

**EVALUACION DEL DESEMPEÑO DE ALGORITMOS DE EVOLUCION
PARAMETRICA PARA OPTIMIZACION**

CAMILO EDUARDO CELIS MELO

CARLOS ANDRES MILLAN PARAMO

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERÍA CIVIL
BUCARAMANGA**

2009

**EVALUACION DEL DESEMPEÑO DE ALGORITMOS DE EVOLUCION
PARAMETRICA PARA OPTIMIZACION**

**CAMILO EDUARDO CELIS MELO
CARLOS ANDRES MILLAN PARAMO**

Trabajo de grado presentado como requisito para optar al título de
Ingeniero Civil

**Director:
PhD. Oscar J. Begambre C.**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERÍA CIVIL
BUCARAMANGA
2009**

AGRADECIMIENTOS

A nuestra Alma Mater y en especial a la Escuela de Ingeniería Civil, la cual por medio de su profesorado nos formo hasta alcanzar este logro.

A nuestro director Oscar Javier Begambre Carrillo, quien siempre creyó y oriento nuestro trabajo.

DEDICATORIA

A mi Dios por no dejarme desfallecer en la lucha por esta meta.

*A Juanfe y a Yuli porque este libro
esta escrito a costa del tiempo que debí
haber compartido con ellos.*

*A mis padres Saul y Maria Helena por ser
el apoyo incasable en el camino de la vida.*

*A mis hermanos Nathalia y Daniel por enseñarme
el significado de la amistad.*

A Carlos por ser el complemento necesario en este logro.

A mis amigos por creer en mí.

Camilo Celis.

DEDICATORIA

*A mi Señor, Jesús, quien me dio la fe, la fortaleza,
y la salud para terminar este trabajo.*

*A mis Padres, Euriel e Irma, ya que gracias
a ellos soy quien soy hoy en día.*

A mis hermanas, Claudia y Vanessa, por su apoyo incondicional.

*A Leydi quien me brindó su amor, su cariño,
su estímulo, y su apoyo constante.*

*A mi hija Sofia que con sus ojitos y cariño me dan la fuerza necesaria para estar de pie
y con la cabeza en alto para enfrentar cualquier situación por difícil que sea.*

A Camilo por ser el complemento necesario en este logro.

A mis Amigos, quienes se convirtieron en mi familia adoptiva en Bucaramanga.

Carlos Millán.

TABLA DE CONTENIDO

INTRODUCCION	1
1. ALGORITMOS DE OPTIMIZACION	3
1.1. INTRODUCCION	3
1.2. ALGORITMOS GENETICOS (AG)	8
1.2.1. Algoritmos Genéticos en Codificación Binaria	11
1.2.2. Representación, Selección, Cruce y Mutación	12
1.2.3. Algoritmos Genéticos en Codificación Real	14
1.2.4. Representación, Selección, Cruce y Mutación	15
1.3. OPTIMIZACION CON EMJAMBRE DE PARTICULAS (PSO)	16
1.3.1. Fundamentos del Movimiento de Partículas	19
1.3.2. Operador de Velocidad y Parámetros de Algoritmo	20
1.3.3. Analogías y Diferencias entre PSO y AG	21
1.4. SIMULATED ANNEALING (SA)	23
2. ALGORITMOS HIBRIDOS	27
2.1. INTRODUCCION	27
2.2. ESTADO DEL ARTE	28
2.2.1. Híbridos referenciados en la literatura internacional	28
2.3. ALGORITMOS HÍBRIDOS DE EVOLUCION PARAMETRICA	32
2.3.1. ALGORITMO HIBRIDO SAA-PSO	34
2.3.2. ALGORITMO HIBRIDO PSO-SAA	35
2.3.3. ALGORITMO HIBRIDO GA-SAA	36
3. EVALUACION DE LOS ALGORITMOS DE OPTIMIZACION Y ALGORITMOS DE EVOLUCION PARAMETRICA CON FUNCIONES DE LA LITERATURA INTERNACIONAL	38
3.3. DESCRIPCION DE ENSAYOS	39
3.4. ALGORITMOS GENETICOS (AG)	45
3.5. OPTIMIZACION CON EMJAMBRE DE PARTICULAS (PSO)	50
3.6. SIMULATED ANNEALING (SA)	55
3.7. ALGORITMO HIBRIDO SAA-PSO	60
3.8. ALGORITMO HIBRIDO PSO-SAA	64

3.9. ALGORITMO HIBRIDO GA-SAA	68
4. ANALISIS E INTERPRETACION DE DATOS	72
4.1. RESULTADOS	72
4.2. ANALISIS	78
CONCLUSIONES	80
RECOMENDACIONES	81
BIBLIOGRAFIA	82

LISTA DE FIGURAS

FIGURA 1: CLASIFICACION DE LOS METODOS DE OPTIMIZACION MAS RELEVANTES	5
FIGURA 2: DIAGRAMA DE FLUJO DE UN ALGORITMO GENETICO	11
FIGURA 3: DIAGRAMA DE FLUJO DE UN ALGORITMO PSO	18
FIGURA 4: PROCESO BASICO QUE SE SIGUE EN UN ALGORITMO SA	25
FIGURA 5: ESQUEMA DE UN ALGORITMO DE EVOLUCION PARAMETRICA	33
FIGURA 6: DIAGRAMA DE FLUJO SAA-PSO	35
FIGURA 7: DIAGRAMA DE FLUJO PSO-SAA	36
FIGURA 8: DIAGRAMA DE FLUJO GA-SAA	37
FIGURA 9: FUNCIÓN DE ROSENBROCK 2D	40
FIGURA 10: FUNCIÓN DE VENTER 2D	41
FIGURA 11: FUNCIÓN DE BROWN 20D	42
FIGURA 12: FUNCION DE SCHWEFEL 5D	43
FIGURA 13: FUNCIÓN DE N-DIMENSIONAL 5D	44

LISTA DE TABLAS

TABLA 1: FUNCIÓN DE ROSENBROCK 2D (AG)	45
TABLA 2: FUNCIÓN DE ROSENBROCK 50D (AG)	46
TABLA 3: FUNCIÓN DE VENTER 2D (AG)	46
TABLA 4: FUNCIÓN DE BROWN 20D (AG)	47
TABLA 5: FUNCIÓN DE SCHWEFEL 5D (AG)	47
TABLA 6: N-DIMENSIONAL TEST FUNCTION 5D (AG)	48
TABLA 7: TABLA RESUMEN AG	49
TABLA 8: FUNCIÓN DE ROSENBROCK 2D (PSO)	50
TABLA 9: FUNCIÓN DE ROSENBROCK 50D (PSO)	51
TABLA 10: FUNCIÓN DE VENTER 2D (PSO)	51
TABLA 11: FUNCIÓN DE BROWN 20D (PSO)	52
TABLA 12: FUNCIÓN DE SCHWEFEL 5D (PSO)	52
TABLA 13: N-DIMENSIONAL TEST FUNCTION 5D (PSO)	53
TABLA 14: TABLA RESUMEN PSO	54
TABLA 15: FUNCIÓN DE ROSENBROCK 2D (SA)	55
TABLA 16: FUNCIÓN DE ROSENBROCK 50D (SA)	56
TABLA 17: FUNCIÓN DE VENTER 2D (SA)	56
TABLA 18: FUNCIÓN DE BROWN 20D (SA)	57
TABLA 19: FUNCIÓN DE SCHWEFEL 5D (SA)	57
TABLA 20: N-DIMENSIONAL TEST FUNCTION 5D (SA)	58
TABLA 21: TABLA RESUMEN SA	59
TABLA 22: FUNCIÓN DE ROSENBROCK 2D (SAA-PSO)	60
TABLA 23: FUNCIÓN DE ROSENBROCK 50D (SAA-PSO)	60
TABLA 24: FUNCIÓN DE VENTER 2D (SAA-PSO)	61
TABLA 25: FUNCIÓN DE BROWN 20D (SAA-PSO)	61
TABLA 26: FUNCIÓN DE SCHWEFEL 5D (SAA-PSO)	61

TABLA 27: N-DIMENSIONAL TEST FUNCTION (SAA-PSO)	62
TABLA 28: TABLA RESUMEN SAA-PSO	63
TABLA 29: FUNCIÓN DE ROSENBROCK 2D (PSO-SAA)	64
TABLA 30: FUNCIÓN DE ROSENBROCK 50D (PSO-SAA)	64
TABLA 31: FUNCIÓN DE VENTER 2D (PSO-SAA)	65
TABLA 32: FUNCIÓN DE BROWN 20D (PSO-SAA)	65
TABLA 33: FUNCIÓN DE SCHWEFEL 5D (PSO-SAA)	65
TABLA 34: N-DIMENSIONAL TEST FUNCTION 5D (PSO-SAA)	66
TABLA 35: TABLA RESUMEN PSO-SAA	67
TABLA 36: FUNCIÓN DE ROSENBROCK 2D (GA-SAA)	68
TABLA 37: FUNCIÓN DE ROSENBROCK 50D (GA-SAA)	68
TABLA 38: FUNCIÓN DE VENTER 2D (GA-SAA)	69
TABLA 39: FUNCIÓN DE BROWN 20D (GA-SAA)	69
TABLA 40: FUNCIÓN DE SCHWEFEL 5D (GA-SAA)	69
TABLA 41: N-DIMENSIONAL TEST FUNCTION 5D (GA-SAA)	70
TABLA 42: TABLA RESUMEN GA-SAA	71
TABLA 43: TABLA GENERAL FUNCIÓN DE ROSENBROCK 2D	72
TABLA 44: TABLA GENERAL FUNCIÓN DE ROSENBROCK 50D	73
TABLA 45: TABLA GENERAL FUNCIÓN DE VENTER 2D	74
TABLA 46: TABLA GENERAL FUNCIÓN DE BROWN 20D	75
TABLA 47: TABLA GENERAL FUNCIÓN DE SCHWEFEL 5D	76
TABLA 48: TABLA GENERAL N-DIMENSIONAL TEST FUNCTION 5D	77

RESUMEN

TITULO:

EVALUACION DEL DESEMPEÑO DE ALGORITMOS DE EVOLUCION PARAMETRICA PARA OPTIMIZACION.*

AUTOR:

CAMILO EDUARDO CELIS MELO, CARLOS ANDRES MILLAN PARAMO. **

PALABRAS CLAVES:

Optimización, Simulated Annealing, Algoritmos Genéticos, Optimización con Enjambre de Partículas, Algoritmo Híbridos.

DESCRIPCIÓN:

Durante la última década, los métodos de optimización heurísticos basados en imitar a nivel computacional procesos naturales, biológicos, sociales o culturales, han despertado el interés de la comunidad científica debido a su capacidad para explorar eficientemente espacios de soluciones multimodales y multidimensionales.

En este ámbito, esta investigación aborda el desarrollo, análisis y puesta a punto de los siguientes métodos de optimización heurísticos: simulated annealing (SA), algoritmos genéticos (AG) y optimización con enjambre de partículas (PSO), posterior a esto se elaborara los algoritmos híbridos (de evolución paramétrica), SAA-PSO, PSO-SAA, GA-SAA, utilizando los algoritmos de optimización heurísticos.

Estos algoritmos serán puestos a prueba con funciones estándar reportadas en la literatura internacional, todo esto con el fin de analizarlos, compararlos y catalogarlos según el desempeño presentado frente a estas pruebas.

En cuanto a las medidas de desempeño están la desviación estándar de las funciones analíticas que se utilizará para medir la estabilidad y precisión de los algoritmos. El algoritmo será descrito como robusto, cuando se prueba en diferentes problemas, y ofrezca precisión y estabilidad. Y por último un algoritmo presentará confianza cuando logra determinar un número de veces el valor óptimo, en una cantidad de pruebas, tomando como referencia la primera cifra decimal.

1

* Proyecto de Grado

** Facultad de Ingenierías Físico-Mecánicas, Escuela de Ingeniería Civil, Director: Begambre Carrillo, Oscar.

ABSTRACT

TITLE:

ASSESSMENT OF PARAMETRIC EVOLUTION ALGORITHMS FOR OPTIMIZATION.*

AUTHOR:

CAMILO EDUARDO CELIS MELO, CARLOS ANDRES MILLAN PARAMO **

KEY WORDS:

Optimization, Simulated Annealing, Genetics Algorithms, Particles Swarm Optimization, Hybrids Algorithms.

DESCRIPTION:

During the last decade, heuristic optimization methods based on imitating natural, biological, social or cultural processes in a computational way have aroused great interest among the scientific community, due to its ability to explore efficiently multimodal and high-dimension solution spaces.

On this basis, this study tackles the development, analysis and tuning of the following heuristic optimization methods: simulated annealing (SA), genetics algorithms, (AG), and particle swarm optimization (PSO). Based on the algorithms described above, three hybrid algorithms (parametric evolution algorithms) named SAA-PSO, PSO-SAA, GA-SAA were developed.

These algorithms will be tested with standard functions reported in the international literature, all this in order to analyze, to classify and to compare them according to the presented performance against these tests.

As for the performance measures are the standard deviation of the analytical functions to be used to measure the stability and accuracy of algorithms. The algorithm is described as robust, when tested on different problems, and offer accuracy and stability. And finally present an algorithm able to determine confidence when a number of times the optimal value in a number of tests, with reference to the first decimal place.

* Degree Project

** Faculty of Physical-Mechanical Engineering, Civil Engineering, Director: Begambre Carrillo, Oscar.

INTRODUCCION

La aparición de los métodos de optimización heurísticos ha tenido una gran acogida en el ámbito científico, gracias a su confiabilidad y veracidad en cuanto a problemas de difícil manejo por medio de técnicas clásicas de optimización.

En primera instancia, en los años 1970 aparece la idea de los Algoritmos Genéticos (GA), la cual fue introducida por John Henry Holland y tiene su fundamento en la evolución biológica, teniendo en cuenta procesos de selección natural, mutación y cruce [1]

Mas adelante, hacia el año 1983 se presentó otra de estas técnicas de optimización conocida como Simulated Annealing (SA), la cual fue introducida por Kirkpatrick y colaboradores, en su escrito clásico "*Optimization by Simulated Annealing*" [2]. El SA se basa en la analogía entre el proceso de enfriamiento lento de solidos y la solución de problemas de optimización de gran porte.

En el año de 1995, fue propuesto el *Particle Swarm Optimization* (PSO), por James Kennedy y Russell C. Eberhart. Este algoritmo, representa el proceso mediante el cual un enjambre de partículas interactúa de forma tal que, mediante pasos controlados, llega hasta un punto óptimo [3].

Adicionalmente, se presenta en este estudio una corriente de hibridización de algoritmos heurísticos con el fin de obtener mejores resultados aprovechando las bondades de cada uno en particular, o facilitando el proceso de selección de parámetros de los mismos.

En este contexto, este trabajo de investigación presenta los resultados de la evaluación de desempeño de tres algoritmos (**SAA-PSO, PSO-SAA, GA-SAA**) propuestos dentro del proyecto “**Métodos heurísticos en la solución de problemas de Ingeniería civil: Estudio teórico-numérico**” desarrollado en el grupo de investigación INME de la escuela de ingeniería civil.

1. ALGORITMOS DE OPTIMIZACION

1.1. INTRODUCCION

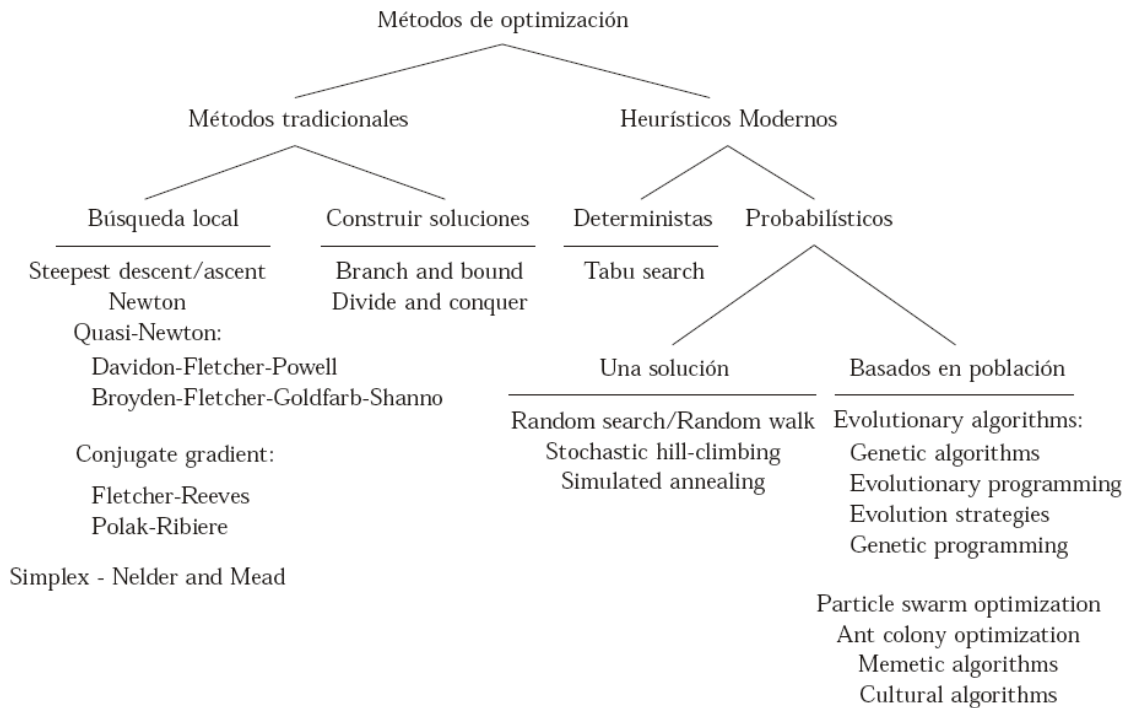
De forma genérica, puede definirse la optimización como aquella ciencia encargada de determinar las mejores soluciones a problemas matemáticos, que a menudo modelan una realidad física. Los problemas complejos de optimización multidimensional, no lineal y altamente multimodal, pueden encontrarse en ingeniería, economía, geofísica y, prácticamente, en todos los campos de la ciencia. En este aspecto, la optimización numérica ha adquirido mucha atención entre la comunidad científica, durante las últimas décadas. Teniendo en cuenta que un aspecto esencial de la optimización heurística es poder decidir que algoritmo y que parámetros de ese algoritmo se ajustan mejor a las características del problema físico bajo análisis, este trabajo pretende, entre otras cosas, dar a los ingenieros y diseñadores herramientas para afrontar esta tarea.

El objetivo que se persigue al resolver un problema de optimización es encontrar una solución óptima con un coste computacional razonable. Aunando estas dos premisas, puede establecerse una clasificación preliminar de los métodos de optimización en dos grandes bloques, distinguiendo por un lado los métodos de búsqueda local y, por otro, las así denominadas, técnicas de optimización global. Los métodos locales obtienen la mejor solución posible en las inmediaciones del punto inicial, atribuyéndoseles una fuerte dependencia del punto de arranque del algoritmo. La mayor parte de los métodos locales utilizan la información del gradiente, requieren el cálculo de derivadas y, en definitiva, imponen sobre el espacio de búsqueda unas condiciones de diferenciabilidad y continuidad difíciles de garantizar y controlar en la práctica. En el extremo opuesto, las técnicas de optimización global son capaces de atravesar un espacio de búsqueda con

múltiples mínimos o máximos locales y alcanzar una solución global al problema, entendiendo como tal, la mejor solución posible o una solución en las inmediaciones de la región que contiene el punto óptimo global. Como contrapartida, los métodos de optimización global exploran de forma exhaustiva el espacio de búsqueda, sin aprovechar sus características locales, como ocurre al operar por ejemplo, con gradientes, disminuyendo así su velocidad de convergencia.

Al margen de la mera distinción entre técnicas locales y globales de optimización, en la Figura 1 se muestra una clasificación detallada que recoge los métodos de optimización más representativos. Junto a los métodos de búsqueda local basados en derivadas, se incluye un método directo basado en simplex, cuyo atractivo reside en que únicamente requiere los valores de la función objetivo a optimizar. Los métodos tradicionales se completan con los así denominados constructivos, que básicamente trabajan sobre soluciones parciales del problema y que de alguna forma aíslan partes del espacio de búsqueda sin examinarlas. Como alternativa a los métodos tradicionales, se presentan los denominados métodos heurísticos modernos han adquirido durante la última década una notable aceptación en diferentes campos de la ingeniería, y entorno a algunos de ellos se centra esta investigación.

Figura1: Clasificación de los métodos de optimización más relevantes.



Fuente: Pérez López, Jesús Ramón. Contribución a los métodos de optimización basados en procesos naturales y su aplicación a la medida de antenas en campo próximo, 2005.

Todas las variantes de los métodos heurísticos probabilísticos que utilizan un único punto de partida, a excepción del Simulated Annealing (SA), tienen unos fundamentos muy sencillos, que se limitan a hacer evolucionar una solución inicial perturbando aleatoriamente los parámetros a optimizar. En lo que respecta al SA, este método imita a nivel computacional el proceso físico a seguir para obtener sólidos con configuraciones de energía mínima. Su sencillez y su potencial frente a otros métodos es un hecho comprobado, y su uso está muy extendido en aplicaciones tan diversas como la detección de daños en estructuras civiles [4], la síntesis de circuitos VLSI (Very Large Scale Integration) [5], el diseño de filtros digitales [6] o la optimización de agrupaciones de antenas [7].

En línea con el SA y en un intento por imitar procesos naturales tan dispares como la evolución de las especies o los propios comportamientos sociales y culturales de diferentes colectivos, entre los cuales puede incluirse a los propios seres humanos, surgen nuevos métodos que establecen una nueva concepción de la optimización. Todos estos algoritmos tienen en común el hecho de utilizar una población o conjunto de soluciones potenciales y someterlos a un proceso iterativo, utilizando diferentes esquemas, operadores y estrategias en función del tipo de algoritmo. La familia más extensa de este tipo de algoritmos es la que agrupa a los así denominados algoritmos evolutivos, estableciendo las cuatro grandes áreas que conforman lo que se conoce en la literatura como cómputo evolutivo. De estas cuatro metodologías, los algoritmos genéticos merecen una mención especial al ser el método que más atención ha recibido en los últimos años. Introducidos por Holland [1] e impulsados en años sucesivos por Goldberg, uno de sus estudiantes, los algoritmos genéticos han sido utilizados con éxito en múltiples campos de la ciencia, destacando a modo de ejemplo en electromagnetismo, su aplicación al diseño de filtros de microondas [8]. Otras aplicaciones de interés incluyen la planificación de red en sistemas de comunicaciones móviles [9], el entrenamiento de redes neuronales [10], la detección multiusuario en sistemas CDMA (Code Division Multiple Access) [11] o la planificación de redes de distribución de energía eléctrica [12]. Basado en los principios de la selección y evolución natural, el esquema clásico de los algoritmos genéticos hace evolucionar la población en base a la presión que ejercen los operadores de selección, cruce y mutación. Sin embargo, la amplia difusión del método de optimización ha hecho que los propios investigadores propongan continuas variantes y mejoras al esquema clásico, en un intento por incrementar el rendimiento del método.

A diferencia de los algoritmos genéticos, en los cuales el concepto de memoria en la optimización no existe como tal, limitándose en el caso de aplicar elitismo a seguir de forma muy pausada las tendencias del mejor individuo, existen otros

métodos heurísticos que por sus principios han experimentado un auge considerable en los últimos años, debido a que introducen el concepto de memoria. Entre estos métodos destaca por su atractivo la optimización con enjambre de partículas, más conocido como *particle swarm optimization*. Introducido como método de optimización por Kennedy y Eberhart [3], este método estocástico de optimización global se basa en imitar a nivel computacional el comportamiento de un colectivo a partir de la interacción entre sus miembros y con el entorno en el que éstos se desenvuelven, haciendo acelerar considerablemente la convergencia.

El término enjambre o *swarm* hace referencia a una colección de agentes, individuos o partículas, a los que se les atribuye una memoria y una capacidad de organizarse y cooperar entre sí. Los ejemplos más claros lo constituyen las abejas en su búsqueda de alimentos alrededor de la colmena, las bandadas de aves, el sistema inmune, que es en realidad un conjunto de células y moléculas, e incluso una muchedumbre puede verse como un grupo de personas que comparten impresiones para tomar decisiones, aprovechándose de los logros de sus congéneres y de su propia experiencia.

En contraste con los algoritmos genéticos, en la optimización con enjambre de partículas la población tiene memoria, es decir, la optimización se dirige y encauza influida por la historia pasada, por la memoria de cada individuo y por el estado presente en el que cada uno se encuentra. Este método ha sido aplicado al diseño de antenas [13]. Otras áreas incluyen, por ejemplo, la aplicación del algoritmo al entrenamiento de redes neuronales [14] o a la planificación de red en servicios de telecomunicación [15].

1.2. ALGORITMOS GENETICOS (AG)

Los Algoritmos Genéticos (AG) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin (1859). Por imitación de este proceso, los AG son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas.

En la naturaleza los individuos de una población compiten entre sí en la búsqueda de recursos tales como comida, agua y refugio. Incluso los miembros de una misma especie compiten a menudo en la búsqueda de un compañero. Aquellos individuos que tienen más éxito en sobrevivir y en atraer compañeros tienen mayor probabilidad de generar un gran número de descendientes. Por el contrario individuos poco adaptados producirán un menor número de descendientes. Esto significa que los genes de los individuos mejor adaptados se propagarán en sucesivas generaciones hacia un número de individuos creciente. La combinación de buenas características provenientes de diferentes ancestros, puede a veces producir descendientes “superindividuos”, cuya adaptación es mucho mayor que la de cualquiera de sus ancestros. De esta manera, las especies evolucionan logrando unas características cada vez mejor adaptadas al entorno en el que viven.

Los AG usan una analogía directa con el comportamiento natural. Trabajan con una población de individuos, cada uno de los cuales representa una solución factible a un problema dado. A cada individuo se le asigna un valor o puntuación, relacionado con la bondad de dicha solución. En la naturaleza esto equivaldría al

grado de efectividad de un organismo para competir por unos determinados recursos. Cuanto mayor sea la adaptación de un individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma. Este cruce producirá nuevos individuos, los cuales comparten algunas de las características de sus padres. Cuanto menor sea la adaptación de un individuo, menor será la probabilidad de que dicho individuo sea seleccionado para la reproducción, y por tanto de que su material genético se propague en sucesivas generaciones [1].

De esta manera se produce una nueva población de posibles soluciones, la cual reemplaza a la anterior y verifica la interesante propiedad de que contiene una mayor proporción de buenas características en comparación con la población anterior. Así a lo largo de las generaciones las buenas características se propagan a través de la población. Favoreciendo el cruce de los individuos mejor adaptados, van siendo exploradas las áreas más prometedoras del espacio de búsqueda. Si el AG ha sido bien diseñado, la población convergerá hacia una solución óptima del problema.

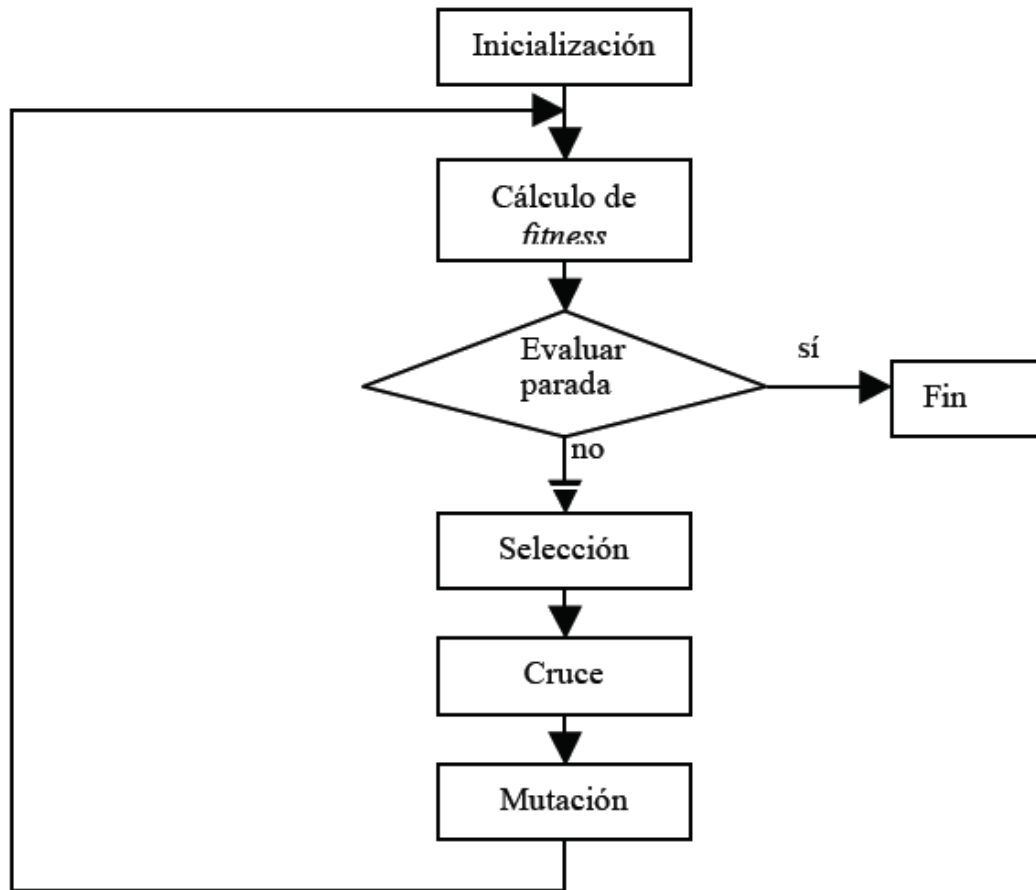
El desarrollo conceptual de los AG tiene, desde sus orígenes, un respaldo matemático que lo avala, basado principalmente en el así denominado teorema del esquema [1], el cual conjuga fitness, cruce y mutación para establecer cómo afectan a la supervivencia y propagación de las soluciones aspectos tales como el esquema de representación de los datos.

Durante la última década, el uso de los AG se ha extendido a múltiples áreas. Entre las diferentes aplicaciones de los AG pueden resaltarse, entre otras, optimización de componentes de concreto [16], el modelado de fuente en compatibilidad electromagnética [17] y la transformación de campo cercano a campo lejano [18].

Originariamente, el esquema básico de los AG incluía selección, cruce y mutación, con una representación en binario de los parámetros a optimizar. Sin embargo, las modificaciones al esquema original se han sucedido, y de hecho aún se suceden de una forma continua, en un intento por mejorar el rendimiento del algoritmo. Así, pueden encontrarse en la literatura como implementaciones más habituales los AG de codificación real [19], los algoritmos micro-genéticos (mAG) [20] y los algoritmos genéticos híbridos (HAG) [21].

El éxito de los AG radica, más que en el exotismo o en la difusión que de éste se ha hecho para ganar nuevos adeptos, en su propio potencial y facilidad de implementación. De hecho, en la actualidad existen múltiples tutoriales [22], y recursos que hacen aún más fácil su uso.

Figura 2: Diagrama de flujo de un Algoritmo Genético.



Fuente: Los Autores

1.2.1. Algoritmos Genéticos en Codificación Binaria

En sus orígenes, los AG utilizaron como esquema de representación de los datos el alfabeto binario, aunque, en realidad, el método puede extenderse sin grandes cambios estructurales a cualquier otro alfabeto, en consonancia con la naturaleza del problema a optimizar.

Tal y como ya se anticipó con anterioridad, los AG son métodos estocásticos de búsqueda basados en los principios de la evolución biológica. Básicamente, un organismo se compone de células, cada una de las cuales contiene el mismo conjunto de cromosomas. A su vez, los cromosomas, compuestos por genes, son cadenas de ADN que definen un modelo para todo el organismo. Cada gen codifica una cierta proteína, puede decirse que codifica un rasgo del organismo, y los valores que puede tomar este rasgo reciben el nombre de alelos. Por último, resaltar que cada gen ocupa su propia posición dentro del cromosoma. A grandes rasgos, la evolución generacional alterará los cromosomas y, en consecuencia, las características de los nuevos organismos. El rango dinámico de cada variable o gen equivale a los alelos y, en conjunto, el rango dinámico de los N genes define el espacio de búsqueda N -dimensional. Para formalizar la búsqueda, una población de I cromosomas o individuos se hace evolucionar generacionalmente hacia una solución óptima, aplicando sobre ellos operaciones de selección, cruce y mutación que garantizan la supervivencia de los mejores individuos, utilizando una función objetivo para cuantificar la calidad de cada uno de ellos.

1.2.2. Representación, Selección, Cruce y Mutación

La representación de los parámetros se hace en código binario, utilizando para codificar cada variable un número de bits nb , que normalmente es seleccionado en función del rango dinámico de la variable y de la resolución deseada. Ahora bien, el número de bits guarda una relación directa con el tamaño de la población. Es decir, cuanto mayor sea la longitud del cromosoma mayor deberá ser el número de individuos que integran la población, con el objetivo de mantener la diversidad entre los cromosomas y explorar eficientemente el espacio N -dimensional. Consecuentemente, al aumentar el tamaño de la población se aumenta el número de llamadas a la función objetivo. Si consideramos que habitualmente en problemas complejos entorno al 95% del tiempo de CPU en cada iteración se

dedica a la evaluación de la función objetivo, puede llegarse a la conclusión de que la selección del número de bits juega un papel determinante [22].

En lo referente a la estrategia de selección, se utiliza selección por torneo, donde básicamente se seleccionan aleatoriamente N_c cromosomas de la población. Aquel cromosoma de entre los N_c con el mejor fitness se selecciona como uno de los dos candidatos a participar en la reproducción. El proceso se repite para escoger al segundo candidato. Otras estrategias tales como la selección proporcional, más conocida como *roulette wheel selection*, y la selección por ranking, también son de gran utilidad. La selección por ranking es determinista y al propagar generacionalmente las mejores soluciones elimina progresivamente la diversidad entre los miembros de la población. En la selección proporcional, aún teniendo naturaleza estocástica, si no se introduce un escalado del fitness adecuado actúa de forma análoga a la selección por ranking, transmitiendo múltiples réplicas de un mismo individuo a las siguientes generaciones, modificadas únicamente por los operadores de cruce y mutación. Sin embargo, los individuos con peor fitness apenas si participan en el ciclo de reproducción, eliminando así la posibilidad de explorar nuevas zonas del espacio de búsqueda y reduciendo paulatinamente la diversidad.

La aplicación de los operadores de cruce y mutación completa la reproducción generacional. El algoritmo permite utilizar para el operador de cruce las técnicas de cruce en un punto o cruce uniforme, y para el operador de mutación las de mutación por salto o mutación sigilosa. Seleccionados los dos padres, los hijos se crean inicialmente como sendas réplicas de sus progenitores y sólo se cruzan con probabilidad P_c . En el caso de proceder a la recombinación, el cruce en un punto simple selecciona aleatoriamente un punto dentro del cromosoma a partir del cual todos los bits de ambos hijos se intercambian entre sí para dar vida a los nuevos individuos.

Para completar la creación de los nuevos hijos (prole) resta aún por aplicar el operador de mutación. La mutación se caracteriza por introducir leves modificaciones en el cromosoma con el objetivo de permitir al algoritmo explorar nuevas áreas del espacio de soluciones, manteniendo a la vez la diversidad en la población. Si no se aplicara mutación, llegaría un momento en el que la presión ejercida por la selección por torneo conduciría a todos los individuos hacia una misma región dentro del espacio de soluciones, que en la mayoría de los casos estaría asociada con una convergencia prematura hacia una solución errónea. En lo referente a las estrategias de mutación empleadas, la mutación por salto (*jump mutation*) se aplica a todo el cromosoma con probabilidad P_m . Para todos aquellos bits que proceda, de acuerdo con P_m , la mutación por salto se limita a cambiar el estado del bit. Por el contrario, la mutación sigilosa (*creep mutation*) es más compleja. En este caso se seleccionan para todo el cromosoma y con probabilidad P_m aquellos genes que van a someterse a mutación. Dichos genes se decodifican, su valor se muta un cierto factor fc hacia arriba o hacia abajo y finalmente se codifica el gen con el nuevo valor modificado.

1.2.3. Algoritmos Genéticos en Codificación Real

En los diseños iniciales de AG propuestos por Holland y desarrollados posteriormente por su discípulo Goldberg, se utiliza codificación binaria para representar a los genes que integran cada cromosoma de la población. Sin embargo, puede utilizarse cualquier alfabeto, siempre y cuando éste permita establecer un mapeo entre la representación de los genes en términos del propio alfabeto y el significado físico de los parámetros a los que éstos representan. Entre las alternativas más triviales al alfabeto binario destacan las representaciones que utilizan números reales, números enteros, símbolos, e incluso híbridos.

1.2.4. Representación, Selección, Cruce y Mutación

La representación de un cromosoma utilizando números reales es más fácil de entender por los seres humanos más que el uso de una cadena de bits. Además, utilizando números reales la representación requiere menos memoria. Otra ventaja de la representación real es su versatilidad en la creación de nuevos operadores y los cambios en los operadores existentes [16].

Inspirado en el proceso de selección natural de los seres vivos, el algoritmo genético a través del operador de selección, selecciona los mejores cromosomas de la población para determinar cuáles individuos pueden participar en el proceso de reproducción y ayudar en la formación de la próxima generación. El proceso de reproducción se realiza mediante un operador llamado cruce genético. Las principales formas de selección de los individuos para la reproducción de la población son: selección por ranking, selección por ruleta y selección por torneo [16].

Entre las múltiples alternativas existentes en la literatura para implementar el operador de cruce en el dominio real, se seleccionaron las estrategias de cruce discreto, continuo, convexo o aritmético y heurístico [23]. Para las cuatro estrategias de cruce analizadas se contempla la posibilidad de obtener uno ó dos hijos como resultado del cruce. El número de hijos permite controlar indirectamente la presión que se ejerce sobre la selección y sobre la convergencia. Si se utiliza un único hijo por cada recombinación es más probable que los mejores individuos participen en un mayor número de torneos y, con ello, transmitan mayor cantidad de información genética a generaciones futuras.

Para los AG de codificación real se puede utilizar ocho alternativas para el operador de mutación: mutación sigilosa [24], mutación límite o frontera, dos

variantes de mutación no uniforme, tres esquemas de mutación multi-no-uniforme y mutación uniforme [25], [26].

1.2.5. Diferencias entre codificación Real y Binaria

Las diferencias, ventajas e inconvenientes del tipo de codificación están relacionadas con tres aspectos fundamentales, cuales son la complejidad del algoritmo, el coste computacional y la precisión de los resultados.

En cuanto a la complejidad del algoritmo, cada alfabeto tiene sus ventajas e inconvenientes. En el caso binario, la selección del número de bits, las operaciones de codificación y decodificación, la mayor longitud del cromosoma y la necesidad de utilizar más recursos para almacenar y manejar las variables, constituyen las principales limitaciones de este esquema. Por el contrario, la codificación real parte de una estructura interna mucho más sencilla en la que se suprimen todas las limitaciones del esquema binario pero que, como contrapartida, utiliza estrategias para los operadores de cruce y mutación mucho más complejas. En lo que respecta al coste computacional, el AG real es supuestamente más rápido. Sin embargo, esta afirmación puede variar dependiendo del tipo de problema al que se aplique una u otra codificación.

1.3. OPTIMIZACION CON ENJAMBRE DE PARTICULAS (PSO)

La optimización con enjambre de partículas, más conocida en la literatura científica como *Particle Swarm Optimization* (PSO), nace, al igual que otras técnicas estocásticas de cálculo evolutivo, en un intento por imitar y mimetizar el comportamiento de procesos naturales. El PSO junto con la optimización con una colonia de hormigas o *Ant Colony Optimization* (ACO) son los dos métodos más

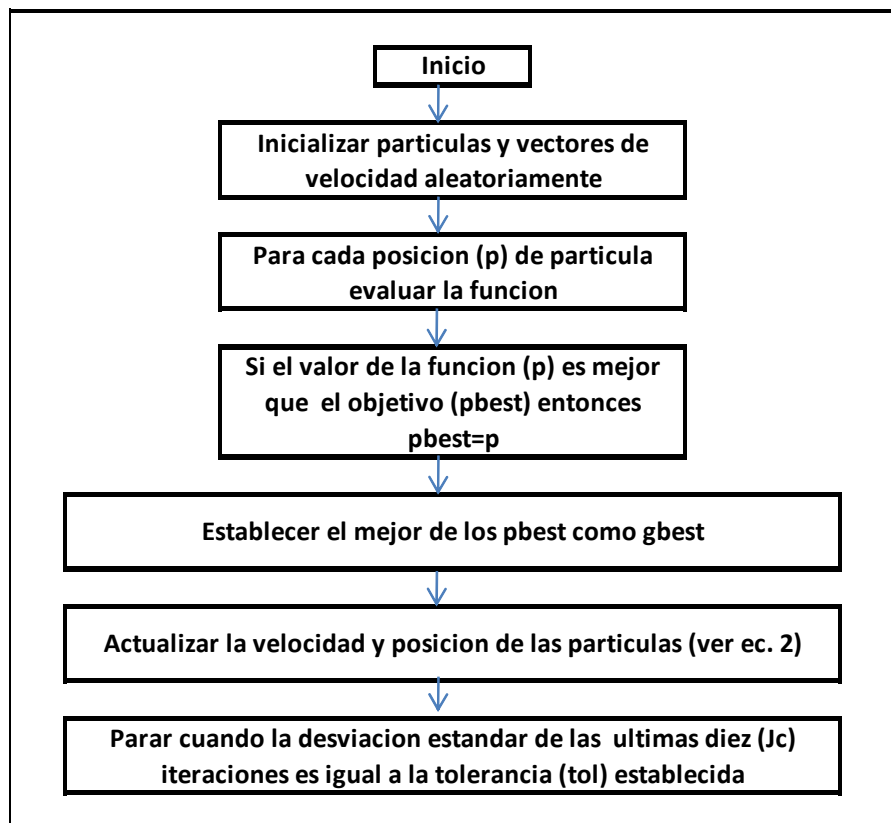
populares y utilizados en el área de la inteligencia computacional. Ambos tratan de imitar los comportamientos sociales de un colectivo a partir de la interacción de los individuos entre sí y con el entorno.

Los orígenes del PSO como método estocástico de optimización global se remontan a los estudios iniciados por Kennedy y Eberhart [3], quienes se fijan como objetivo inicial simular gráficamente el movimiento sincronizado e impredecible de grupos tales como los bancos de peces o las bandadas de aves, intrigados por la capacidad de estos grupos para separarse, reagruparse o encontrar alimento.

En la terminología utilizada en PSO, Kennedy y Eberhart, introducen el término general *partícula* o *agente* para representar a los peces, pájaros, abejas, hormigas o cualquier otro tipo de individuos que exhiban un comportamiento social como grupo, en forma de una colección de agentes que interactúan entre sí. De acuerdo con los fundamentos teóricos del método, el movimiento de cada una de estas partículas hacia un objetivo común en dos dimensiones está condicionado por dos factores básicos, la memoria autobiográfica de la partícula o nostalgia y la influencia social de todo el enjambre. A nivel computacional, como método de optimización, esta filosofía puede extenderse a un espacio N -dimensional de acuerdo con el problema bajo análisis. La posición instantánea de cada una de las partículas de la población en el espacio N -dimensional representa una solución potencial, siendo N el número de incógnitas del problema original. Básicamente, el proceso evolutivo se reduce a mover cada partícula dentro del espacio de soluciones con una velocidad que variará de acuerdo a su velocidad actual, a la memoria de la partícula y a la información global que comparte el resto del enjambre, utilizando una función objetivo para cuantificar la calidad de cada partícula en función de la posición que ésta ocupe.

Trasladando la filosofía de PSO al campo de la vida artificial y del cómputo evolutivo, este método guarda cierta analogía con los AG. Sin embargo, su extremada simplicidad de programación y el menor número de parámetros de los que depende el funcionamiento interno del algoritmo, le convierten en una alternativa atractiva a los AG. Entre las múltiples áreas donde ha sido aplicado con éxito, destacan por su importancia: identificación de fallas en estructuras civiles [27], optimización de funciones y resolución de problemas matemáticos complejos [28], entrenamiento de redes neuronales [29], optimización de sistemas dinámicos [30], gestión, planificación y optimización de recursos en redes de distribución de energía eléctrica [31], gestión empresarial [32] y teoría de juegos [33], entre otros.

Figura 3: Diagrama de Flujo algoritmo PSO



Fuente: www23.homepage.villanova.edu/varadarajan.komanduri/PSO_meander-line.ppt

1.3.1. Fundamentos del Movimiento de Partículas

Para aplicaciones en el ámbito de la vida artificial se deben respetar cinco principios básicos de lo que se entiende como inteligencia de grupo, más comúnmente recogido en la literatura como *swarm intelligence* [34]. Estos principios reciben el nombre de proximidad, calidad, diversidad de respuesta, estabilidad y adaptabilidad. En el presente apartado se justifica el modo en el que el PSO se adhiere a estos cinco postulados [3].

De acuerdo con el principio de proximidad, la población debiera ser capaz de realizar cálculos sencillos de espacio y tiempo, lo cual se traduce en PSO a movimientos en N dimensiones llevadas a cabo durante una serie de intervalos de tiempo que coinciden con movimientos de la población a una determinada velocidad. El segundo principio promueve la capacidad de la población para responder a factores de calidad en el espacio de soluciones, lo que se consigue en PSO con la memoria de la partícula y con la historia o conocimiento social que comparten entre sí todos los congéneres. El tercer principio promueve la diversidad de respuesta dentro de la población, y está garantizado en PSO por las diferentes tendencias marcadas por la memoria personal de cada partícula y por la historia de la mejor posición visitada por todo el conjunto. El cuarto y quinto principio resaltan aspectos contrapuestos. Por un lado, la población sólo cambia su comportamiento como grupo cuando se actualiza la mejor posición históricamente visitada por alguno de los miembros que lo integran, respetando así el principio de estabilidad. Por el contrario, de acuerdo con el quinto principio, la población debe de ser a su vez adaptativa, es decir, debe ser capaz de modificar su comportamiento y movimiento cuando hay alguna señal que así lo recomienda desde el punto de vista de ahorro computacional o de mejora en la precisión. Esta premisa se consigue fácilmente en PSO, dado que la población en

su conjunto cambia su rumbo cuando alguna de las partículas alcanza una solución global que mejora el resultado [34].

1.3.2. Operador de Velocidad y Parámetros de Algoritmo

Las partículas evolucionan de acuerdo al siguiente conjunto de ecuaciones vectoriales:

$$P_{k+1}^i = P_k^i + V_{k+1}^i \quad (1)$$

$$V_{k+1}^i = wV_k^i + C_1rand_1(b_k^i - P_k^i) + C_2rand_2(b_k^g - P_k^i) \quad (2)$$

Donde, P_k^i representa la posición de cada partícula i en la iteración k , P_{k+1}^i es la posición de la partícula i en la iteración $k + 1$, b_k^i almacena la mejor posición de la partícula i en la iteración k (mejor posición individual), b_k^g es la mejor posición lograda por el enjambre en la iteración k (mejor posición del líder), V_k^i corresponde a la velocidad de la partícula i durante la iteración k y V_{k+1}^i es la velocidad de la ajustada partícula i en la iteración $k + 1$.

Todos los vectores de la ecuación 1 y 2 son de dimensión $m + 1$ donde m es la cantidad de parámetros optimizados, $rand_1$ y $rand_2$ son números aleatorios independientes con probabilidad uniforme entre 0 y 1.

Los parámetros cognitivo (C_1) y social (C_2) controlan el flujo de información dentro del enjambre, si $C_2 > C_1$ entonces la partícula va a confiar más en los resultados de la búsqueda del enjambre, de lo contrario, $C_2 < C_1$, ella va a tener más confianza en los resultados de su búsqueda que en los del enjambre.

El factor de inercia w controla la influencia de la velocidad previa de la partícula sobre la velocidad actual. Un factor de inercia alto facilita la exploración global del espacio de búsqueda mientras que un valor pequeño posibilita realizar una búsqueda local. Es posible reducir el número de iteraciones del algoritmo al

encontrar un equilibrio entre las capacidades de búsqueda local y búsqueda global del algoritmo.

Con el fin de mantener un equilibrio entre las capacidades de búsqueda local y global se propone usar de forma general las siguientes condiciones, $0.4 < w < 1.4$, $0.5 < C_1 < 2$, $0.5 < C_2 < 2$. Como se ha mencionado, la escogencia de estos parámetros depende de las particularidades del problema de resolver, en consecuencia se debe realizar pruebas exhaustivas para encontrar el mejor conjunto de parámetros que garanticen el éxito de la búsqueda del PSO en cada caso, ya que valores inadecuados podrían causar fallas en el algoritmo [35].

1.3.3. Analogías y Diferencias entre PSO y AG

El método de optimización PSO guarda ciertas similitudes con otras técnicas de computación evolutiva tales como los AG. Ambos métodos estocásticos se inicializan con una población de soluciones potenciales, partículas o agentes en el caso de PSO y cromosomas o individuos en el caso de AG, y realizan la búsqueda de una solución óptima en base a un proceso iterativo y generacional, utilizando una función de aptitud para medir la precisión de cada solución. Así mismo, ninguna de las dos técnicas garantiza el éxito pleno de la optimización, el cual está relacionado directamente con la configuración del algoritmo para el problema a optimizar. A grandes rasgos, se puede afirmar que PSO es a la simulación de la interacción social entre individuos, y que los algoritmos genéticos son la simulación de la evolución de las especies.

Los AG utilizan los operadores de selección, cruce y mutación, cada uno de los cuales admiten múltiples implementaciones y tiene diferentes parámetros a seleccionar y afinar de acuerdo con la naturaleza del problema a optimizar. Sin embargo, el PSO únicamente tiene un operador, la velocidad de partícula. Aunque

el PSO no tiene explícitamente definidos operadores de evolución tales como cruce o mutación, en realidad, el ajuste de la posición de las partículas en la dirección de la mejor solución personal, b_k^i , y de la mejor solución de conjunto, b_k^g , de acuerdo con el numeral anterior, tiene un significado conceptualmente similar al de cruce y mutación en los AG.

Otra diferencia existente entre AG y PSO hace referencia a la capacidad de control sobre la convergencia de la población. Las probabilidades de cruce y mutación en los AG sugieren una vía para controlar la convergencia del método, y por tanto, la diversidad de los cromosomas, mientras que en PSO el factor de inercia w , permite realizar dicho control de forma mucho más sutil y directa. El mecanismo que posee PSO para compartir información entre congéneres es significativamente distinto al de los AG. En los AG todos los cromosomas comparten información entre sí, de forma que la población se mueve como un grupo hacia una región óptima, mientras que en PSO es el mejor agente, b_k^g , el que transmite la información al resto y condiciona el movimiento del enjambre. En este caso, la evolución sólo busca la mejor solución, de tal forma que, por lo general, todas las partículas tienden a converger más rápidamente hacia la solución óptima.

A diferencia de los AG, las partículas en PSO tienen memoria. Los cambios en la población de individuos en AG, asociados con los operadores de cruce y mutación, deparan la destrucción del conocimiento previo del problema, salvo que se utilice elitismo, en cuyo caso sólo uno o un reducido grupo de individuos retienen información de las mejores soluciones. En PSO sin embargo, los individuos que al atravesar el espacio de soluciones pasan cerca de puntos óptimos son forzados a volver hacia éstos, reteniendo información de las soluciones alcanzadas. Esta información es aprovechada a posteriori por el resto de las partículas del enjambre.

Expresado en otros términos, se puede afirmar que en PSO los agentes mantienen sus identidades con el transcurso del tiempo, mejorando por imitación y a través de la interacción con sus vecinos.

En comparación con los AG, las diferencias y ventajas del método PSO se resumen en tres puntos fundamentales: la sencillez de implementación, la rapidez de convergencia y el reducido número de parámetros a ajustar para sintonizar el algoritmo. Sin embargo, estas conclusiones no pueden generalizarse, dado que el rendimiento de uno u otro método de optimización están condicionados en última instancia por la naturaleza del problema a analizar [36].

1.4. SIMULATED ANNEALING (SA)

El SA es uno de los métodos de optimización clásicos. Su simplicidad y buenos resultados en numerosos problemas lo han convertido en una herramienta muy popular con cientos de aplicaciones en los más variados campos. Su nombre viene de la idea en que está basado un algoritmo de los años 50 [37] para simular el endurecimiento por calentamiento y enfriado lento de un material. Treinta años más tarde fue creado el algoritmo actual [2].

A diferencia de otros métodos como los AG o el PSO, en el SA sólo se hace evolucionar una solución y no se conserva memoria alguna respecto a la historia pasada. Al igual que en los otros métodos estocásticos de optimización global, la optimización se basa en un proceso que, aún siendo simple, lleva asociado un coste computacional muy superior al de las técnicas de optimización local. La ventaja del método de SA está relacionada con el número de parámetros que se precisan ajustar para poner a punto el algoritmo, reduciéndose básicamente a definir de forma correcta el esquema de enfriamiento y la temperatura inicial para el problema bajo análisis.

El SA es una variante de la búsqueda local que permite movimientos ascendentes para evitar quedar atrapado prematuramente en un óptimo local. Los algoritmos tradicionales de búsqueda local parten de una solución inicial que ha modo paulatino va siendo transformada en otras que a su vez son mejoradas al introducirle pequeños cambios (tales como modificar el valor de una variable). Si el cambio da origen a una solución mejorada se sustituye la nueva solución por la actual y se continúa el proceso hasta que no sea posible encontrar una nueva mejora. En consecuencia, la búsqueda finaliza en un óptimo local, que no tiene porque ser forzosamente el global.

Un modo de evitar este problema es permitir que algunos movimientos sean hacia soluciones peores, sin embargo, estos movimientos de escape deben realizarse de un modo controlado por si el algoritmo en realidad está yendo hacia una buena solución. En el caso particular del SA, esto se realiza controlando la frecuencia de los movimientos de escape mediante una función de probabilidad que hará disminuir la probabilidad de esos movimientos hacia soluciones peores, conforme avanza la búsqueda.

La fundamentación de este control se basa en el trabajo realizado por Metrópolis en el año de 1953 [30], en el cual modelo el proceso de atemperamiento de un material simulando los cambios energéticos en un sistema conforme decrece la temperatura t , la probabilidad de un incremento energético de magnitud δE se puede aproximar por la ecuación 3, siendo k la constante de Boltzmann.

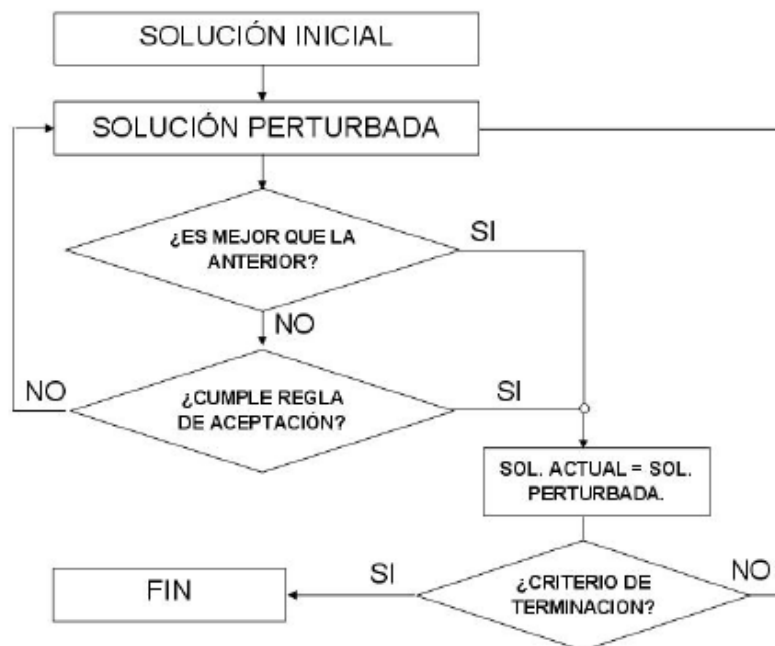
$$P[\delta E] = e^{\frac{\delta E}{kt}} \quad (3)$$

En cuanto al ámbito de aplicación del método, éste está muy extendido entre la comunidad científica. El SA se aplica a la optimización de antenas y a problemas

de síntesis de agrupaciones de muy diversa índole. Por otra parte, se aplica el método a la optimización de los coeficientes de filtros digitales [35].

El proceso de SA al igual que el de otros métodos es iterativo y cada iteración corresponde a una fase de enfriamiento, en cada iteración se realiza una perturbación de la solución, el reemplazo que produce esta perturbación se hará si es mejor que la solución actual o si cumple cierta regla de aceptación que es probabilística y está en función de la temperatura.

Figura 4: Proceso básico que se sigue en un algoritmo de Simulated Annealing.



Fuente: Estado del Arte del Problema de Ruteo de Vehículos (VRP), Nicolás Barajas, Universidad Nacional de Colombia.

La regla de aceptación consiste en que existe una probabilidad, de que la solución perturbada se convierta en la actual, y está dada por (3).

De esta manera, finaliza la parte conceptual en lo que concierne a los Algoritmos de Optimización heurísticos tratados en este estudio.

2. ALGORITMOS HIBRIDOS

2.1. INTRODUCCION

Una de las principales áreas de trabajo en el ámbito de la optimización ha sido tradicionalmente el diseño de algoritmos cada vez más eficientes para la solución de problemas en diversas áreas de la ingeniería y la ciencia. En este dominio, el objetivo consiste en obtener nuevos algoritmos que necesiten un esfuerzo computacional menor, o que posean propiedades de convergencia y estabilidad igual o mejor al de los existentes, así como caracterizar su comportamiento para las clases de problemas que demanda la comunidad científica e industrial en general.

La investigación en algoritmos tanto exactos como heurísticos para resolver problemas de optimización tiene vigencia en la actualidad, ya que nos enfrentamos a nuevos problemas de ingeniería, al mismo tiempo que contamos con nuevos recursos computacionales tales como nuevos tipos de máquinas, redes y entornos como Internet.

De esta forma, en este momento, es acuciante la necesidad de algoritmos muy eficientes que puedan dar respuesta en tiempo real a problemas del usuario o del sistema, y que permitan trabajar con los nuevos sistemas operativos y máquinas. Para ello, es muy importante poder caracterizar el funcionamiento de dichos algoritmos para conocer limitaciones y potencia intrínseca al resolver nuevos problemas no estudiados.

Una de las ramas con éxito para diseñar algoritmos eficientes en la actualidad es la hibridización. La hibridización permite incorporar información del problema en el

algoritmo de resolución para trabajar en contacto con sus características diferenciadoras; esto tiene también relación con la posibilidad de involucrar a varios algoritmos distintos en el proceso de búsqueda de soluciones de manera más eficiente, ya que usualmente se obtienen mejoras notables (en relación a precisión, estabilidad o tiempo de computo) en la determinación de puntos óptimos cuando se utiliza un algoritmo híbrido.

2.2. ESTADO DEL ARTE

2.2.1. Híbridos referenciados en la literatura internacional.

Híbrido PSOSA, estrategia eficaz para la optimización y su aplicación para la estimación de parámetros [39].

La estimación de parámetros, en especial para sistemas no lineales, es un problema muy importante en cuanto a la teoría de control y la práctica de la ingeniería de control. Esencialmente, la estimación de parámetros es un problema de optimización numérica multidimensional, con múltiples óptimos locales, muy difícil de resolver, especialmente para algunos tipos especiales de parámetros.

Híbrido PSOSA, estrategia

Todo el proceso de PSOSA se describe de la siguiente manera:

Paso 1 (Inicialización):

Paso 1.1: $k=0$ y al azar inicializar $X_i(0)$ y $V_i(0)$, $i=1,2,\dots, n$.

Paso 1.2: Evaluar la función objetivo F_i para todas las partículas.

Paso 1.3: Inicializar P_i con una copia de $X_i(0)$, $i=1,2,\dots, n$; e inicializar P_g con una copia del mejor P_i .

Paso 1.4: Inicializar la temperatura $t(0)=t_0$.

Paso 2: Repetir hasta que finalice con el criterio de parada

Paso2.1: Evaluar el fitness para cada P_i con la siguiente ecuación (Metrópolis):

$$Fitness_i = \exp \left[-\frac{Fp_i - Fp_g}{t(k)} \right]$$

Paso 2.2: Para cada partícula i , se aplica la selección por ruleta para seleccionar P'_g del conjunto de los P_i y luego actualizar X_i y V_i acorde a las siguientes ecuaciones:

$$V_i(k + 1) = \chi * \{V_i(k) + C_1 * rand(.) * [P_i - X_i(k)] + C_2 * Rand(.) * [P'_g - X_i(k)]\}$$

$$X_i(k + 1) = X_i(k) + V_i(k + 1)$$

Donde χ es un coeficiente de constricción y se halla de la siguiente manera:

$$\chi = \frac{2}{|2 - C - \sqrt{C^2 - 4C}|}, \quad C = C_1 + C_2.$$

Paso 2.3: Evaluar F_i para todas las partículas.

Paso 2.4: Actualizar P_i y P_g (incluyendo valores de posición y valores objetivos).

Paso 2.5: Disminuir la temperatura, $t(k+1)$.

Paso 3: Salida de la mejor solución P_g y su valor de función F_g .

GASA, Algoritmo Híbrido aplicado a aerolíneas en el sistema de turnos (o listas) de tripulación [40].

Las listas (turnos) de tripulación es el proceso para generar las horas de trabajo que satisfaga a algunas restricciones por un período de tiempo. Una justa y razonable lista desempeña un papel muy importante para despertar el entusiasmo de la productividad del trabajador y el trabajo, y también trae un gran beneficio económico.

Algoritmo Genético (AG) es un algoritmo de optimización global formado a partir de la simulación biológica de la evolución genética. Su esencia es un proceso en grupo interactivo. Simulated Annealing provee un método que gasta poco tiempo de trabajo. Este híbrido tiene diferencias con los algoritmos originales ya que ha enriquecido la búsqueda en el proceso de optimización y ha incrementado en conjunto la capacidad de la búsqueda local y su eficiencia. Esta es una optimización con alta capacidad, eficiencia y confiabilidad.

Los pasos del algoritmo híbrido son los siguientes:

Paso 1

Producir aleatoriamente un grupo inicial y evaluarlos en la función objetivo.

Paso 2

Comprobar si cumple la regla de aceptación. Si la cumple, exportar los resultados. De lo contrario separar el grupo en 5 partes.

Paso 3

Para 1, 2, 3, 4 y 5 cruzarlos y escoger un nuevo grupo utilizando una buena técnica de elección.

Paso 4

Ejecutar Simulated Annealing

Paso 5

Chequear el criterio de Metrópolis, Si no lo cumple volver al Paso 2.

Paso 6

Generar una población partiendo de la actual.

Paso 7

Estimar el nuevo estado. Si es satisfactorio cambiar el valor optimo. Si no volver al Paso 5.

Paso 8.

Fin de los algoritmos.

Hibrido de evolución prametrica PSOS para la identificación de daño estructural [41]

Este estudio propone un nuevo modelo PSOS basado en la identificación de daño utilizando el procedimiento de identificación de frecuencia de datos de dominio. La formulación de la función objetivo para la reducción del problema se basa en la respuesta en frecuencia de Funciones (FRFs) del sistema. Una nueva estrategia para el control de los parámetros del PSO basándose en la escogencia realizada por Nelder-Mead (método simplex) es presentada, por consiguiente, la convergencia de las PSO se independiza de la heurística y sus constantes de la estabilidad y la confianza se refuerzan.

La idea central del método es que el algoritmo Simplex seleccione los parámetros (N , w , $C1$ y $C2$) limitados por un rango de valores, y así procurar una configuración óptima para el PSO. Dentro de esta heurística cada vértice del Simplex queda definido por las coordenadas (N , w , $C1$ y $C2$). El PSO toma valores de parámetros determinados por el Simplex y evalúa la función objetivo. El valor de la función objetivo es devuelto al Simplex y este reajusta sus vértices siguiendo en la búsqueda de los parámetros óptimos del PSO. Este ciclo se repite hasta llegar al valor óptimo de la función objetivo.

2.3. ALGORITMOS HÍBRIDOS DE EVOLUCIÓN PARAMÉTRICA

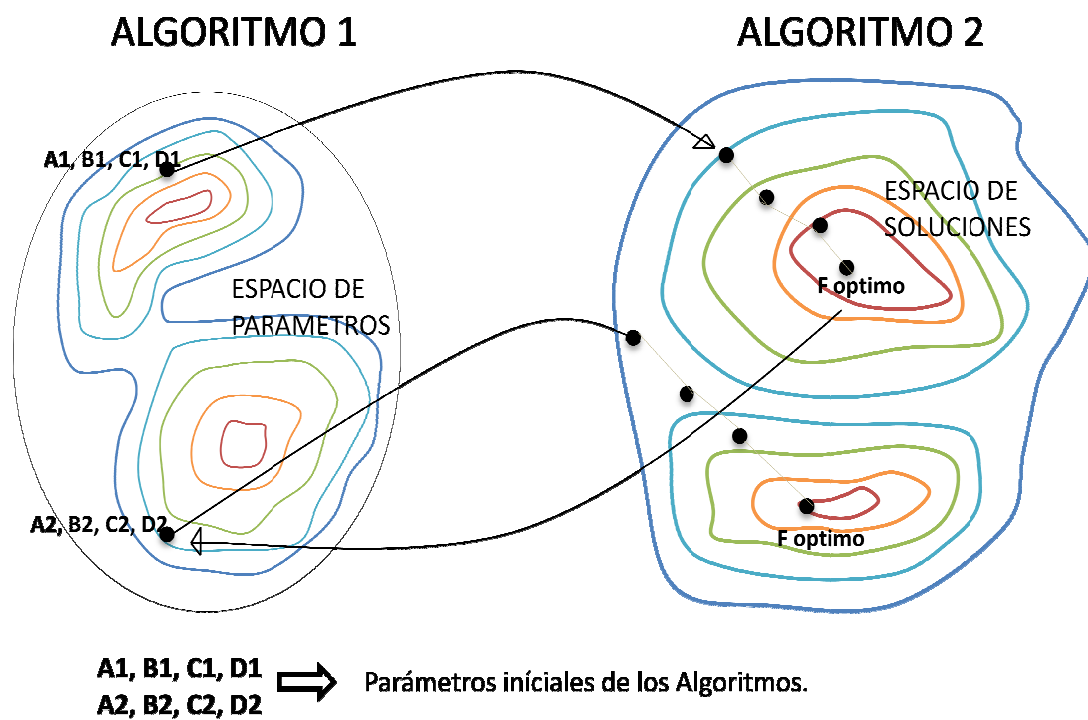
Los algoritmos de evolución paramétrica [41], son una nueva tendencia en el proceso de hibridación de algoritmos heurísticos. Este tipo de híbridos se fundamenta en realizar un proceso de optimización, sobre un algoritmo optimizador, esto quiere decir que un algoritmo puro evaluara como función otro algoritmo de la misma clase, el cual a su vez se encuentra evaluando la función objetivo propuesta.

La validez de esta clase de algoritmos híbridos radica en la capacidad de extender el rango de búsqueda. Cuando un algoritmo puro realiza el proceso de optimización este depende, en primera instancia, de los parámetros propuestos inicialmente, los cuales varían dependiendo del algoritmo. Esto hace que dependan en gran medida del carácter heurístico y estocástico que poseen.

Al realizar un proceso de evolución paramétrica lo que se busca es, variar el conjunto de parámetros que posee el algoritmo que evaluará la función, pero esta variación se encuentra dirigida por un algoritmo el cual buscará los valores de los parámetros casi óptimos, para de esta forma, llegar hasta el valor óptimo de la función.

Todo este proceso tiene como objetivo realizar un proceso de optimización sin depender de los parámetros iniciales de búsqueda y tener una mayor probabilidad de éxito. Esto se puede lograr sin tener un vasto conocimiento del algoritmo, puesto que este probará una gran cantidad de opciones. La figura 5 muestra el esquema conceptual de los algoritmos estudiados en este trabajo.

Figura 5: Esquema de un Algoritmo de Evolución Paramétrica.



Fuente: Los Autores.

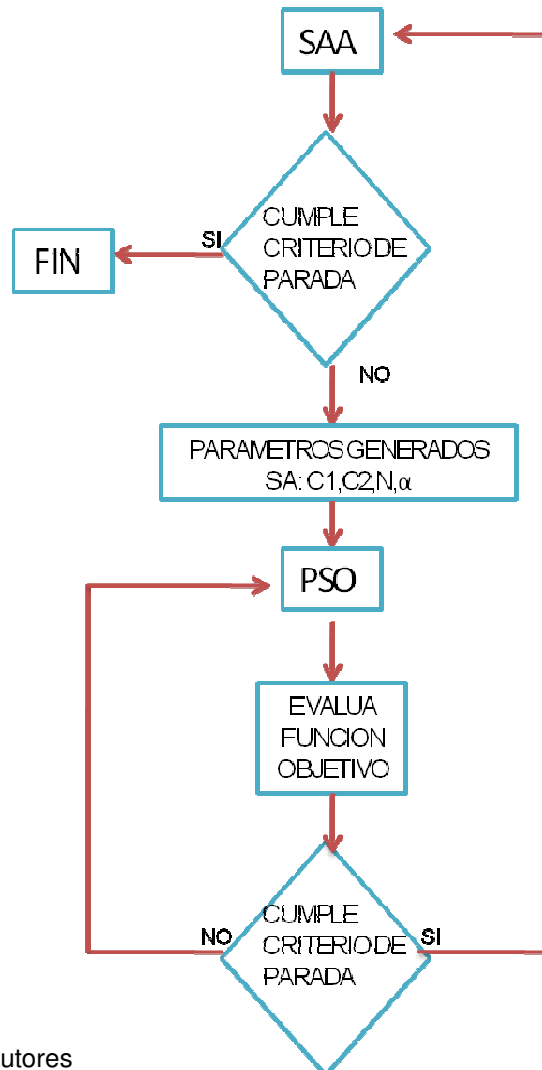
El funcionamiento básico de un algoritmo híbrido de evolución paramétrica se basa en la generación de los parámetros del algoritmo 2 utilizando el algoritmo 1, teniendo en cuenta que el algoritmo 2 devolverá el valor óptimo de la función objetivo y entorno de este serán optimizados los parámetros del algoritmo 2. De

esta forma, se espera obtener el mínimo de la función objetivo, con mayor precisión.

2.3.1. ALGORITMO HIBRIDO SAA-PSO

El fundamento del algoritmo híbrido es muy simple, se puede decir que cada algoritmo está trabajando por separado evaluando cada uno una función diferente y cada uno realiza el mismo proceso, como si se encontrara en forma pura, puesto que en este caso el PSO se encuentra evaluando la función objetivo, teniendo unos parámetros iniciales, pero a su vez el SA se encuentra evaluando al PSO como función cuyo valor objetivo es el valor óptimo del PSO, de esta manera el SA se encuentra optimizando el PSO por medio de sus parámetros. A continuación se presenta el funcionamiento del SAA-PSO. El criterio de parada se estableció igual al propuesto para el SAA (ver pag.67), teniendo como segunda alternativa un límite de 100 evaluaciones para todas exceptuando Rosenbrock 50 en el cual se realizaron 1000 evaluaciones.

Figura 6: Diagrama de Flujo SAA-PSO



Fuente: Los Autores

2.3.2. ALGORITMO HIBRIDO PSO-SAA

Podemos decir que este algoritmo de evolución paramétrica es similar al anterior, solo que se invierten los papeles en la evaluación de la función, teniendo en cuenta que el algoritmo que evaluará la función objetivo será el SAA y el algoritmo que optimizará los parámetros del SAA será el PSO. Como en el algoritmo anterior

la parada se establecerá con los mismos criterios de la parada del PSO. Los parámetros del PSO fueron:

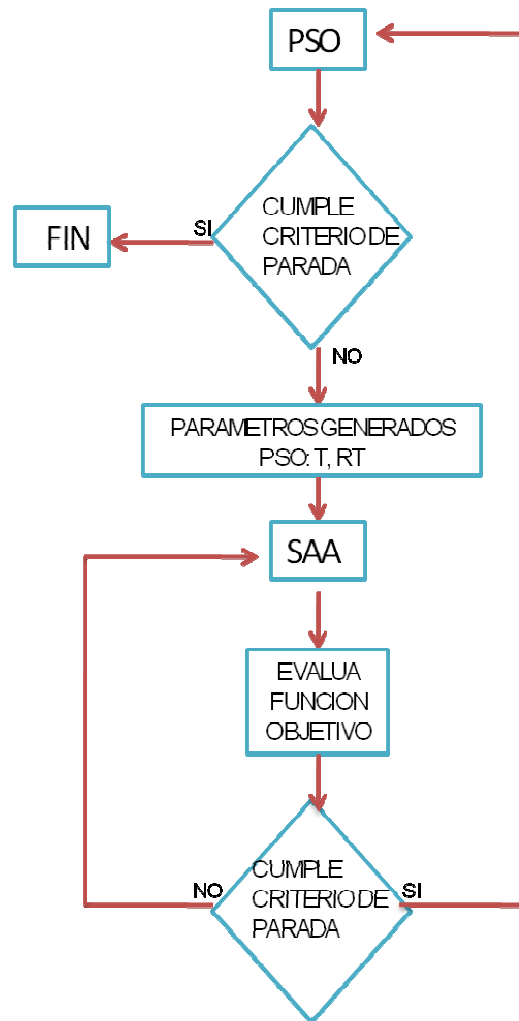
Figura 7: Diagrama de Flujo PSO-SAA

W=1.4

C1=2.0

C2=2.0

N=3



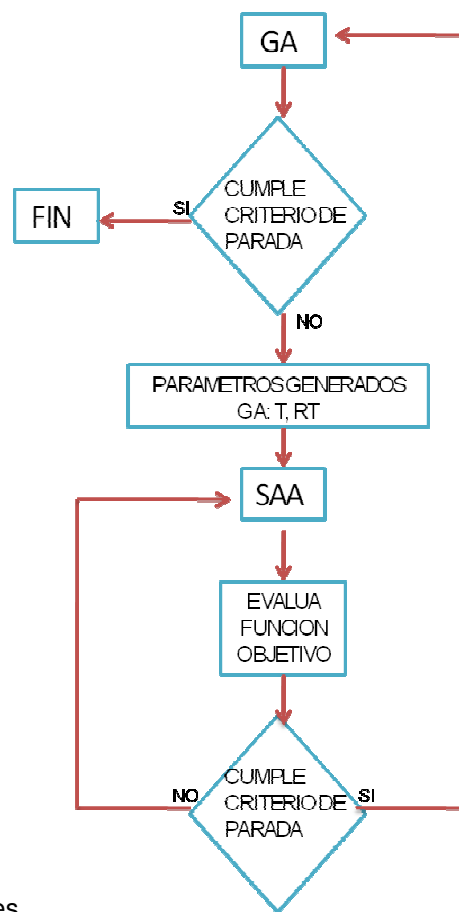
Fuente: Los Autores

2.3.3. ALGORITMO HIBRIDO GA-SAA.

En este algoritmo planteamos la misma plataforma del PSO-SAA puesto que en primera instancia se establece un algoritmo de amplio rango como lo son AG o PSO y a su vez evaluamos, la función con el SA. Como se ha expresado

anteriormente, el AG enviará valores de temperatura (T) y del factor de reducción de temperatura (RT) y el SA evaluará la función con estos parámetros y devolverá el valor de la función al AG y de esta forma se realizara el ciclo el cual terminara con un número máximo de generaciones. Los parámetros básicos del AG fueron 3 individuos, una probabilidad de mutación del 1%, una probabilidad de cruce del 50%, y se limitaron las generaciones a 10.

Figura 8: Diagrama de Flujo GA-SAA



Fuente: Los Autores

3. EVALUACION DE LOS ALGORITMOS DE OPTIMIZACION Y ALGORITMOS DE EVOLUCION PARAMETRICA CON FUNCIONES DE LA LITERATURA INTERNACIONAL.

3.1. INTRODUCCION

El potencial y las limitaciones tanto de los algoritmos de optimización como de los algoritmos de evolución paramétrica son presentados en esta investigación y se muestran por su aplicación en las funciones de prueba las cuales se encuentran en la literatura internacional.

Este proceso es realizado con el fin de catalogar sus virtudes y desventajas frente a los diversos planteamientos y de esta manera poseer un conocimiento inicial de su comportamiento.

Para poder realizar estas evaluaciones primero se partió por programar los algoritmos en un lenguaje de programación, el lenguaje escogido fue FORTRAN, teniendo en cuenta y aprovechando el uso en la comunidad científica de este. De esta manera, se programaron los tres algoritmos inicialmente para posteriormente realizar el proceso de hibridización, y tener las herramientas necesarias para realizar el proceso de evaluación y con ello un proceso de análisis.

3.2. MEDIDAS DE DESEMPEÑO

Teniendo en cuenta que los algoritmos estudiados en esta investigación son de naturaleza estocástica, los criterios para evaluar su desempeño se describen a continuación. La desviación estándar de las funciones analíticas se utilizó para medir la precisión y la estabilidad de los métodos. Un método heurístico de

optimización se dice que es estable si su desviación estándar es baja. Se cataloga precisión en el método cuando, la diferencia entre la media de las pruebas y el valor óptimo analítico es pequeño. El algoritmo puede ser descrito como robusto, cuando se prueba en diferentes problemas, y ofrezca precisión y estabilidad. Definimos que un algoritmo presenta confianza (n/m) cuando logra determinar un número de veces n el valor óptimo, en una cantidad de m pruebas, tomando como referencia la primera cifra decimal. Estimamos el tiempo empleado por los algoritmos como el promedio de la duración de las evaluaciones realizadas por estos, dicho tiempo será expresado en segundos. Cabe recordar que para estas evaluaciones se utilizó un computador de escritorio con las siguientes características: procesador AMD Sempron 3000, 1.81 GHz y 1 Gb de Ram.

3.3. DESCRIPCION DE ENSAYOS

Con el fin de evaluar el desempeño de los algoritmos se realizaron 6 pruebas (test o funciones de prueba), dichas funciones se extrajeron de la literatura internacional [38] y son presentadas a continuación. Para realizar el proceso de evaluación de los algoritmos se realizaron 10 evaluaciones de cada función para los algoritmos puros y 5 evaluaciones para los algoritmos híbridos, teniendo en cuenta la naturaleza estocástica de los mismos. De esta manera es posible realizar una comparación del desempeño de los algoritmos con relación a las medidas descritas anteriormente.

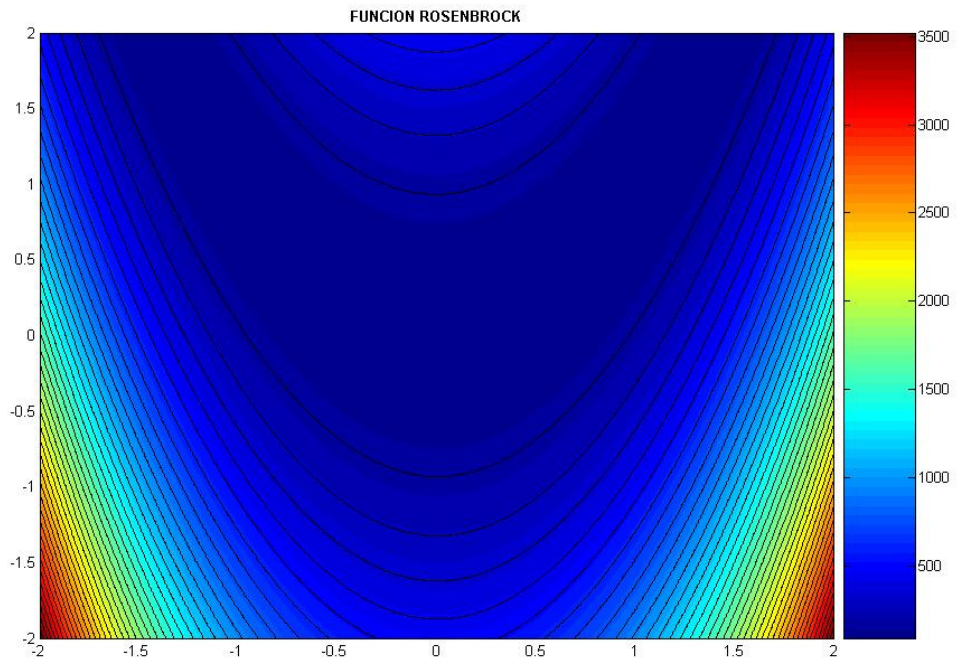
Las funciones que se utilizaron para evaluar el desempeño de los algoritmos se presentan a continuación [38]:

Función de Rosenbrock 2D y 50D

$$F(x) = \sum_{i=1}^{N-1} [100(X_{i+1} - X_i^2)^2 + (X_i - 1)^2],$$

$$\text{Límites } [-2,2] \quad , \quad F_{xopt}(1, \dots, 1) = 0$$

Figura 9: Funcion Rosenbrock 2D



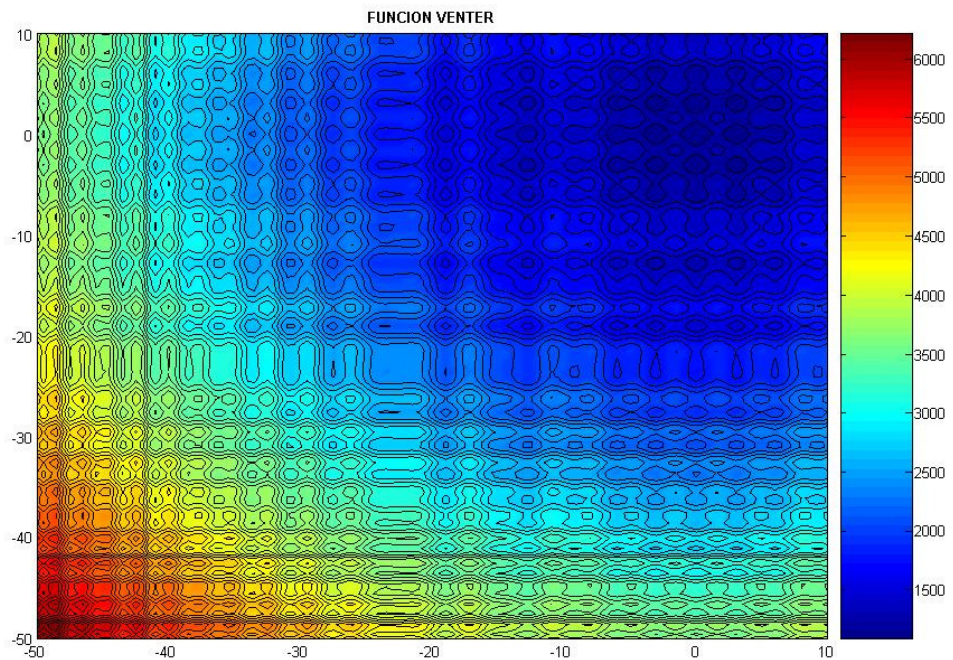
Fuente: Los Autores

Función de Venter 2D

$$F(X_1, X_2) = X_1^2 - 100\text{Cos}(X_1) - 100\text{Cos}(X_1^2/30) + X_2^2 - 100\text{Cos}(X_2)^2 - 100\text{Cos}(X_2^2/30) + 1400,$$

$$\text{Límites } [-10, 50] \quad , \quad F_x(0,0) = 1000$$

Figura 10: Función de Venter 2D



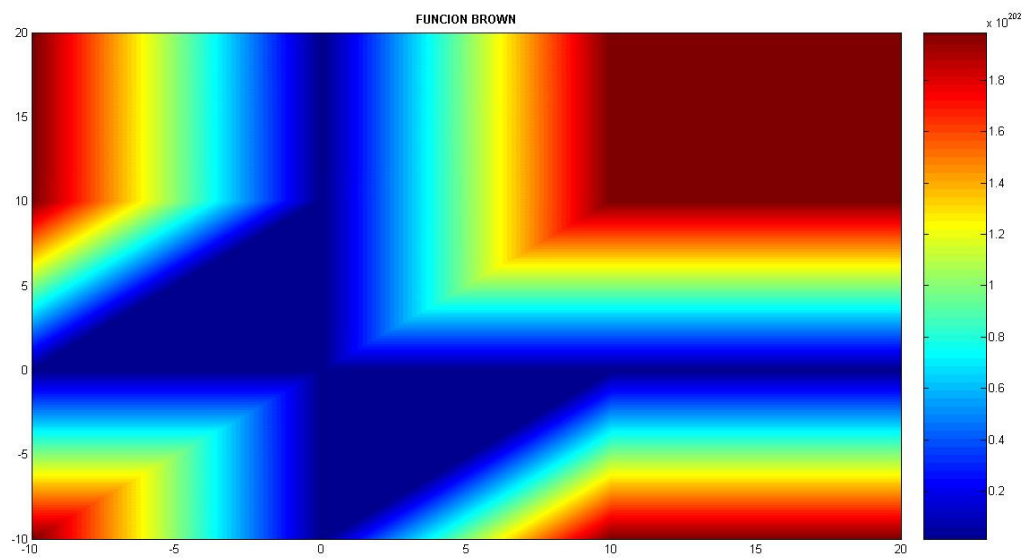
Fuente: Los Autores

Función de Brown 20D

$$F(x) = \sum_{i=1}^{19} (X_i^2)^{(x_{i+1}^2+1)} + (X_{i+1}^2)^{(x_i^2+1)},$$

$$\text{Límites } [-10,10] \quad , \quad F_x(0, \dots, 0) = 0$$

Figura 11: Función de Brown 20D



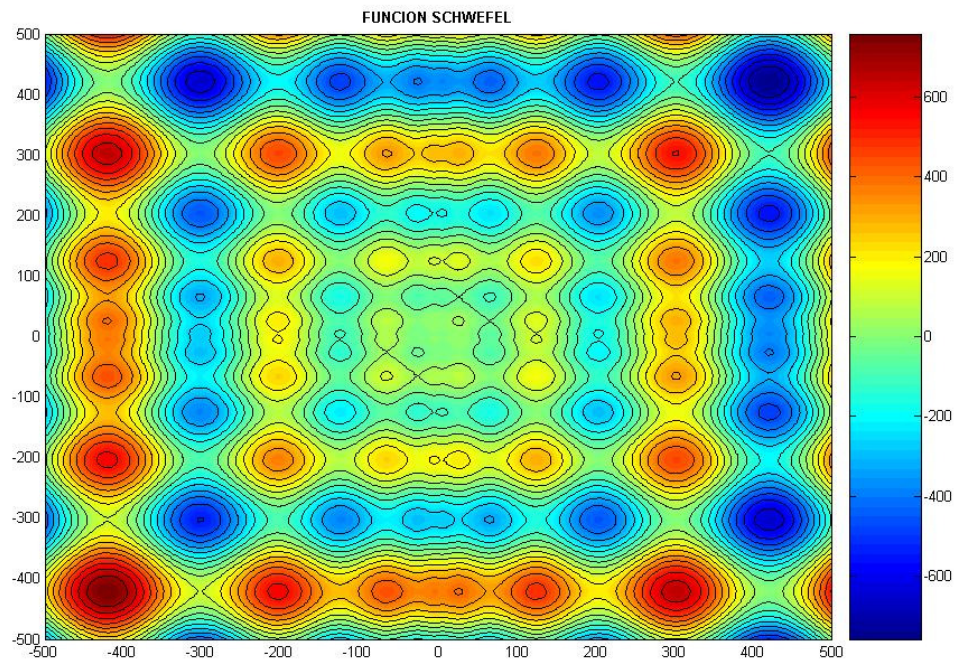
Fuente: Los Autores

Función Schwefel 5D

$$F(x) = \sum_{i=1}^N -X_i \sin(\sqrt{|X_i|}),$$

Limites $[-500,500]$, $F_x(420.968, \dots, 420.968) = -2094.914$

Figura 12: Función de Schwefel 2D



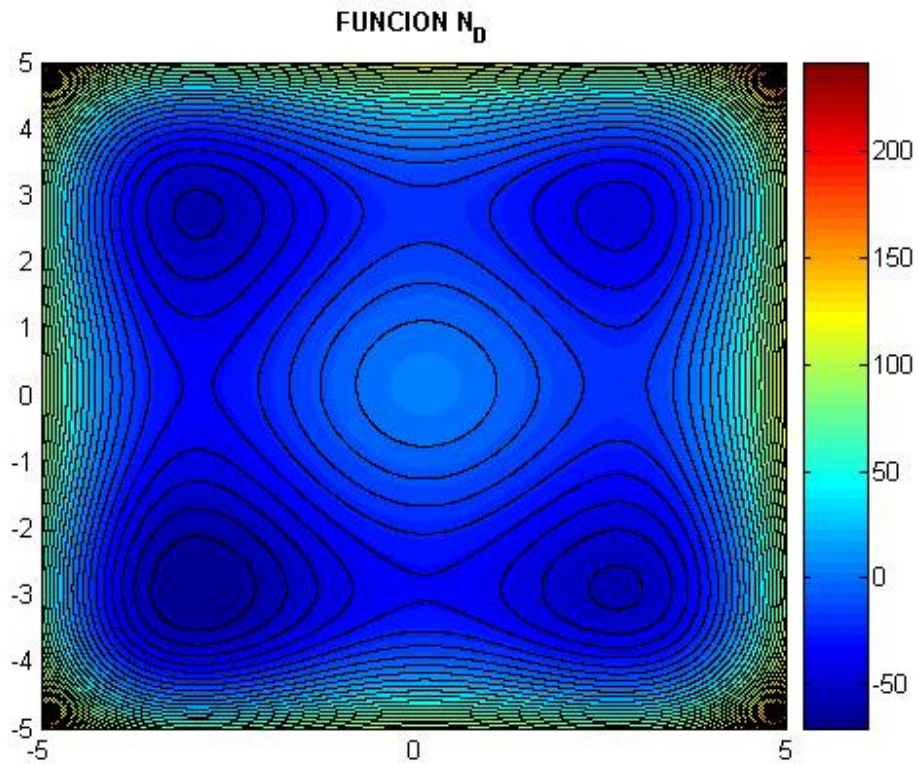
Fuente: Los Autores

N-dimensional Test Function 5D

$$F(x) = \frac{1}{N} \sum_{i=1}^N (X_i^4 - 16X_i^2 + 5X_i),$$

Limites $[-5,5]$, $F_x(-2.9, \dots, -2.9) = -78.332$

Figura 13: Función de N-Dimensional 2D



Fuente: Los Autores

A continuación, se presentan los resultados de la valoración de desempeño de los algoritmos analizados en este trabajo frente a las funciones descritas anteriormente.

3.4. ALGORITMOS GENETICOS (AG)

La metodología utilizada fue la propuesta por David Carroll. Para evaluar el desempeño de un AG con las funciones de prueba, se debe tener en cuenta que este presenta un tamaño de población de 5 individuos, una probabilidad de mutación del 1%, una probabilidad de cruce del 50%, y se limitaron las generaciones a 100. Con estos parámetros se realizaron las pruebas. La programación establecida para los algoritmos genéticos no posee un criterio de parada específico, su parada se produce cuando llega a un límite de generaciones. Los resultados de las pruebas realizadas se muestran en las tablas 1-6.

Tabla 1: Función de Rosenbrock 2D (AG)

FUNCION DE ROSENBROCK 2D		
Prueba	Foptimo	Iteraciones
1	0.001	66000
2	0.000	67000
3	0.000	50500
4	0.000	79000
5	0.000	48500
6	0.000	101500
7	0.000	87000
8	0.000	126000
9	0.000	88500
10	0.000	99500

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 2: Función de Rosenbrock 50D (AG)

FUNCION DE ROSENBROCK 50D		
Prueba	Foptimo	Iteraciones
1	0.026	147500
2	0.050	139000
3	0.075	118500
4	0.100	110000
5	0.125	150000
6	0.150	150000
7	0.175	150000
8	0.200	150000
9	0.225	150000
10	0.250	150000

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 3: Función de Venter 2D (AG)

FUNCION DE VENTER 2D		
Prueba	Foptimo	Iteraciones
1	1000.000	45500
2	1000.000	51500
3	1000.000	13500
4	1000.000	7500
5	1000.000	37500
6	1000.000	25500
7	1000.000	62000
8	1000.000	15000
9	1000.000	29500
10	1000.000	49500

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 1000.00

Tabla 4: Función de Brown 20D (AG)

FUNCION DE BROWN 20D		
Prueba	Foptimo	Iteraciones
1	0.015	123500
2	0.027	138500
3	0.041	145500
4	0.055	105000
5	0.068	114000
6	0.082	136500
7	0.096	141000
8	0.109	150000
9	0.124	150000
10	0.137	150000

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 5: Función de Schwefel 5D (AG)

FUNCION DE SCHWEFEL 5D		
Prueba	Foptimo	Iteraciones
1	3.760	150000
2	-181.830	150000
3	-184.680	150000
4	-184.624	150000
5	-184.246	150000
6	-184.523	150000
7	.186.325	150000
8	-185.791	150000
9	-184.246	150000
10	-182.745	150000

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = -2094.914

Tabla 6: N-dimensional Test Function 5D (AG)

N-DIMENSIONAL TEST FUNCTION 5D		
Prueba	Foptimo	Iteraciones
1	-20.000	150000
2	-18.061	150000
3	-20.000	150000
4	-20.000	150000
5	-18.524	150000
6	-20.000	150000
7	-18.330	150000
8	-20.000	150000
9	-20.000	150000
10	-20.000	150000

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = -78.33

Se presenta la tabla 7 correspondiente al Algoritmo Genético (AG) en la cual se puede observar el desempeño que tuvo en las diversas funciones de prueba.

Tabla 7: Tabla Resumen AG

		ALGORITMO GENETICO					
		ROSENBROCK 2D	ROSENBROCK 50D	VENTER 2D	BROWN 20D	SCHWEFEL 5D	NDIMENSIONAL
VALOR DE LA PROMEDIO	OPTIMO FUNCION	0.000	0.138	1000.000	0.075	-163.214	-19.492
DESVIACION ESTANDAR		0.000	0.076	0.000	0.041	62.626	0.826
CONFIABILIDAD DE ÉXITO		100%	30%	100%	70%	0%	0%
VALOR PROMEDIO DEL NUMERO DE EVALUACIONES DE LA FUNCION		81350	141500	33700	135400	150000	150000
TIEMPO PROMEDIO DE EJECUCION		4.00seg	415.00seg	3.00seg	294.00seg	30.00seg	35.00seg

Fuente: Los Autores

3.5. OPTIMIZACION CON ENJAMBRE DE PARTICULAS (PSO)

La metodología utilizada fue la propuesta por Kennedy. Los parámetros que rigen este algoritmo se asumieron diferentes para cada función puesto que de ellos depende el éxito del algoritmo, los valores empleados se muestran en las Tablas 8-13. El algoritmo se detiene cuando la desviación estándar de los últimos 10 valores es igual a una tolerancia establecida que para la prueba fue de 1×10^{-7} . Los resultados obtenidos se encuentran a continuación en las tablas 8-13.

Tabla 8: Función de Rosenbrock 2D (PSO)

FUNCION DE ROSENBROCK 2D		
Prueba	Foptimo	Iteraciones
1	0.000	1660
2	0.000	1420
3	0.000	1740
4	0.000	1580
5	0.000	1700
6	0.000	1340
7	0.000	1540
8	0.000	1600
9	0.000	1500
10	0.000	1500
C1	2	
C2	2	
W	1.4	
N	20	

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo)= 0.00

Tabla 9: Función de Rosenbrock 50D (PSO)

FUNCION DE ROSENBROCK 50D		
Prueba	Foptimo	Iteraciones
1	43.929	209500
2	48.746	232000
3	46.002	195000
4	45.853	121500
5	49.164	228000
6	46.747	181000
7	47.497	263500
8	48.510	131500
9	48.963	272000
10	46.981	198000
C1	2	
C2	2	
W	1.4	
N	500	

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 10: Función de Venter 2D (PSO)

FUNCION DE VENTER 2D		
Prueba	Foptimo	Iteraciones
1	1000.000	4100
2	1000.000	1600
3	1000.000	2360
4	1000.000	2940
5	1000.000	1780
6	1000.000	2520
7	1000.000	1960
8	1015.023	17080
9	1000.000	1640
10	1000.000	3220
C1	1.35	
C2	1.35	
W	1.15	
N	20	

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 1000.00

Tabla 11: Función de Brown 20D (PSO)

FUNCION DE BROWN 20D		
Prueba	Foptimo	Iteraciones
1	0.000	35600
2	0.000	30600
3	0.001	31200
4	0.000	35800
5	0.000	34400
6	0.000	41800
7	0.011	38000
8	0.001	30200
9	0.000	36800
10	0.001	42200
C1	2.184	
C2	1.539	
W	0.83	
N	200	

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 12: Función de Schwefel 5D (PSO)

FUNCION DE SCHWEFEL 5D		
Prueba	Foptimo	Iteraciones
1	-2094.914	7350
2	-2094.914	6975
3	-1976.476	7425
4	-1976.476	6450
5	-1976.476	6375
6	-2094.914	7725
7	-2094.914	6825
8	-1976.476	7200
9	-1976.475	8475
10	-1976.470	9300
C1	1.75	
C2	1.75	
W	1.4	
N	75	

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = -2094.914

Tabla 13: N-dimensional Test Function 5D (PSO)

N-DIMENSIONAL TEST FUNCTION 5D		
Prueba	Foptimo	Iteraciones
1	-78.332	6675
2	-72.678	5700
3	-78.332	8550
4	-78.332	7125
5	-78.332	6675
6	-78.332	7575
7	-72.678	6450
8	-78.332	6375
9	-72.678	6975
10	-78.332	6525

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = -78.33

A continuación se muestra la tabla 14, que corresponde al PSO. En ella se puede apreciar el comportamiento que tuvo el algoritmo frente a las diversas funciones de prueba.

Tabla 14: Tabla Resumen PSO

		OPTIMIZACION CON ENJAMBRE DE PARTICULAS					
		ROSENBROCK 2D	ROSENBROCK 50D	VENTER 2D	BROWN 20D	SCHWEFEL 5D	NDIMENSIONAL
VALOR DE LA PROMEDIO	OPTIMO FUNCION	0.000	47.239	1001.502	0.002	-2023.851	-76.636
DESVIACION ESTANDAR		0.000	1.678	4.751	0.003	61.162	2.731
CONFIABILIDAD DE ÉXITO		100%	0%	90%	100%	40%	70%
VALOR PROMEDIO DEL NUMERO DE EVALUACIONES DE LA FUNCION		1558	203200	3920	35660	7410	6862
TIEMPO PROMEDIO DE EJECUCION		1.00seg	49.00seg	2.00seg	4.00seg	15.00seg	5.00seg

Fuente: Los Autores

3.6. SIMULATED ANNEALING (SA)

La metodología utilizada fue la propuesta por Corona. En el proceso de ejecución de este algoritmo sus parámetros básicos son la temperatura (T) y el factor de reducción de la temperatura (RT), se estableció una temperatura inicial de $T=100$, exceptuando la prueba de Brown 20D en la cual se utilizó una $T=250$, el factor de reducción usado en todas las pruebas fue $RT=0.85$. El algoritmo realizará la parada cuando la diferencia de los 2 últimos valores sea igual a una tolerancia que para el caso era igual 1×10^{-7} . Los resultados obtenidos se encuentran a continuación en las tablas 15-20.

Tabla 15: Función de Rosenbrock 2D (SA)

FUNCION DE ROSENBROCK 2D		
Prueba	Foptimo	Iteraciones
1	0.000	160001
2	0.000	132001
3	0.000	164001
4	0.000	140001
5	0.000	148001
6	0.000	140001
7	0.000	132001
8	0.000	112001
9	0.000	120001
10	0.000	172001

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 16: Función de Rosenbrock 50D (SA)

FUNCION DE ROSENBROCK 50D		
Prueba	Foptimo	Iteraciones
1	0.051	6800001
2	0.055	6800001
3	0.058	6600001
4	0.041	6800001
5	0.036	6800001
6	0.044	6800001
7	0.040	6900001
8	0.064	6600001
9	0.050	6800001
10	0.043	6900001

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 17: Función de Venter 2D (SA)

FUNCION DE VENTER 2D		
Prueba	Foptimo	Iteraciones
1	1000.000	156001
2	1000.000	148001
3	1000.000	156001
4	1000.000	152001
5	1000.000	148001
6	1000.000	152001
7	1000.000	144001
8	1000.000	160001
9	1000.000	144001
10	1000.000	140001

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 1000.00

Tabla 18: Función de Brown 20D (SA)

FUNCION DE BROWN 20D		
Prueba	Foptimo	Iteraciones
1	0.002	3120001
2	0.001	3160001
3	0.002	3160001
4	0.002	3120001
5	0.002	3120001
6	0.001	3280001
7	0.002	3080001
8	0.003	3080001
9	0.001	3120001
10	0.002	3160001

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 19: Función de Schwefel 5D (SA)

FUNCION DE SCHWEFEL 5D		
Prueba	Foptimo	Iteraciones
1	-2094.897	380001
2	-2094.913	440001
3	-2094.900	400001
4	-2094.910	430001
5	-2094.900	420001
6	-2094.900	430001
7	-2094.913	460001
8	-2094.905	440001
9	-2094.885	400001
10	-2094.909	440001

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = -2094.914

Tabla 20: N-dimensional Test Function 5D (SA)

N-DIMENSIONAL TEST FUNCTION 5D		
Prueba	Foptimo	Iteraciones
1	-78.327	420001
2	-78.324	420001
3	-78.322	390001
4	-78.326	450001
5	-78.320	440001
6	-78.327	420001
7	-78.328	470001
8	-78.325	440001
9	-78.322	410001
10	-78.329	460001

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = -78.33

En la tabla 21 se puede apreciar el desempeño que tuvo el SA en las diferentes funciones prueba

Tabla 21: Tabla Resumen SA

		SIMULATED ANNEALING					
		ROSENBROCK 2D	ROSENBROCK 50D	VENTER 2D	BROWN 20D	SCHWEFEL 5D	NDIMENSIONAL
VALOR DE LA PROMEDIO	OPTIMO FUNCION	0.000	0.048	1000.000	0.002	-2094.903	-78.325
DESVIACION ESTANDAR		0.000	0.009	0.000	0.001	0.009	0.003
CONFIABILIDAD DE ÉXITO		100%	100%	100%	100%	90%	100%
VALOR PROMEDIO DEL NUMERO DE EVALUACIONES DE LA FUNCION		142001	6780001	150001	3140001	424001	432001
TIEMPO PROMEDIO DE EJECUCION		1.00seg	37.00seg	1.00seg	33.00seg	2.00seg	2.00seg

Fuente: Los Autores

3.7. ALGORITMO HIBRIDO SAA-PSO

Los resultado de la evaluación de desempeño para este hibrido se presentan a continuación en las tablas 22-27. Los parámetros para los algoritmos fueron C1, C2, N y Alfa, limitados por los siguientes valores:

$$0.5 < C1 < 3.0$$

$$0.5 < C2 < 3.0$$

$$100 < N < 200$$

$$0.75 < \text{Alfa} < 1.0$$

Tabla 22: Función de Rosenbrock 2D (SAA-PSO)

FUNCION DE ROSENBROCK 2D						
Prueba	Foptimo	Iteraciones	C1	C2	N	Alfa
1	0.000	155800	1.671	2.725	83	0.972
2	0.000	155800	1.219	1.611	90	0.952
3	0.000	155800	1.859	0.839	97	0.862
4	0.000	155800	1.634	1.083	61	0.892
5	0.000	155800	1.450	1.133	68	0.845

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 23: Función de Rosenbrock 50D (SAA-PSO)

FUNCION DE ROSENBROCK 50D						
Prueba	Foptimo	Iteraciones	C1	C2	N	Alfa
1	0.000	203200000	0.503	2.198	147	0.906
2	0.000	203200000	0.575	2.231	133	0.827
3	0.000	203200000	0.628	2.124	82	0.903
4	0.000	203200000	0.623	2.272	153	0.946
5	0.000	203200000	0.603	2.246	194	0.823

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 24: Función de Venter 2D (SAA-PSO)

FUNCION DE VENTER 2D						
Prueba	Foptimo	Iteraciones	C1	C2	N	Alfa
1	1000.000	392000	1.036	2.017	95	0.947
2	1000.000	392000	1.911	2.959	97	0.960
3	1000.000	392000	1.465	1.510	52	0.928
4	1000.000	392000	1.939	2.269	53	0.974
5	1000.000	392000	0.593	2.101	72	0.932

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 1000.00

Tabla 25: Función de Brown 20D (SAA-PSO)

FUNCION DE BROWN 20D						
Prueba	Foptimo	Iteraciones	C1	C2	N	Alfa
1	0.000	3566000	0.970	1.891	61	0.887
2	0.000	3566000	0.835	2.090	68	0.826
3	0.000	3566000	0.968	1.945	97	0.918
4	0.000	3566000	1.180	1.912	60	0.928
5	0.000	3566000	0.941	1.965	76	0.875

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 26: Función de Schwefel 5D (SAA-PSO)

FUNCION DE SCHWEFEL 5D						
Prueba	Foptimo	Iteraciones	C1	C2	N	Alfa
1	-2094.907	741000	1.308	2.382	72	0.941
2	-2094.914	741000	1.413	1.279	67	0.980
3	-2094.914	741000	1.412	1.662	79	0.938
4	-2094.914	741000	1.051	1.223	59	0.978
5	-2094.914	741000	1.470	1.958	80	0.918

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = -2094.914

Tabla 27: N-dimensional Test Function (SAA-PSO)

N-DIMENSIONAL TEST FUNCTION 5D						
Prueba	Foptimo	Iteraciones	C1	C2	N	Alfa
1	-78.332	682000	1.704	1.529	81	0.927
2	-78.332	682000	0.946	2.010	63	0.970
3	-78.332	682000	1.341	1.424	79	0.977
4	-78.332	682000	0.595	1.475	79	0.871
5	-78.332	682000	1.599	1.425	87	0.918

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = -78.33

Tabla 28: Tabla Resumen SAA-PSO

		SAA-PSO					
		ROSENBROCK 2D	ROSENBROCK 50D	VENTER 2D	BROWN 20D	SCHWEFEL 5D	NDIMENSIONAL
VALOR DE LA PROMEDIO	OPTIMO DE LA FUNCION	0.000	0.000	1000.000	0.000	-2094.913	-78.332
DESVIACION ESTANDAR		0.000	0.000	0.000	0.000	0.003	0.000
CONFIABILIDAD DE ÉXITO		100%	100%	100%	100%	100%	100%
VALOR DEL PROMEDIO DE EVALUACIONES DE LA FUNCION	NUMERO DE LA FUNCION	155800	203200000	392000	3566000	741000	682000
TIEMPO PROMEDIO DE EJECUCION		14.00seg	590.00seg	12.00seg	311.00seg	25.00seg	15.00seg

Fuente: Los Autores

3.8. ALGORITMO HIBRIDO PSO-SAA

Los resultado de la evaluación de desempeño para este hibrido se presentan a continuación en las tablas 29-34. Los parámetros para los agoritmos fueron RT y T.

Tabla 29: Función de Rosenbrock 2D (PSO-SAA)

FUNCION DE ROSENBROCK 2D				
Prueba	F optimo	Iteraciones	RT	T
1	0.000	852006	0.883	309.392
2	0.000	852006	0.889	393.765
3	0.000	852006	0.840	420.245
4	0.000	852006	0.838	320.481
5	0.000	852006	0.891	408.348

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 30: Función de Rosenbrock 50D (PSO-SAA)

FUNCION DE ROSENBROCK 50D				
Prueba	F optimo	Iteraciones	RT	T
1	0.003	40680006	0.832	338.247
2	0.090	40680006	0.808	383.493
3	0.088	40680006	0.797	290.774
4	0.067	40680006	0.831	331.496
5	0.030	40680006	0.824	391.521

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 31: Función de Venter 2D (PSO-SAA)

FUNCION DE VENTER 2D				
Prueba	F optimo	Iteraciones	RT	T
1	1000.000	900006	0.926	301.126
2	1000.000	900006	0.846	300.810
3	1000.000	900006	0.819	320.351
4	1000.000	900006	0.833	384.978
5	1000.000	900006	0.794	368.564

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 1000.00

Tabla 32: Función de Brown 20D (PSO-SAA)

FUNCION DE BROWN 20D				
Prueba	F optimo	Iteraciones	RT	T
1	0.000	18840006	0.340	477.841
2	0.000	18840006	0.280	394.088
3	0.000	18840006	0.429	500.000
4	0.000	18840006	0.307	483.553
5	0.000	18840006	0.421	375.292

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 33: Función de Schwefel 5D (PSO-SAA)

FUNCION DE SCHWEFEL 5D				
Prueba	F optimo	Iteraciones	RT	T
1	-2094.914	2544006	0.483	484.557
2	-2094.914	2544006	0.941	390.195
3	-2094.914	2544006	0.657	500.000
4	-2094.914	2544006	0.484	495.733
5	-2094.914	2544006	0.567	381.380

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = -2094.914

Tabla 34: N-dimensional Test Function 5D (PSO-SAA)

N-DIMENSIONAL TEST FUNCTION 5D				
Prueba	F optimo	Iteraciones	RT	T
1	-78.332	2592006	0.474	500.000
2	-78.332	2592006	0.646	336.502
3	-78.332	2592006	0.566	322.546
4	-78.332	2592006	0.492	448.031
5	-78.332	2592006	0.353	314.623

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = -78.33

Tabla 35: Tabla Resumen PSO-SAA

		PSO-SAA					
		ROSENBROCK 2D	ROSENBROCK 50D	VENTER 2D	BROWN 20D	SCHWEFEL 5D	NDIMENSIONAL
VALOR DE LA PROMEDIO	OPTIMO DE LA FUNCION	0.000	0.230	1000.000	0.000	-2094.914	-78.332
DESVIACION ESTANDAR		0.000	0.000	0.000	0.000	0.000	0.000
CONFIABILIDAD DE ÉXITO		100%	100%	100%	100%	100%	100%
VALOR PROMEDIO DEL NUMERO DE EVALUACIONES DE LA FUNCION		852006	40680006	900006	18840006	2544006	2592006
TIEMPO PROMEDIO DE EJECUCION		20.00seg	715.00seg	30.00seg	480.00seg	68.00seg	42.00seg

Fuente: Los Autores

3.9. ALGORITMO HIBRIDO GA-SAA

Los resultado de la evaluación de desempeño para este hibrido se presentan a continuación en las tablas 36-41. Los parámetros para los agoritmos fueron RT y T.

Tabla 36: Función de Rosenbrock 2D (GA-SAA)

FUNCION DE ROSENBROCK 2D				
Prueba	F optimo	Iteraciones	RT	T
1	0.000	852006	0.061	0.000
2	0.000	852006	0.408	0.000
3	0.000	852006	0.502	0.000
4	0.000	852006	0.447	0.000
5	0.000	852006	0.160	0.000

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 37: Función de Rosenbrock 50D (GA-SAA)

FUNCION DE ROSENBROCK 50D				
Prueba	F optimo	Iteraciones	RT	T
1	0.000	40680006	0.539	0.000
2	0.000	40680006	0.146	0.000
3	0.000	40680006	0.399	0.000
4	0.000	40680006	0.455	0.000
5	0.000	40680006	0.423	0.000

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 38: Función de Venter 2D (GA-SAA)

FUNCION DE VENTER 2D				
Prueba	F optimo	Iteraciones	RT	T
1	1000.000	900006	0.269	0.000
2	1000.000	900006	0.097	0.000
3	1000.000	900006	0.222	0.000
4	1000.000	900006	0.462	0.000
5	1000.000	900006	0.024	0.000

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) =1000.00

Tabla 39: Función de Brown 20D (GA-SAA)

FUNCION DE BROWN 20D				
Prueba	F optimo	Iteraciones	RT	T
1	0.000	18840006	0.087	0.001
2	0.000	18840006	0.539	0.000
3	0.000	18840006	0.271	0.000
4	0.000	18840006	0.428	0.000
5	0.000	18840006	0.396	0.000

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Tabla 40: Función de Schwefel 5D (GA-SAA)

FUNCION DE SCHWEFEL 5D				
Prueba	F optimo	Iteraciones	RT	T
1	-2094.899	19080045	0.542	2.263
2	-2094.899	5088012	0.406	0.172
3	-2094.899	10176024	0.239	0.002
4	-2094.899	2544006	0.875	0.002
5	-2094.899	2544006	0.321	0.024

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = -2094.914

Tabla 41: N-dimensional Test Function 5D (GA-SAA)

N-DIMENSIONAL TEST FUNCTION 5D				
Prueba	F optimo	Iteraciones	RT	T
1	-78.333	2592006	0.239	0.002
2	-78.333	2592006	0.195	0.000
3	-78.333	2592006	0.065	0.000
4	-78.333	2592006	0.182	0.000
5	-78.333	2592006	0.168	0.000

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = -78.33

Tabla 42: Tabla Resumen GA-SAA

		GA-SAA					
		ROSENBROCK 2D	ROSENBROCK 50D	VENTER 2D	BROWN 20D	SCHWEFEL 5D	NDIMENSIONAL
VALOR DE LA PROMEDIO	OPTIMO DE LA FUNCION	0.000	0.000	1000.000	0.000	-2094.899	-78.333
DESVIACION ESTANDAR		0.000	0.000	0.000	0.000	0.000	0.000
CONFIABILIDAD DE ÉXITO		100%	100%	100%	100%	100%	100%
VALOR PROMEDIO DEL NUMERO DE EVALUACIONES DE LA FUNCION		852006	40680006	900006	18840006	7886419	2592006
TIEMPO PROMEDIO DE EJECUCION		32.00seg	305.00seg	25.00seg	124.00seg	80.00seg	62.00seg

Fuente: Los Autores

4. ANALISIS E INTERPRETACION DE DATOS

4.1. RESULTADOS

Tabla 43: Tabla General Función de Rosenbrock 2D

	FUNCION DE ROSENBROCK 2D					
	AG	PSO	SA	SAA-PSO	PSO-SAA	GA-SAA
VALOR OPTIMO DE LA FUNCION PROMEDIO	0.000	0.000	0.000	0.000	0.000	0.000
DESVIACION ESTANDAR	0.000	0.000	0.000	0.000	0.000	0.000
CONFIABILIDAD DE ÉXITO	100%	100%	100%	100%	100%	100%
VALOR PROMEDIO DEL NUMERO DE EVALUACIONES DE LA FUNCION	81350	1558	142001	155800	852006	852006
TIEMPO PROMEDIO DE EJECUCION	4.00seg	1.00seg	1.00seg	14.00seg	20.00seg	32.00seg

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

En un primer vistazo podemos observar el incremento en las iteraciones en los algoritmos híbridos, lo cual como se ha comentado anteriormente es algo de esperar. En cuanto a la precisión, todos los algoritmos llegaron a su objetivo obteniendo desviaciones estándar muy bajas.

Después de un arduo proceso de evaluación se puede decir que esta función no se dificulta para ningún algoritmo.

Tabla 44: Tabla General Función de Rosenbrock 50D

		FUNCION DE ROSENBROCK 50D					
		AG	PSO	SA	SAA-PSO	PSO-SAA	GA-SAA
VALOR OPTIMO DE LA FUNCION PROMEDIO		0.138	47.239	0.048	0.000	0.056	0.000
DESVIACION ESTANDAR		0.076	1.678	0.009	0.000	0.038	0.000
CONFIABILIDAD DE ÉXITO		30%	0%	100%	100%	100%	100%
VALOR PROMEDIO DEL NUMERO DE EVALUACIONES DE LA FUNCION		141500	203200	6780001	203200000	40680006	40680006
TIEMPO PROMEDIO DE EJECUCION		415.00seg	49.00seg	37.00seg	590.00seg	715.00seg	305.00seg

Fuente: **Los Autores**

Valor Fobjetivo óptimo (mínimo) = 0.00

Teniendo en cuenta lo observado en la tabla anterior, se aprecia el aumento en la complejidad de la función, puesto que la capacidad de los algoritmos puros (AG, PSO) se ve limitada. Para este caso, resaltamos la eficiencia de los algoritmos híbridos de evolución paramétrica ya que sus medidas de desempeño son superiores al compararlas con las mostradas por los algoritmos puros.

En el caso particular del SAA-PSO se propuso un número elevado de iteraciones, con el fin de obtener el mejor resultado de la función, el cual el PSO no pudo hallar de manera individual.

Tabla 45: Tabla General Función de Venter 2D

		FUNCION DE VENTER 2D					
		AG	PSO	SA	SAA-PSO	PSO-SAA	GA-SAA
VALOR OPTIMO DE LA FUNCION PROMEDIO		1000.000	1001.502	1000.000	1000.000	1000.000	1000.000
DESVIACION ESTANDAR		0.000	4.751	0.000	0.000	0.000	0.000
CONFIABILIDAD DE ÉXITO		100%	90%	100%	100%	100%	100%
VALOR PROMEDIO DEL NUMERO DE EVALUACIONES DE LA FUNCION		33700	3920	150001	392000	900006	900006
TIEMPO PROMEDIO DE EJECUCION		3.00seg	2.00seg	1.00seg	12.00seg	30.00seg	25.00seg

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo)= 1000.00

La función de Venter en términos generales es una función de fácil manejo, por ende el comportamiento en general de todos los algoritmos es bastante bueno encontrando una falencia en el PSO, la cual afecta su confiabilidad y se ve expresado en la desviación estándar.

Tabla 46: Tabla General Función de Brown 20D

		FUNCION DE BROWN 20D					
		AG	PSO	SA	SAA-PSO	PSO-SAA	GA-SAA
VALOR OPTIMO DE LA FUNCION PROMEDIO		0.075	0.002	0.002	0.000	0.000	0.000
DESVIACION ESTANDAR		0.041	0.003	0.001	0.000	0.000	0.000
CONFIABILIDAD DE ÉXITO		70%	100%	100%	100%	100%	100%
VALOR PROMEDIO DEL NUMERO DE EVALUACIONES DE LA FUNCION		135400	35660	3140001	3566000	18840006	18840006
TIEMPO PROMEDIO DE EJECUCION		294.00seg	4.00seg	33.00seg	311.00seg	480.00seg	124.00seg

Fuente: Los Autores

Valor Fobjetivo óptimo (mínimo) = 0.00

Esta función presenta un mayor nivel de complejidad, pero en términos generales se tiene altos niveles de confiabilidad y precisión, es bastante notorio el avance de los híbridos en comparación con las técnicas puras, teniendo en cuenta que estos no pararon por criterios específicos sino por un número máximo de evaluaciones.

El comportamiento de los híbridos en todas las medidas de desempeño es excelente, y demuestra su ventaja sobre las técnicas puras.

Tabla 47: Tabla General Función de Schwefel 5D

		FUNCION DE SCHWEFEL 5D					
		AG	PSO	SA	SAA-PSO	PSO-SAA	GA-SAA
VALOR OPTIMO DE LA FUNCION PROMEDIO		-163.214	-2023.851	-2094.903	-2094.913	-2094.914	-2094.899
DESVIACION ESTANDAR		62.626	61.162	0.009	0.003	0.000	0.000
CONFIABILIDAD DE ÉXITO		0%	40%	90%	100%	100%	100%
VALOR PROMEDIO DEL NUMERO DE EVALUACIONES DE LA FUNCION		150000	7410	424001	741000	2544006	7886419
TIEMPO PROMEDIO DE EJECUCION		30.00seg	15.00seg	2.00seg	25.00seg	68.00seg	80.00seg

Fuente: **Los Autores**

Valor Fobjetivo óptimo (mínimo) = -2094.914

En esta función las técnicas puras presentan falencias al momento de realizar la búsqueda (AG, PSO), por otro lado, los algoritmos híbridos muestran excelentes resultados, como se ven en la tabla 47.

Tabla 48: Tabla General N-dimensional Test Function 5D

		N-DIMENSIONAL TEST FUNCTION 5D					
		AG	PSO	SA	SAA-PSO	PSO-SAA	GA-SAA
VALOR OPTIMO DE LA FUNCION PROMEDIO		-19.492	-76.636	-78.325	-78.332	-78.332	-78.333
DESVIACION ESTANDAR		0.826	2.731	0.003	0.000	0.000	0.000
CONFIABILIDAD DE ÉXITO		0%	70%	100%	100%	100%	100%
VALOR PROMEDIO DEL NUMERO DE EVALUACIONES DE LA FUNCION		150000	6862	432001	682000	2592006	2592006
TIEMPO PROMEDIO DE EJECUCION		35.00seg	5.00seg	2.00seg	15.00seg	42.00seg	62.00seg

Fuente: **Los Autores**

Valor Fobjetivo óptimo (mínimo) = -78.33

Como última función de prueba encontramos una función de gran dificultad la cual afecto a los algoritmos puros limitando su desempeño, pero a su vez se observa nuevamente un buen comportamiento para los algoritmos híbridos de evolución paramétrica.

4.2. ANALISIS

En cuanto a los algoritmos puros, se puede decir que, el PSO lo podemos considerar estable mas no preciso ni robusto debido que en la mayoría de las funciones no mostró confiabilidad para llegar al optimo global, el SA se puede considerar como estable, preciso y robusto debido que en todas las funciones encontró el valor óptimo. En cuanto al AG este muestra precisión y confiabilidad, pero cuando el nivel de complejidad de la función objetivo se incrementa y a su vez aumenta el rango este no logra llegar a su objetivo.

Simulated Annealing es un algoritmo de gran precisión y confiabilidad pero a su vez genera un alto costo computacional. Al realizar el proceso de hibridación es de esperar que aumente su costo, pero al generarse una mayor interacción y a su vez al tener un mayor rango de acción el PSO-SAA genera un gasto similar al de la técnica pura (como se puede ver en las tablas 46-48). Esto representa una mejora sustancial puesto que la evaluación del SA se realizo utilizando parámetro óptimos que facilitaron el proceso de búsqueda.

El algoritmo genético presenta un buen desempeño al ser evaluado en rangos pequeños, pero al extenderse el rango la búsqueda el algoritmo no logra converger lo cual genera soluciones erróneas. Al elaborar el Hibrido GA-SAA, se pudo apreciar que al unir las bondades del SA, con la capacidad de búsqueda global del AG, estaríamos abarcando un amplio espacio de posibles conjuntos de parámetros. Además de esto, reseñamos que el costo computacional de este algoritmo es alto, pero esto se ve recompensado con su precisión y confiabilidad.

El hibrido SAA-PSO posee la desventaja que el algoritmo que esta realizando la búsqueda de optimo es el PSO el cual no mostró gran desempeño, pero al ponerle el SA como optimizador de parámetros, en conjunto logran llegar a optimo generando un costo computacional mas alto que los anteriores.

Al observar el comportamiento en general de los algoritmos de evolución paramétrica podemos concluir que estos son excelentes siempre y cuando se los deje trabajar libremente sin restringir las iteraciones a realizar, puesto que ellos pueden demorarse más pero existe una alta probabilidad de encontrar el valor óptimo.

Haciendo una comparación entre los algoritmos puros y los algoritmos híbridos (de evolución paramétrica) se puede decir que los algoritmos híbridos son más precisos y confiables ya que en todas las funciones llegan al valor óptimo.

CONCLUSIONES

- Es de resaltar que los algoritmos híbridos (de evolución paramétrica) generan un mayor costo computacional y por ende un mayor tiempo de trabajo. Esto se debe a que dichos algoritmos realizan una mayor cantidad de iteraciones en comparación con los algoritmos puros.
- Los Algoritmos Híbridos presentan una gran ventaja con respecto a los algoritmos puros puesto que a estos no hay que ingresarle valores de los parámetros, debido a que estos valores serán optimizados en el proceso.
- Para manejar los algoritmos híbridos propuestos, no se necesita ser experto en el uso de estos, solo basta con ingresar la función a optimizar y dejar que el algoritmo haga su trabajo.
- En problemas de optimización con múltiples variables, soluciones locales, discontinuidades y otras múltiples limitaciones que complican la obtención de una solución global, los algoritmos híbridos manejan una alta precisión y confiabilidad.
- El desempeño de los algoritmos, tanto híbridos como puros, en gran medida están ligados al nivel de mejoras que se realice a su forma de búsqueda y por ende a la programación de los mismos.
- Debemos tener en cuenta que las pruebas realizadas a estos algoritmos son de muy alta complejidad (ver Rosenbrock 50D y Brown 20D), y por lo mismo los algoritmos se exigen al máximo, es de esperar que en problemas más sencillos su costo computacional disminuya.

RECOMENDACIONES

- Optimizar el proceso de programación con el fin de obtener mejoras en el funcionamiento.
- Utilizar procesamiento en paralelo con el fin de alcanzar mejores resultados y de mas alta precisión en menos tiempo.
- Aplicar los algoritmos híbridos (de evolución paramétrica) a problemas de la Ingeniería Civil aprovechando sus cualidades.
- Profundizar en la investigación de los algoritmos propuestos teniendo en cuenta los resultados obtenidos.
- Realizar diversas combinaciones entre los algoritmos puros con el fin de crear una mayor base de datos y poder escoger el mejor dependiendo de las exigencias del problema.

BIBLIOGRAFIA

[1] Goldberg, D.E., "Genetic Algorithm in search, optimization and machine learning. Adisson-Wesley Publishing, Reading, USA, 1989.

[2] Kirkpatrick, S., Gelatt, A., Vecchi, M.P., "Optimization by simulated annealing" Science, 220, 1983

[3] Kennedy, J., Eberhart, R., "Particle Swarm Optimization" Proc. IEEE Int. Conf. Neural Networks, 4, 1995.

[4] O. Begambre, J.E Laier, "Damage Detection Using the Simulated Annealing Algorithm", Proceedings of McMat2005: 2005 Joint ASME/ASCE/SES Conference on Mechanics and Materials June 1 – 3, 2005, Baton Rouge, Louisiana, USA.

[5] B. Mitra, S. Jha, P.P. Choudhurl, "A simulated annealing based state assignment approach for control synthesis", *Proceedings of the IEEE International Symposium on VLSI Design*, New Delhi (India), Enero 1991.

[6] R.V. Kacelenga, P.J. Graumann, L.E. Turner, "Design of digital filters using simulated annealing", *Proceedings of the IEEE International Symposium on Circuits and Systems*, New Orleans (USA), Mayo 1990.

[7] F. Ares, S.R. Rengarajan, E. Villanueva, E. Skochinski, E. Moreno, "Application of genetic algorithms and simulated annealing technique in optimizing the aperture distributions of antenna arrays", *Proceedings of the IEEE Antennas and Propagation Society International Symposium*, Baltimore (USA), Julio 1996, Vol. 2.

- [8] Y. Rahmat-Samii, E. Michielssen, "Electromagnetic optimization by genetic algorithms", John Wiley & Sons, New York, 1999.
- [9] J.M. Johnson, Y. Rahmat-Samii, "Genetic algorithm optimization of wireless communication networks", *Proceedings of the IEEE Antennas and Propagation Society International Symposium*, California (USA), Junio 1995, Vol. 4.
- [10] V. Petridis, S. Kazarlis, A. Papaikonomou, "A genetic algorithm for training recurrent neural networks", *Proceedings of the International Joint Conference on Neural Networks*, Nagoya (Japan), Octubre 1993.
- [11] M.M.S. Al-Sawafi, J.A. Jervase, "A micro-genetic algorithm-based CDMA multi-user detector", *Proceedings of the Conference on Communication Networks and Services Research*, Fredericton (Canada), Mayo 2004.
- [12] J. Tippayachai, W. Ongsakul, I. Ngamroo, "Parallel micro genetic algorithm for constrained economic dispatch", *IEEE Transactions on Power Systems*, Vol. 17, No. 3, Agosto 2002.
- [13] J. Robinson, Y. Rahmat-Samii, "Particle swarm optimization in electromagnetics", *IEEE Transactions on Antennas and Propagation*, Vol. 52, No. 2, Febrero 2004.
- [14] R.C. Eberhart, X. Hu, "Human tremor analysis using particle swarm optimization", *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*, Washington (USA), Julio 1999, Vol. 3.
- [15] Z. Yangyang, J.I. Chunlin, Y. Ping, L.I. Manlin, W. Chaojin, W. Guangxing, "Particle swarm optimization for base station placement in mobile communication",

Proceedings of the 2004 IEEE International Conference on Networking, Sensing & Control, Taipei (Taiwan), Marzo 2004.

[16] De Castilho, Vanessa Cristina, “Otimização de componentes de Concreto Pré-moldado Protendidos mediante Algoritmos Genéticos”, Tesis de Doctorado, Universidade de São Paulo, Febrero de 2003.

[17] J.R. Regué, M. Ribó, J.M. Garrell, “Radiated emissions conversion from anechoic environment to OATS using a hybrid genetic algorithm-gradient method”, *Proceedings of the IEEE International Symposium on Electromagnetic Compatibility*, Montreal (Canada), Agosto 2001, Vol. 1.

[18] J.R. Pérez, J. Basterrechea, “Near to far-field transformation for antenna measurements using a GA based method”, *Proceedings of the IEEE Antennas and Propagation Society International Symposium*, Texas (USA), Junio 2002, Vol. 1.

[19] Rong-Song He, Shun-Fa Hwang, “Damage detection by an adaptive real-parametersimulated annealing genetic algorithm” Elsevier, 2006.

[20] M.M.S. Al-Sawafi, J.A. Jervase, “A micro-genetic algorithm-based CDMA multi-user detector”, *Proceedings of the Conference on Communication Networks and Services Research*, Fredericton (Canada), May 2004.

[21] Z. Feizhou, Y. Dongkai, C. Xiuwan, “Intelligent scheduling of public traffic vehicles based on hybrid genetic algorithm”, *Proceedings of the IEEE Intelligent Transportation Systems*, Shanghai (China), October 2003.

[22] J.M. Johnson, Y. Rahmat-Samii, “Genetic algorithms in engineering electromagnetics”, *IEEE Antennas and Propagation Magazine*, Vol. 39, No. 4, Agosto 1997.

- [23] D. Dumitrescu, B. Lazzerini, L.C. Jain, A. Dumitrescu, "Evolutionary Computation", CRC Press LLC, Florida, 2000.
- [24] D.L. Carroll, "Fortran genetic algorithm (GA) driver", CU Aerospace. (<http://cuaerospace.com/carroll/ga.html>).
- [25] C.R. Houck, J.A. Joines, M.G. Kay, "A genetic algorithm for function optimization: a matlab implementation", *Technical Report NCSU-IE-TR-95- 09*, North Carolina State University. Octubre 1995.
- [26] D. Dumitrescu, B. Lazzerini, L.C. Jain, A. Dumitrescu, "Evolutionary Computation", CRC Press LLC, Florida, 2000.
- [27] O. Begambre, J.E Laier, Structural Damage Identification Using Particle Swarm Optimization, Proceedings of the Fifth International Conference on Engineering Computational Technology, B.H.V.Topping, G.Montero and R.Montenegro,(Editors),Civil-CompPress,Stirlingshire,Scotland,1-2.
- [28] E.C. Laskari, K.E. Parsopoulos, M.N. Vrahatis, "Particle swarm optimization for minimax problems", *Proceedings of the 2002 Congress on Evolutionary Computation-CEC02*, Honolulu (USA), Mayo 2002.
- [29] D. Srinivasan, W.H. Loo, R.L. Cheu, "Traffic incident detection using particle swarm optimization", *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, Indianapolis (USA), April 2003.
- [30] J.S. Vesterstrøm, J. Riget, "Particle swarms: Extensions for improved local, multi-modal, and dynamic search in numerical optimization", *Master's Thesis*, University of Aarhus, Mayo 2002.

- [31] V. Miranda, N. Fonseca, "EPSO – Evolutionary particle swarm optimization, a new algorithm with applications in power systems", *IEEE/PES Transmission and Distribution Conference and Exhibition*, Porto Alegre (Brazil), Octubre 2002.
- [32] M.F. Tasgetiren, Y-C. Liang, "A binary particle swarm optimization algorithm for lot sizing problem", *Journal of Economic and Social Research*, Vol. 5,2003.
- [33] N. Franken, A.P. Engelbrecht, "PSO approaches to co-evolve IPD strategies", *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, Oregon (USA), June 2004.
- [34] R.C. Eberhart, Y. Shi, "Particle swarm optimization: developments, applications and resources", *Proceedings of the 2001 Congress on Evolutionary Computation*, Seoul (South Korea), Mayo 2001.
- [35] A.E. Mercado, C.R Seña, "El PSOS (Particle Swarm Optimization + Simplex) en la optimización del calentamiento de un dieléctrico con microondas", Tesis de Pre-grado, Universidad Industrial de Santander, 2008.
- [36] D.W. Boeringer, D.H. Werner, "Particle swarm optimization versus Genetic algorithms for phased array synthesis", *IEEE Transactions on Antennas and Propagation*, Vol. 52, No. 3, March 2004.
- [37] Metropolis, N., Rosenbluth, A., Rosenbluth, M.,Teller, A.,Teller, E., "Equations of state calculations by fast computing machines " *J. of Chem. Physics*. 21,1953.
- [38] Georgieva A., Jordanov I., *European Journal of Operational Research*. 2009.

[39] Ling-lai Li, Ling Wang, Li-heng Liu “An effective hybrid PSOSA strategy for optimization and its application to parameter estimation” Elsevier, 2006.

[40] Zhang Yinghui, Rao Yunbo, Zhou Mingtian “GASA Hybrid Algorithm Applied in Airline Crew Rostering System” TSINGHUA SCIENCE AND TECHNOLOGY ISSN 1007-0214 46/49 pp255-259 Volume 12, Number S1, Julio 2007.

[41] O. Begambre, J.E. Laier “A hybrid Particle Swarm Optimization – Simplex algorithm (PSOS) for structural damage identification” Elsevier, 2009.