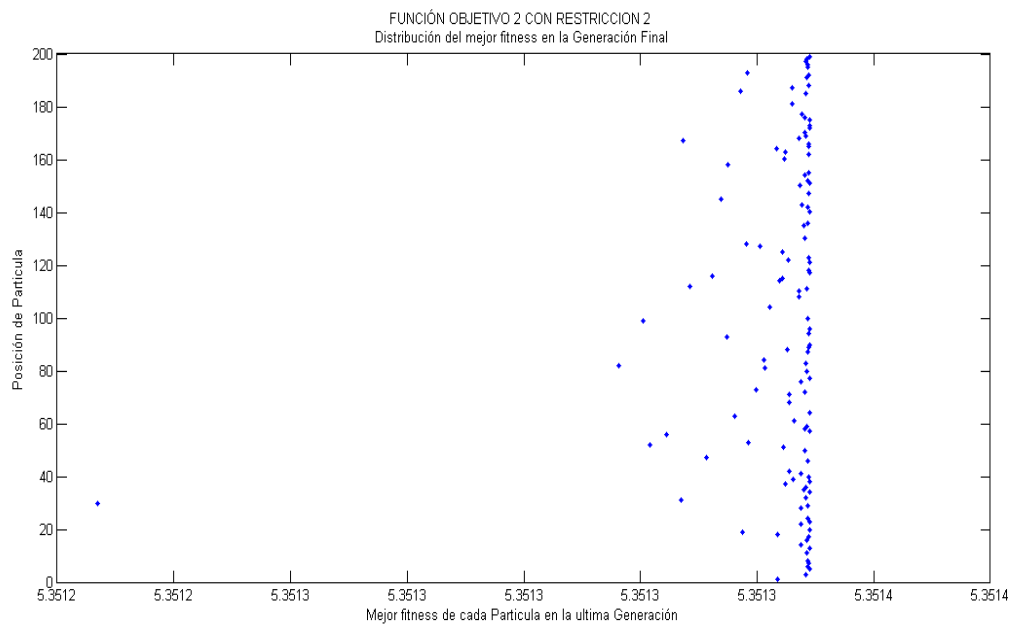
- **Desviación estandar de fitness vs. Generación actual para F2.**



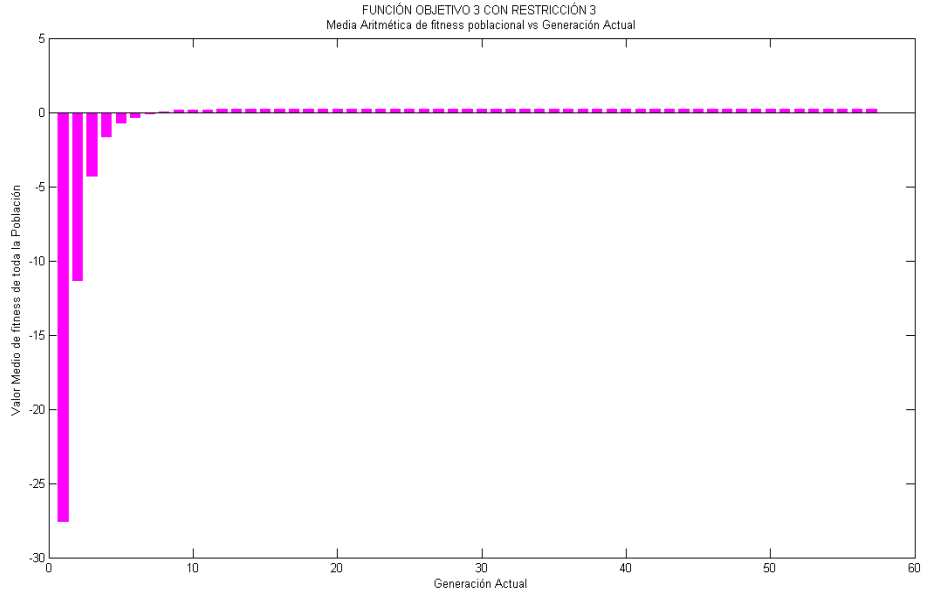
- **Error medio cuadrático de fitness vs. Generación actual para F2**



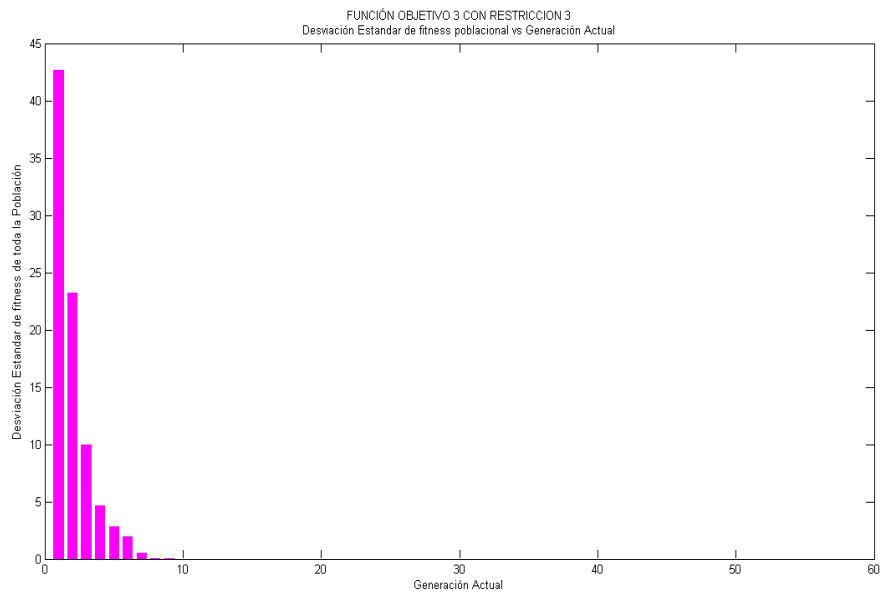
- **Distribución del mejor fitness en la generación actual.**



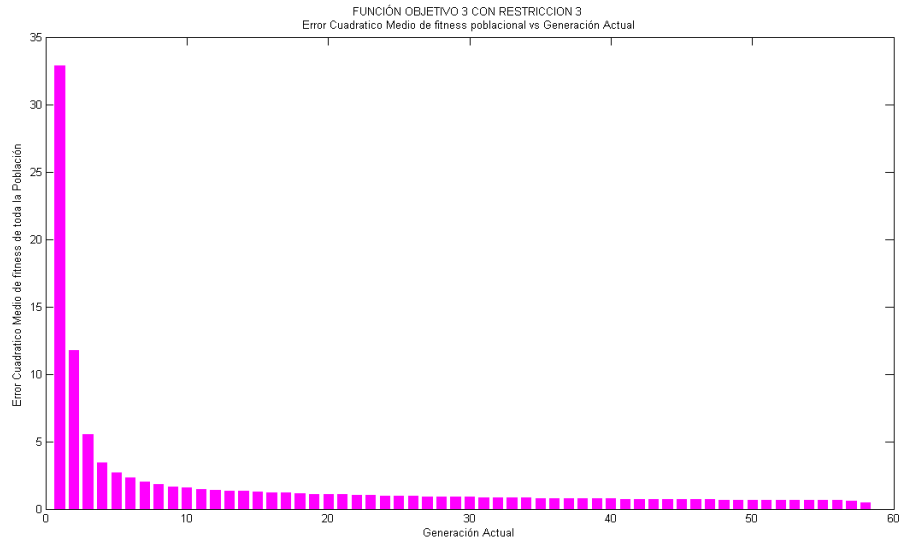
- **Media aritmética de fitness vs. Generación actual para F3.**



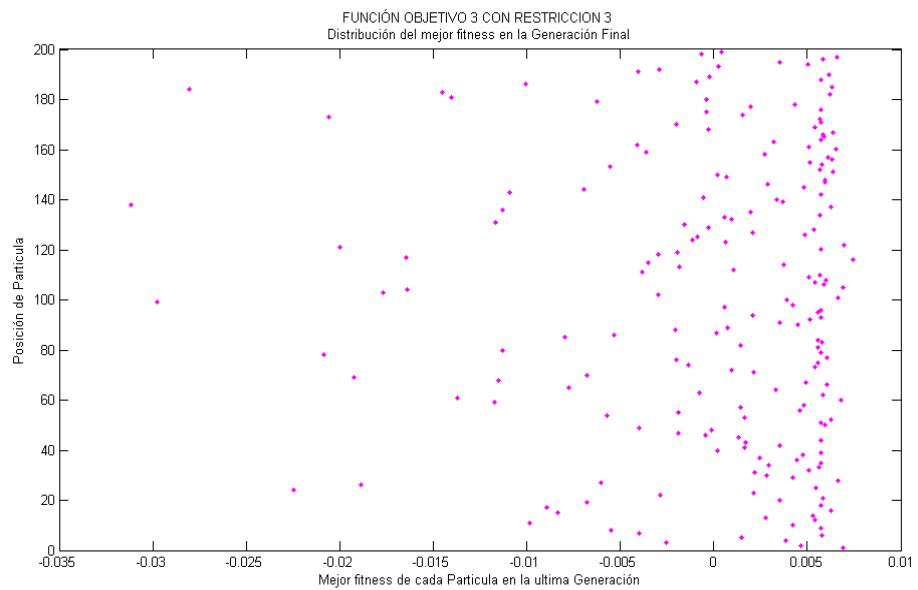
- **Desviación estandar de fitness vs. Generación actual para F3.**



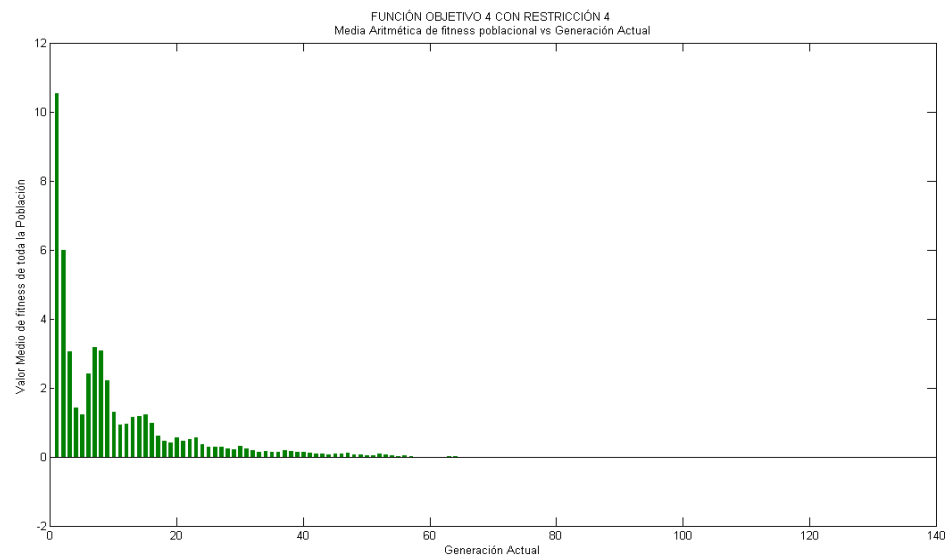
- **Error medio cuadrático de fitness vs. Generación actual para F3**



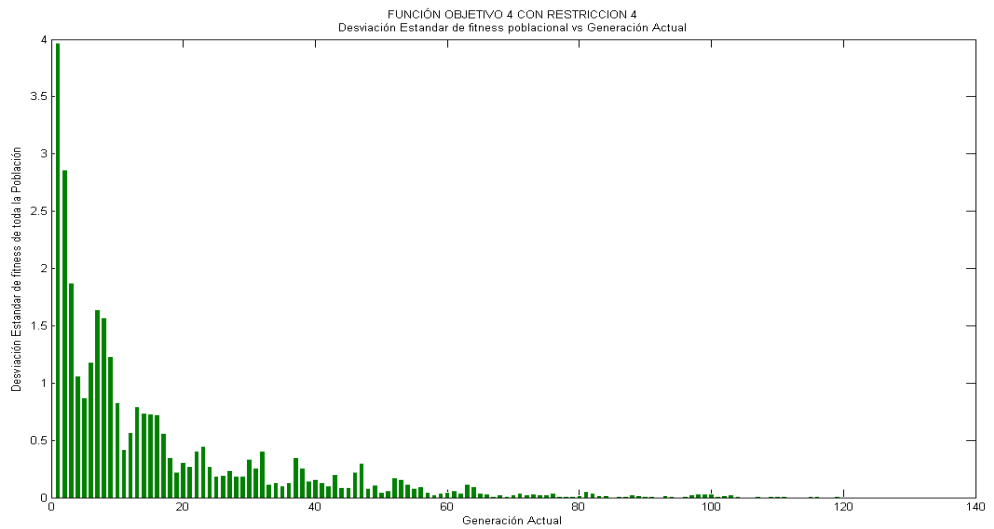
- **Distribución del mejor fitness en la generación actual para F3**



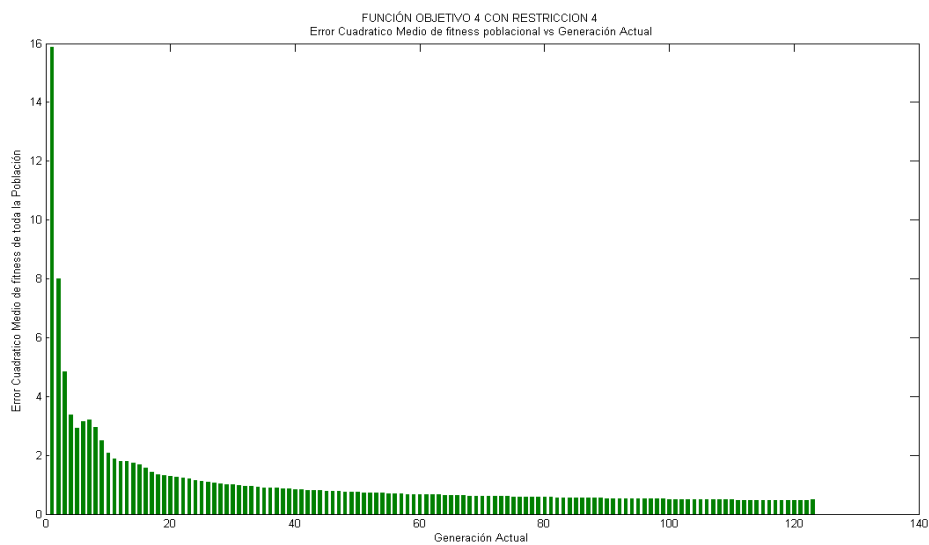
- **Media aritmética de fitness vs. Generación actual para F4.**



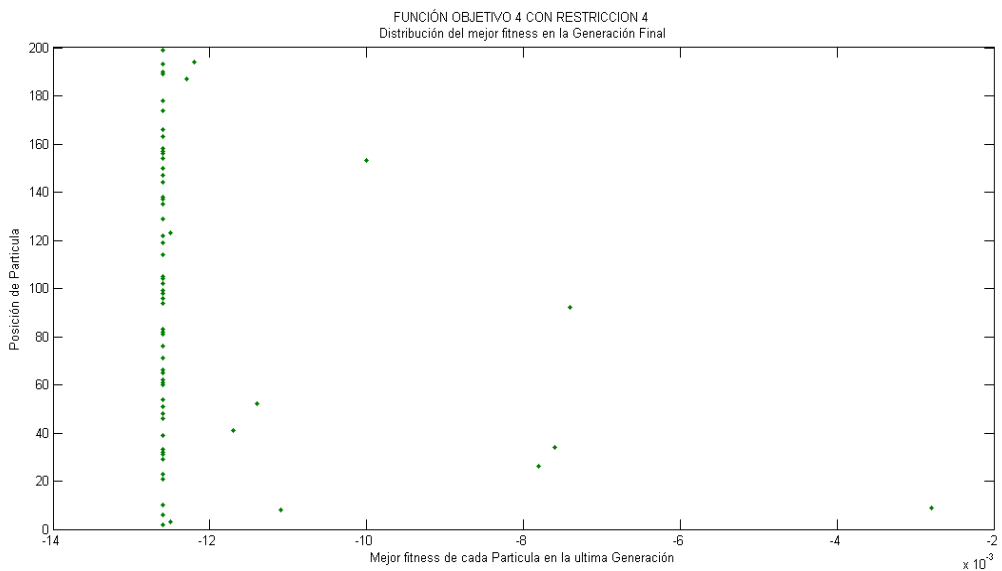
- **Desviación estandar de fitness vs. Generación actual para F4.**



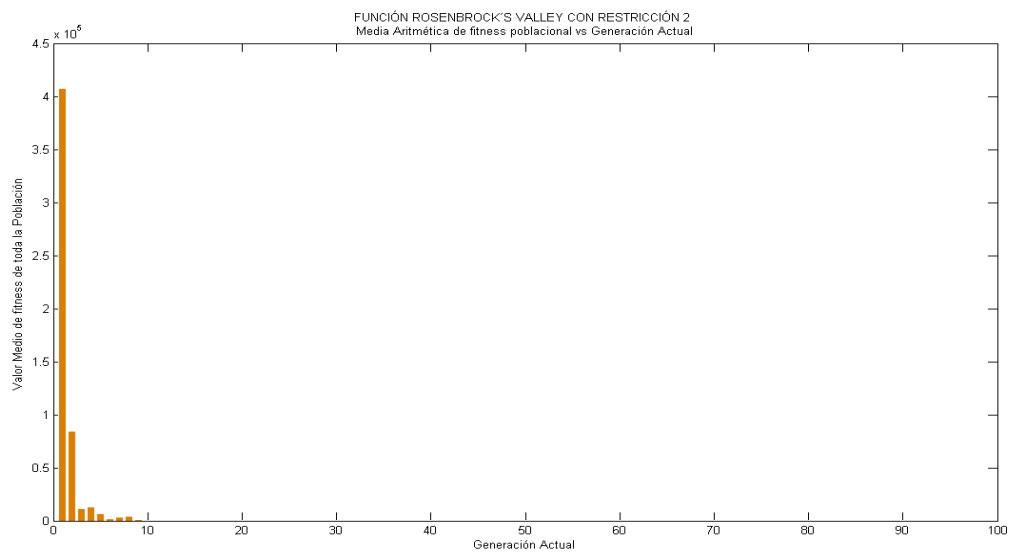
- **Error medio cuadrático de fitness vs. Generación actual para F4**



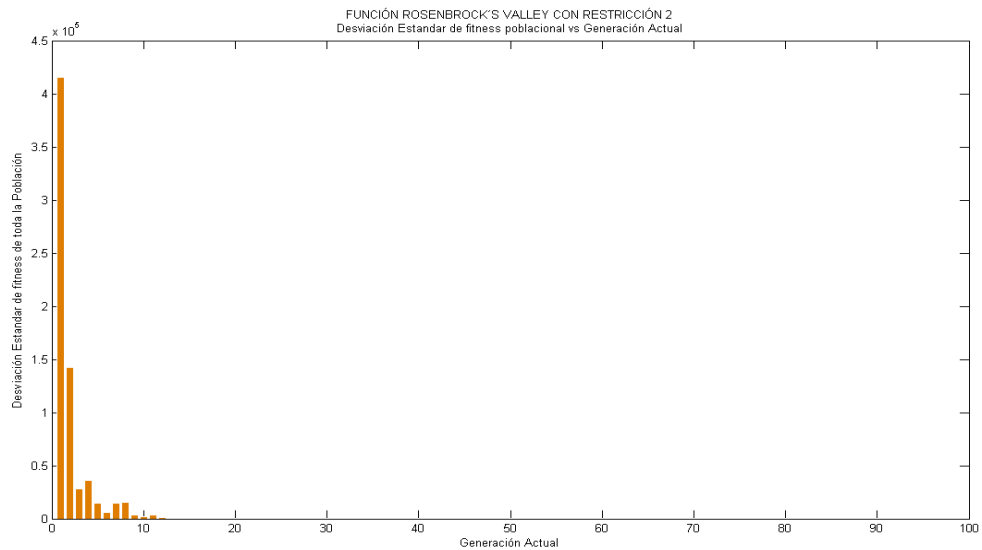
- **Distribución del mejor fitness en la generación actual para F4.**



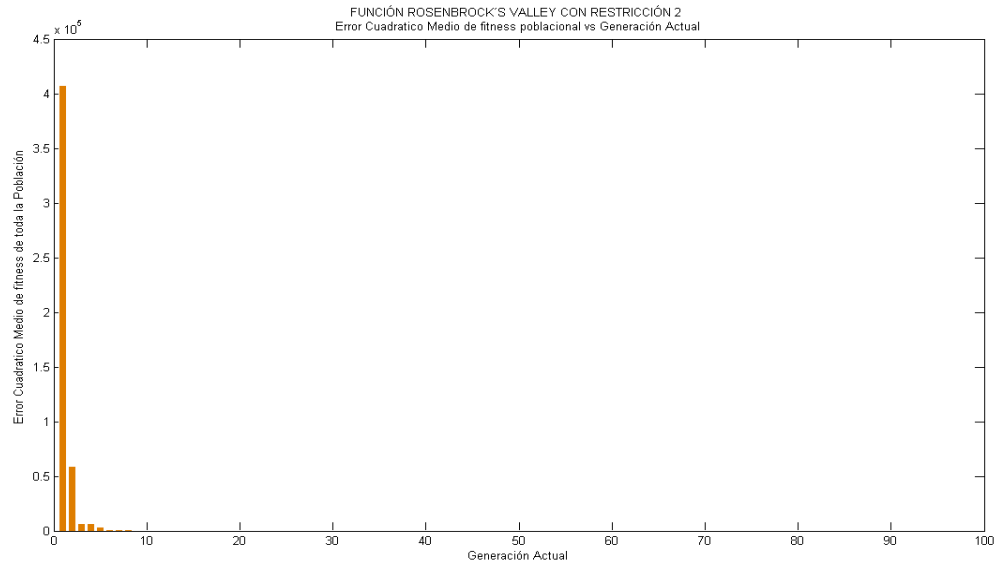
- **Media aritmética de fitness vs. Generación actual para F5.**



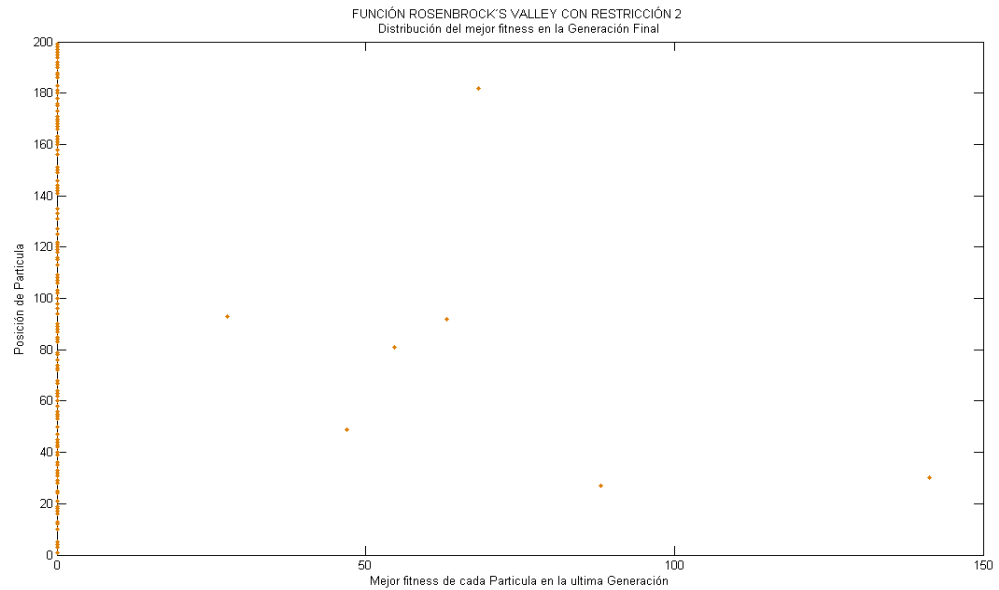
- **Desviación estandar de fitness vs. Generación actual para F5.**



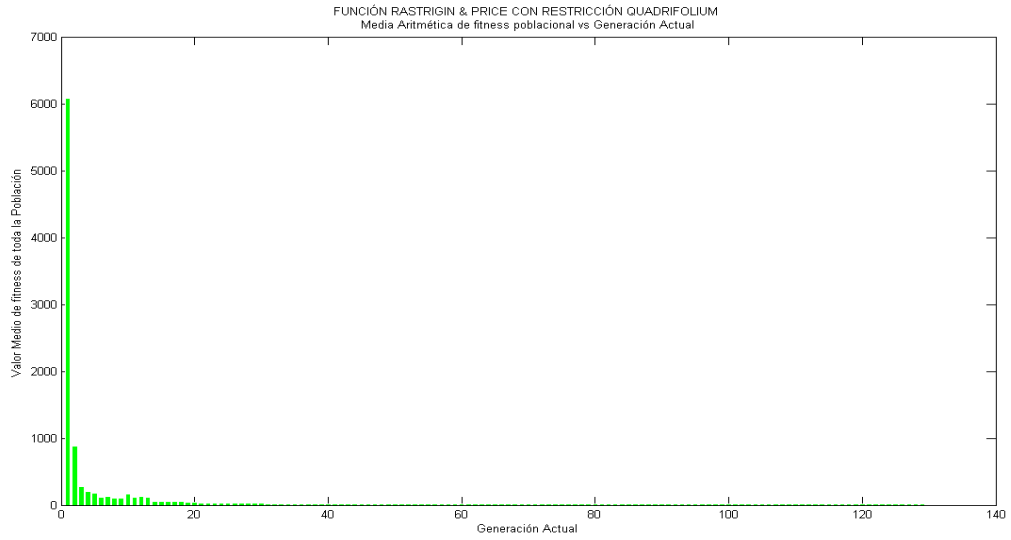
- **Error medio cuadrático de fitness vs. Generación actual para F5.**



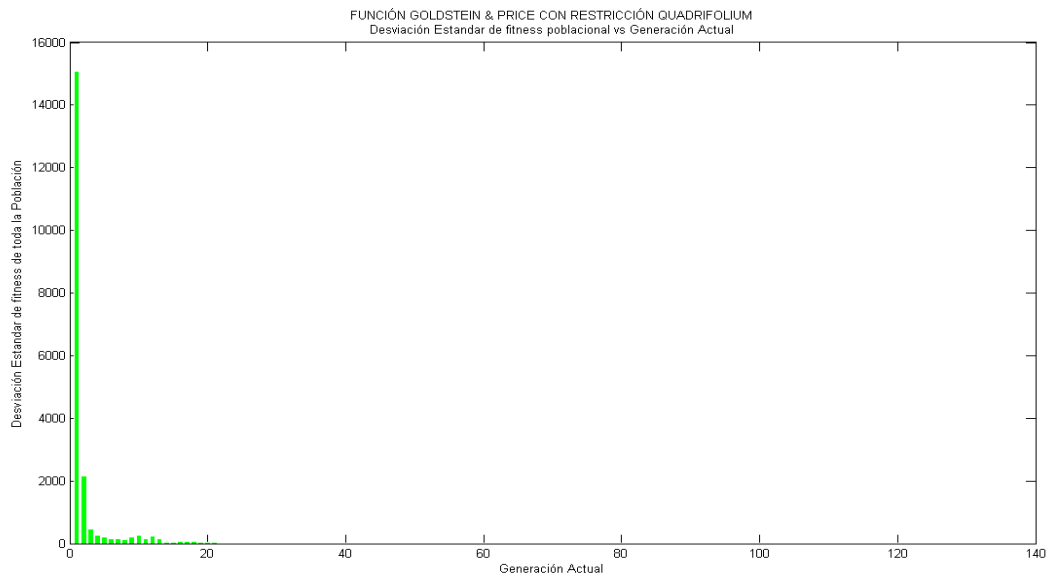
- **Distribución del mejor fitness en la generación actual para F5.**



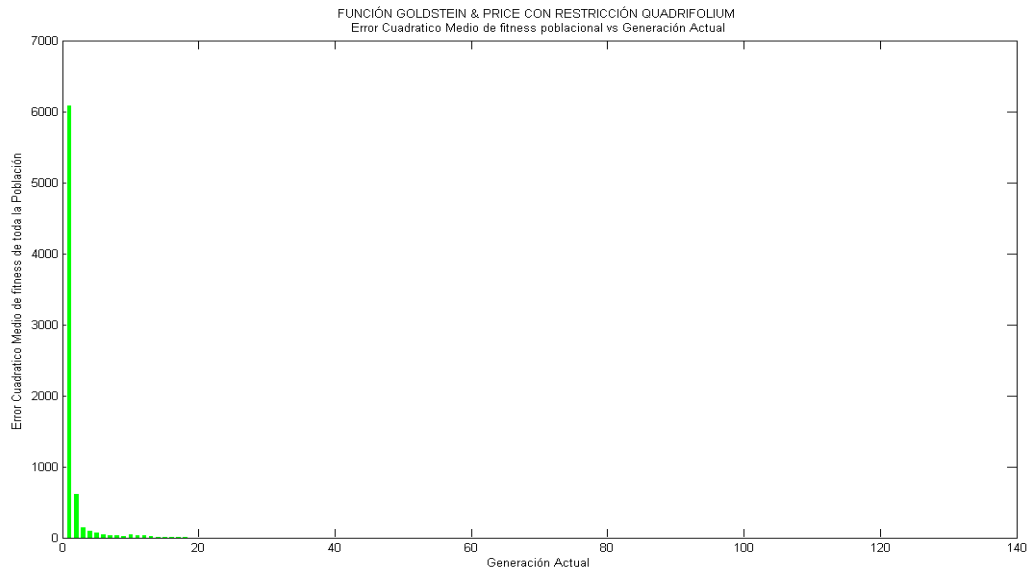
- **Media aritmética de fitness vs. Generación actual para F6**



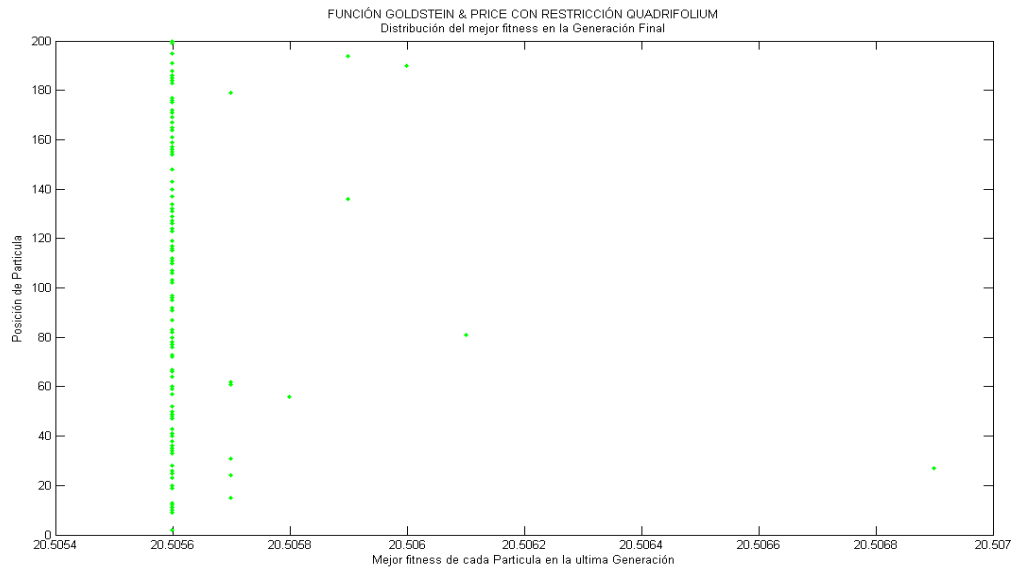
- **Desviación estandar de fitness vs. Generación actual para F6.**



- **Error medio cuadrático de fitness vs. Generación actual para F6.**



- **Distribución del mejor fitness en la generación actual para F6.**



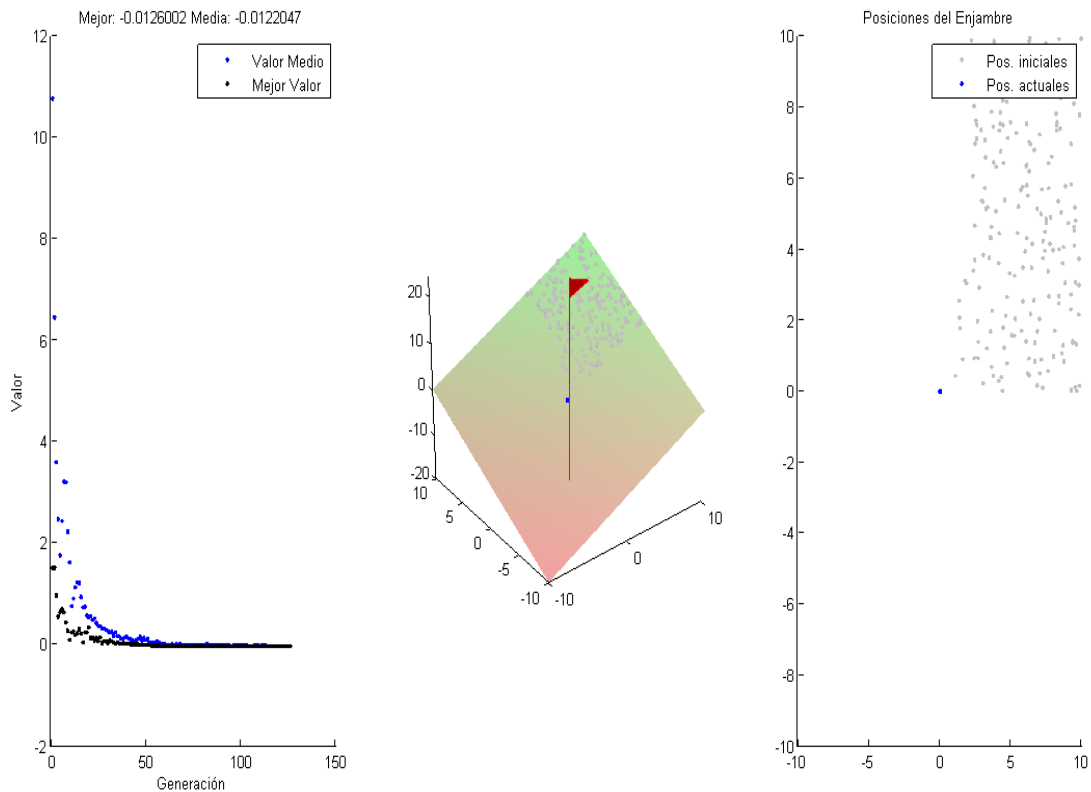
ANEXO B. GUIA DE USO INTERFACES

- **Guía de uso interfaz algoritmo PSO con restricciones.**

Para la solución de los problemas de optimización planteados en este documento mediante el método PSO, se empleó un algoritmo para optimización con restricciones disponible en:

<http://www.mathworks.com/matlabcentral/fileexchange/25986-another-particle-swarm-toolbox>. Consultada: 20 Marzo 2010

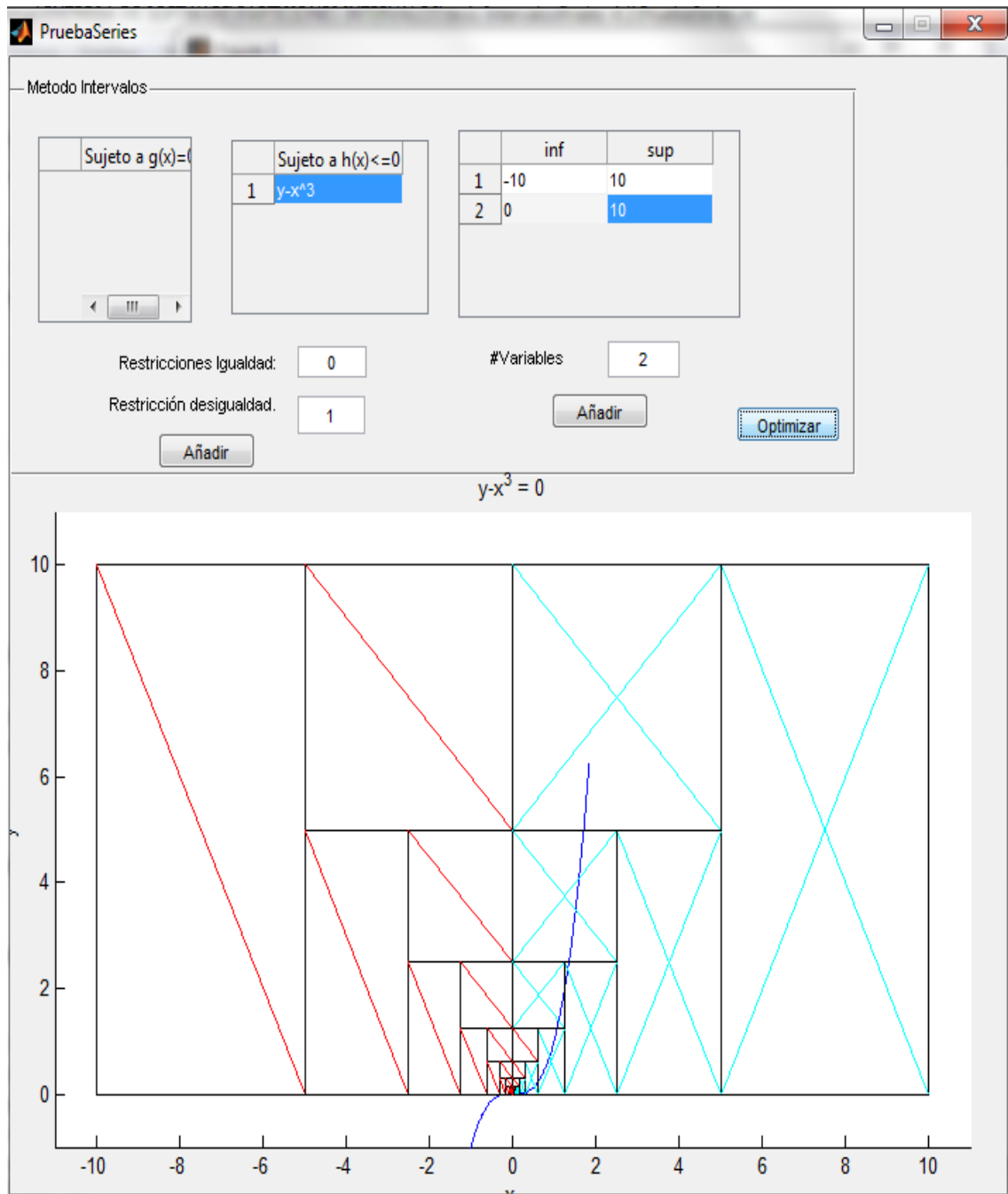
Las gráficas obtenidas al utilizar este algoritmo fueron las siguientes:



Las cuales pueden ser obtenidas al habilitar en la línea 52 del archivo *psodemo.m* la opción `problem.options.PlotFcns = {@psoplotbestf,@psoplotswarmsurf,@psoplotswarm}` y luego correr el archivo *psodemo.m*.

- **Guía de uso interfaz método de análisis de intervalos**

La interfaz desarrollada para la optimización de funciones no lineales sujetas a restricciones de igualdad y/o desigualdad se implementó en MATLAB y fue desarrollada por los autores del proyecto de grado. Esta interfaz presenta el siguiente aspecto:



La interfaz se puede visualizar al correr el archivo *pruebaseries.m* perteneciente al algoritmo implementado.

Una vez generada la interfaz, se deben introducir las restricciones de igualdad y desigualdad, el número de variables y los límites de las restricciones. Las restricciones deben estar en términos de variables tales como $x, y, \dots etc$ y no en términos de vectores como están definidos los archivos *.m*.

Una vez introducidos estos valores, se hace clic en el botón optimizar y la interfaz redirecciona al archivo principal *pruebaseries.m* donde el proceso de cálculo de los valores óptimos es realizado. Una vez que se calculan los valores óptimos, la interfaz gráfica las particiones y el descarte de las regiones realizada por el método de intervalos, pero solo para el caso de 2 variables. Cabe aclarar que aunque la interfaz solo grafica la respuesta para dos variables, siempre debe ser utilizada la interfaz como método de solución del problema de optimización al introducir los parámetros de entrada nombrados anteriormente.

En el archivo *pruebaseries.m* se deben modificar algunos parámetros dependiendo de la cantidad de variables con que se desee trabajar (en el caso particular de este documento, el máximo de variables fue tres). Cada variable tiene un criterio de parada bajo el cual calcula los valores óptimos. Este criterio de parada es una tolerancia definida en el algoritmo. Para cada variable se define un error E que es la diferencia entre el valor supremo y el infimo de cada posible solución. Cuando todos los errores E estén por debajo de la tolerancia el algoritmo se detiene. Así para cada variable en el algoritmo *pruebaseries.m* deben estar activos los valores de E , dependiendo de la cantidad de variables que manejen las funciones objetivo.