UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

EL OPTIMIZADOR DE ENJAMBRE DE PARTÍCULAS Y SU APLICACIÓN EN SISTEMAS ELÉCTRICOS DE POTENCIA

TRABAJO MONOGRÁFICO
PARA OBTENER EL GRADO DE

INGENIERO EN SISTEMAS DE ENERGÍA

PRESENTA
JUAN JOSÉ CAMBRANIS LÓPEZ

ASESORES
DR. VÍCTOR MANUEL SÁNCHEZ HUERTA
M.C. EMMANUEL TORRES MONTALVO
M.E.S. ROBERTO ACOSTA OLEA

CHETUMAL QUINTANA ROO, MÉXICO, NOVIEMBRE DE 2014

UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

TRABAJO MONOGRÁFICO BAJO LA SUPERVISIÓN DEL COMITÉ DEL PROGRAMA DE LICENCIATURA Y APROBADA COMO REQUISITO PARA OBTENER EL GRADO DE:

INGENIERO EN SISTEMAS DE ENERGÍA

COMITÉ DE TRABAJO MONOGRÁFICO

SUPERVISOR: ______________________
DR. VÍCTOR MANUEL SÁNCHEZ HUERTA

SUPERVISOR: ______________________
M.C. EMMANUEL TORRES MONTALVO

SUPERVISOR: ______________________
M.E.S. ROBERTO ACOSTA OLEA

UNIVERSIDAD DE QUINTANA ROO
DCI DIVISIÓN DE CIENCIAS E INGENIERÍA

UNIVERSIDAD DE QUINTANA ROO
SERVICIOS ESCOLARES
TITULACIONES

CHETUMAL QUINTANA ROO, MÉXICO, NOVIEMBRE DE 2014

Agradecimientos

A MIS ABUELOS

Por haberme educado como a un hijo y estar en todo momento a mi lado, por enseñarme el camino correcto para ser hoy un hombre honesto. Hoy más que nunca los tengo presentes aunque ya no estén conmigo.

A MI MAMÁ

Por apoyarme siempre, aunque por momentos la vida fuera difícil, nunca dejo de estar pendiente de mí. Es a quien le debo toda mi formación en la vida. Gracias.

A MIS MAESTROS

Por haber compartido sus conocimientos conmigo, procurando despertar en mí el interés por la ciencia. En especial al Doctor Víctor Manuel Sánchez Huerta, por aconsejarme a lo largo de mi formación universitaria.

Resumen

En los sistemas eléctricos de potencia existen muchos problemas de optimización que se les presenta a los ingenieros en las etapas de diseño, operación o mantenimiento, tales como problemas de flujo óptimo de potencias en una red eléctrica, despacho económico de energía, programación de mantenimiento, identificación de modelos, predicción y control de carga eléctrica, dimensionamiento de sistemas de alimentación, por mencionar algunos de ellos. Para la solución de estos problemas de optimización existen diferentes técnicas como programación no-lineal, programación cuadrática, programación lineal, método de Monte Carlo; sin embargo estas técnicas requieren de poder establecer las ecuaciones matemáticas que rigen la dinámica del problema. Una alternativa a esta problemática es el utilizar algoritmos de computación evolutiva en los problemas de optimización ya que estos no requieren del modelado matemático completo de las relaciones de las variables que rigen el comportamiento del problema. Uno de los algoritmos de la computación evolutiva que ha estado ganando gran interés de aplicación en problemas de optimización en las últimas dos décadas es el optimizador de enjambre de partículas (Particle Swarm Optimization, PSO por sus siglas en inglés). En este trabajo de monografía se describe el principio básico del optimizador de enjambre de partículas con el objetivo de proporcionar las bases de este algoritmo evolutivo. Además se presenta una revisión de la aplicación del PSO en problemas de optimización en Sistemas Eléctricos de Potencia.

ÍNDICE

ÍNDICE DE FIGURAS

CAPÍTULO 1

CAPÍTULO 1

1.1 Definición de un problema de optimización

Los problemas de optimización son muy comunes que se presentan en diferentes áreas del conocimiento como la ingeniería, las finanzas, el transporte, por mencionar algunos ejemplos. A lo largo de la historia del hombre, la mente humana ha tenido que enfrentar problemas de optimización y ha realizado grandes esfuerzos para solucionarlos.

De manera general, podemos decir que la optimización es el proceso de encontrar la mejor forma de utilizar los recursos disponibles, a la vez que no se dejan de cumplir algunas de las restricciones que son impuestas en la solución encontrada. De una forma más detallada, podemos decir que el proceso de optimización consiste en definir matemáticamente un sistema, identificar sus variables y las condiciones que deben satisfacer, definir las propiedades del sistema y entonces buscar los valores de las variables del sistema que le proporcionan las propiedades más deseables.

En los problemas de optimización, se tienen que encontrar las soluciones óptimas o cercanas a las óptimas con respecto a ciertos objetivos a cumplir. Generalmente, la solución óptima de un problema no se puede encontrar en un solo paso, sino que se debe seguir todo un proceso que sirva de guía en la solución del problema. Frecuentemente, el proceso de solución es dividido en diferentes etapas las cuáles son ejecutadas secuencialmente. De esta forma, el proceso de solución de un problema de optimización se puede describir mediante un algoritmo o diagrama de flujo.

Para la solución de un problema de optimización existen diferentes alternativas de solución, en dónde el usuario o diseñador debe decidir una de ellas. La selección del proceso de solución tiene impacto sobre el usuario o usuarios afectado por el problema a resolver. Por lo que en los problemas de optimización los usuarios están interesados en seleccionar la alternativa que maximice o minimice una función objetivo la cual es definida de acuerdo al criterio de evaluación seleccionado.

Usualmente, en un problema de optimización no todas las alternativas de solución son factibles de realizar o implementar, ya que existen restricciones que limitan el número de soluciones factibles. Estas restricciones comúnmente están relacionadas con leyes, limitaciones técnicas o físicas, económicas, por mencionar algunos ejemplos.

1.1.1 Elementos en un problema de optimización.

Un problema de optimización está constituido por los siguientes elementos:

- La función objetivo, la cual representa la cantidad a ser optimizada, esto es la cantidad a ser minimizada o maximizada. En algunos problemas de optimización, no se define una función objetivo explícitamente, sino que el objetivo del problema es encontrar una solución que satisfaga a un conjunto de restricciones.
- El conjunto de variables o incógnitas que inciden en el valor de la función objetivo. Si x representa el conjunto de las incógnitas del problema, entonces $f(x)$ cuantifica la calidad de la solución candidata, x.
- El conjunto de restricciones, el cual restringe los valores que pueden ser asignados a las incógnitas del problema. En la mayoría de los problemas de optimización se define al menos un conjunto de restricciones límite, el cuál define el dominio de los valores para cada variable. La complejidad de las restricciones a cumplir pueden limitar el conjunto de soluciones factibles a un problema.

De esta forma, el objetivo de un método de optimización es asignar valores, dentro de un dominio permitido, a las incógnitas del problema de tal forma que la función objetivo es optimizada y todas las restricciones son cumplidas. Para lograr este objetivo, el algoritmo de optimización busca una solución dentro de un espacio de búsqueda, S, de soluciones candidatas. En el caso de problemas con restricciones, su solución debe estar en el espacio factible de soluciones, de tal forma que $F \subseteq S$.

1.1.2 Clasificación de los problemas de optimización.

Los problemas de optimización se pueden clasificar con base a las siguientes características

- El número de variables que influyen en la función objetivo. Un problema con sólo una variable a ser optimizado se le considera como un problema univariable. Por otra parte, si existe más de una variable que influye en la función objetivo a este problema se le considera como un problema multivariable.
- El tipo de variables. Las variables en un problema de optimización pueden ser continuos, enteros o una combinación de estos. De allí que los problemas de optimización pueden ser referidos como problemas de optimización continuos, enteros o mixtos.
- El grado de no-linealidad de la función objetivo. Los problemas de optimización se consideran lineales cuando la función objetivo es lineal con respecto a sus variables. Si la función objetivo es cuadrática entonces el problema de optimización es cuadrático. De esta forma, cuando la función objetivo utilizada es no-lineal, el problema de optimización es referido como un problema de optimización no lineal.
- Las restricciones utilizadas. Un problema que sólo emplea restricciones límite se le conoce como un problema sin restricciones. Los problemas restringidos son aquellos que tienen igualdades o desigualdades en sus restricciones.
- El número de óptimos. Cuando existe una única solución al problema de optimización, el problema se considera unimodal. Por otra parte, si existe más de un óptimo, el problema de optimización es considerado multimodal.
- El número de criterios a optimizar. Si la cantidad a ser optimizada es expresada usando sólo una función objetivo, el problema es considerado como un problema mono-objetivo. Por su parte, un problema de optimización multi-objetivo es aquel en que se deben optimizar más de una función objetivo simultáneamente.

1.1.3 Tipos de óptimos en un problema de optimización.

Las soluciones encontradas por un método de optimización se clasifican con base en la calidad de la solución. La calidad de la solución se refiere a que tan buena es la solución encontrada por el método de optimización. Para ejemplificar la calidad de una solución consideremos un problema de minimización a optimizar. Asumamos que el conjunto de soluciones al problema de optimización tiene un comportamiento como el que se muestra en la Figura 1.1.

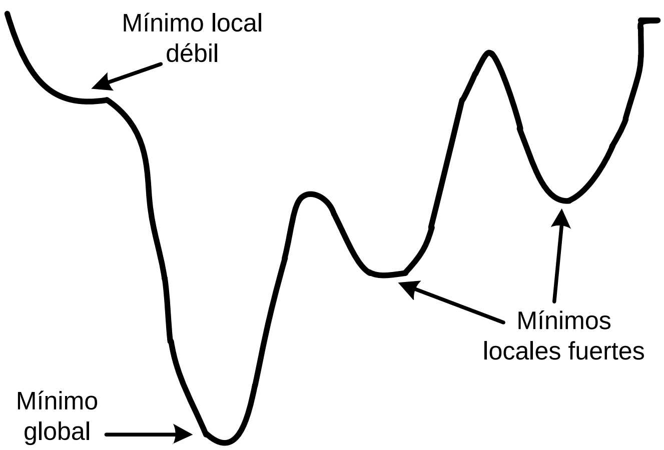


Figura 1.1 Tipos de óptimos en un problema de optimización.

Cómo se observa en la Figura 1.1, el espacio de soluciones al problema de minimización tiene una única solución óptima que corresponde al mínimo global. Sin embargo, existen otros mínimos que pueden hacer "creer" al método de optimización que han encontrado la solución al problema, debido a que en la vecindad de estos mínimos locales el valor de la solución es mayor con respecto a este punto; sobre todo en los mínimos locales fuertes que pueden "atrapar" al método de optimización y dejar que este siga explorando en el espacio de soluciones y encontrar la solución óptima al problema de minimización.

1.1.4 Tipos de métodos de optimización.

Un algoritmo de optimización busca la solución óptima de forma iterativa transformando la solución candidata actual en una probable mejor solución. Los métodos de optimización se pueden clasificar en dos categorías con base en el tipo de solución que encuentran. La primer categoría corresponde a los algoritmos de búsqueda local los cuales sólo usan información local del espacio de búsqueda alrededor de la solución actual para producir una nieva solución. Debido a que sólo emplean información local, los algoritmos de búsqueda local sólo encuentran mínimos locales, pudiéndose dar el caso que el mínimo local encontrado sea el mínimo global al problema. Por otra parte los algoritmos de búsqueda global usan mayor información del espacio de búsqueda para localizar el mínimo global. Se podría decir que los algoritmos de búsqueda global exploran completamente el espacio de búsqueda, mientras que los algoritmos de búsqueda local sólo exploran ciertas regiones del mismo.

Además, los métodos de optimización también pueden clasificarse con base en las características del problema a resolver en las siguientes categorías:

- Métodos sin restricción, los cuales son utilizados en problemas donde no existen restricciones.
- Métodos con restricciones, los cuales se usan en espacios de búsqueda restringidos.
- Métodos de optimización multi-objetivos, los cuales tienen la habilidad en encontrar más de una solución.
- Métodos dinámicos, los cuales tienen la característica de poder localizar y realizar el seguimiento de trayectoria de un óptimo variable.

1.1.5 Formulación matemática de un problema de optimización.

Un problema de optimización se define matemáticamente de forma general como:

Encontrar $x^* = [x_1^* \quad x_2^* \cdots x_N^*] \in D^N = D_1 \cap D_2 \cap \ldots \cap D_N$

donde

$$f_i^{min}(x^*) \le f_i^{min}(x) \forall x = [x_1 \quad x_2 \cdots x_N] \in D^N, \quad 1 \le i \le N_{f^{min}},$$
$$f_i^{max}(x^*) \ge f_i^{max}(x) \forall x = [x_1 \quad x_2 \cdots x_N] \in D^N, \quad 1 \le i \le N_{f^{max}},$$

$$c_i^{=}(x^*) = 0 \qquad 1 \le i \le N_c^{=}$$
$$c_i^{+}(x^*) > 0 \qquad 1 \le i \le N_c^{+}$$
$$c_i^{-}(x^*) < 0 \qquad 1 \le i \le N_c^{-}$$

donde x^* es la solución óptima en el espacio de búsqueda de N dimensiones D^N. N es el número de parámetros de optimización, o lo que es similar, la dimensión del problema de optimización. D_i es el espacio de búsqueda del i-ésimo parámetro de optimización x_i. x es un vector de N dimensiones que contiene los parámetros a optimizar. $f_i^{min}(x)$ es la i-ésima función objetivo a ser minimizada. $N_{f^{min}}$ es el número de funciones objetivo a minimizar, $f_i^{max}(x)$ es la i-ésima función objetivo a ser maximizada, $N_{f^{max}}$ es el número de funciones objetivo a maximizar. $c_i^{=}(x)$ es la i-ésima función de restricción o condición de igualdad, $N_c^{=}(x)$ es el número de funciones de restricción de igualdad, $c_i^{+}(x)$ es la i-ésima función de restricción positiva, $N_c^{+}(x)$ es el número de funciones de restricción positivas, $c_i^{-}(x)$ es la i-ésima función de restricción negativa, $N_c^{-}(x)$ es el número de funciones de restricción negativas.

En la práctica, las funciones objetivo y restricciones del problema de optimización pueden ser planteadas de diferentes formas. Sin embargo, entre mejor sea la formulación de las funciones objetivo y restricciones, el problema de optimización será descrito de mejor forma y puede ser la diferencia de encontrar su solución óptima. Cualquier conocimiento del problema de optimización debe ser expresado en las funciones objetivo y restricciones del problema de optimización.

1.2 Métodos de optimización determinísticos.

Los algoritmos de optimización determinísticos son aquellos que encuentran la misma solución con el mismo número de evaluaciones de las funciones objetivo sin importar en un mismo tiempo si las condiciones del espacio de búsqueda, punto de inicio y las condiciones de terminación permanecen sin cambio. Si el algoritmo es procesado múltiples veces en la misma computadora, el tiempo de búsqueda en cada procesamiento es exactamente el mismo. Formalmente, un algoritmo determinista resuelve una función matemática la cual tiene un único valor para una determinada entrada en su dominio, siendo el algoritmo el proceso que determina este valor particular de salida.

Los algoritmos de optimización determinísticos se clasifican en función de la dimensión del problema a resolver. De esta forma, se tienen algoritmos de optimización determinísticos uni-dimensionales y algoritmos de optimización determinísticos multi-dimensionales.

Algunos de los algoritmos de optimización determinísticos uni-dimensionales son:

- Algoritmo de búsqueda exhaustiva. Este algoritmo determinista muestrea (divide) el espacio de búsqueda [a,b] en m puntos. Usualmente, los puntos muestreados están equitativamente espaciados en el intervalo [a,b]. De esta forma, el algoritmo de búsqueda exhaustiva determina el punto muestreado en que la función objetivo toma el valor mínimo y al cual considera como la solución óptima al problema. Al algoritmo de búsqueda exhaustiva también se le conoce como algoritmo enumerativo o algoritmo de fuerza bruta.
- Algoritmos dicotómicos. En estos algoritmos se inicia con la selección de dos puntos dentro del intervalo de búsqueda [a,b] y se evalúan mediante la función objetivo. De los dos puntos evaluados, el algoritmo reduce el intervalo de búsqueda, considerando como punto inicial del intervalo aquel en que la función objetivo tuvo el menor valor y como punto final del intervalo al otro valor obtenido en la función objetivo. Este proceso se repite hasta que una condición de terminación es alcanzada.
- Algoritmo de interpolación parabólica. Este algoritmo realiza una aproximación cuadrática local a la función objetivo $f(x)$. Este algoritmo interpola la función objetivo mediante un polinomio cuadrático el cual se ajusta a los valores de la función objetivo evaluada en tres puntos. De esta forma, el punto mínimo de la parábola, el cual es un nuevo estimado del punto mínimo de la función objetivo, reemplaza a uno de los tres puntos previos. Este proceso es repetido por el algoritmo hasta que las condiciones de terminación son cumplidas.

Por otra parte, algunos algoritmos de optimización determinísticos multi-dimensionales son:

- Algoritmo de búsqueda de patrones. Este tipo de algoritmo también es conocido como el algoritmo de Hooke-Jeeves. La idea básica de este algoritmo consiste en moverse de un punto de solución a un próximo. Hay dos tipos de movimiento en el algoritmo de búsqueda de patrones: el movimiento exploratorio y el movimiento de patrones. El movimiento exploratorio consiste de una serie de búsquedas univariadas. Cada búsqueda univariada se mueve un poco en una dirección determinada. Por otra parte, en el movimiento de patrones se tienen movimientos de mayor tamaño.
- Algoritmo Simplex. Este algoritmo también se le conoce como algoritmo colina- abajo (down-hill en inglés). Este algoritmo es un método numérico para minimizar una función objetivo es un espacio multidimensional en problemas dónde no es fácil utilizar las funciones derivadas del espacio de búsqueda. Este método utiliza el concepto de un simplex, el cual es un politipo de N+1 vértices diferentes en N dimensiones. Así por ejemplo, un simplex puede ser un triángulo sobre un plano en un espacio bi-dimensional o un tetraedro en un espacio tridimensional. De esta forma, el algoritmo simplex aproxima a un óptimo local de un problema con N variables cuando la función objetivo varia suavemente y es unimodal. El algoritmo simplex va probando diferentes puntos del simplex en la función objetivo. Si el punto evaluado es mejor que el mejor punto actual lo reemplaza y se mueve suavemente; si por el contario resulta peor el algoritmo se mueve hacia una mejor área del simplex.
- El algoritmo de Newton. La versión multidimensional de este algoritmo es una generalización de su versión uni-dimensional. Este algoritmo requiere del cálculo de la matriz hessiana de las variables de la función objetivo. La matriz hessiana de una función de n variables es la matriz cuadrada de n x n de las segundas derivadas parciales. Si todas las derivadas parciales de la función existen, entonces la matriz hessiana $H_f(x)$ se define como :

$$H_f(x)_{i,j} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

De esta forma, el algoritmo de newton encuentra el mínimo o máximo de una función, encontrando los valores de la matriz Hessiana que la hacen cero.

1.3 Métodos estocásticos

Si bien lo algoritmos de optimización determinístico han sido estudiados por años y han tenido éxito en la solución de problemas; cada vez están enfrentando problemas mucho más complejos en diferentes aplicaciones del mundo real. Cada vez es más frecuente que los algoritmos determinísticos fallen en la búsqueda de una solución, lo que ha expuesto su inherente debilidad. Muchos de los algoritmos determinísticos son formalmente matemáticos; sin embargo, ello también ocasiona que no sean muy amigables con usuarios con profundos conocimientos matemáticos, restringiendo su uso. Los algoritmos determinísticos requieren que el diseñados no tan sólo deduzca la función objetivo y las restricciones del problema de optimización, sino también de sus derivadas. Por lo que el cálculo de estas derivadas produce serias complicaciones computacionales. Además, los algoritmos de optimización determinísticos requieren de uno o más puntos de inicio en la búsqueda de una solución. La buena selección de estos puntos de inicio son críticos para el éxito del algoritmo determinístico.

En las últimas décadas ha crecido el interés en estudiar y desarrollar algoritmos de optimización estocásticos. A diferencia de los algoritmos determinísticos, en los algoritmos estocásticos son impredecibles en lo general por su naturaleza aleatoria. En la práctica, uno no puede encontrar dos soluciones óptimas idénticas, aunque estas soluciones varíen ligeramente en sus valores. Sin embargo, desde el punto de vista práctico dos soluciones diferentes se pueden considerar soluciones óptimas al problema si estas satisfacen los requerimientos impuestos por la aplicación práctica.

Sin embargo, los algoritmos de optimización estocásticos tienen una debilidad al momento de intentar probar su éxito en la búsqueda de la solución óptima, ya sea de forma teórica o numérica. Ningún algoritmo de optimización estocástica garantiza el éxito de encontrar la solución óptima, aunque el porcentaje de falla pueda ser muy pequeño.

A pesar de que los algoritmos estocásticos tienen estas debilidades, en general son matemáticamente mucho más simples que los algoritmos de optimización determinísticos. Para empezar, en los algoritmos estocásticos no se requieren de derivadas o aproximaciones de las mismas. La mayoría de los algoritmos estocásticos parten de una solución inicial aleatoria y de esta forma evitan el problema de requerir un punto de partida inicial como el los algoritmos determinísticos. Todos los algoritmos de optimización estocásticos tienen al menos un parámetro de control intrínseco. El desempeño del algoritmo estocástico depende en cierta medida de este parámetro de control intrínseco. Es bien conocido que la selección adecuada de los valores de los parámetros intrínsecos para un mejor desempeño es una tarea ardua, y en este sentido los algoritmos de optimización estocásticos no son tan simples como uno podría pensar.

1.3.1 Clasificación de los algoritmos de optimización estocástica.

Los algoritmos de optimización estocásticas se pueden dividir en dos grandes categorías de acuerdo a sus orígenes: algoritmos físicos y algoritmos evolutivos. En esta sub-sección se presenta un resumen de los principales algoritmos físicos estocásticos; los algoritmos evolutivos serán abordados en la siguiente sección.

Los algoritmos físicos estocásticos de optimización están inspirados en fenómenos físicos. El algoritmo de Monte Carlo y el algoritmo de recocido simulado son dos de los algoritmos más conocidos de los algoritmos estocásticos.

1.3.1.1 Algoritmo de Monte Carlo.

Este algoritmo recibe su nombre del famoso casino localizado en la ciudad de Mónaco. El algoritmo de Monte Carlo se basa en un repetido muestreo aleatorio para encontrar la solución óptima del problema. El uso de la aleatoriedad y la naturaleza repetitiva del algoritmo son análogos a las actividades que se realizan en un casino. Originalmente, el algoritmo de Monte Carlo fue conocido con otros nombres, como por ejemplo el de muestreo estadístico. Fue en los años 40 en que los científicos que trabajaron en los proyectos de armas nucleares en el laboratorio Nacional de Los Álamos acuñaron el nombre de Algoritmo de Monte Carlo. El principio en que se basa el algoritmo de Monte Carlo consiste en generar aleatoriamente un conjunto de soluciones que son evaluadas en la función objetivo y de las cuales el algoritmo selecciona la solución que mejor satisface los requerimientos del problema.

1.3.1.2 Algoritmo de Recocido Simulado.

El algoritmo de recocido simulado (simulated annealing en inglés) imita el proceso de recocido en la industria metalúrgica. La técnica de recocido implica el calentamiento de un material para posteriormente someterlo a un proceso de enfriamiento controlado con la finalidad de incrementar el tamaño de sus cristales y reducir de esta forma sus defectos. Mediante la analogía con este proceso físico, en cada iteración del algoritmo de recocido simulado reemplaza la solución actual por una nueva solución con una probabilidad que depende de la diferencia de los valores de la función objetivo en dos puntos solución y la temperatura del proceso. La función de distribución de probabilidad más frecuentemente usada es la distribución de probabilidad de Boltzman. Con el uso de esta función de probabilidad es que el algoritmo de recocido simulado previene de ser atrapado en una solución mínima local.

1.4 Algoritmos evolutivos

Los algoritmos evolutivos están inspirados en la teoría de la evolución de Charles Darwin, donde la selección natural es el fundamento de esta teoría de evolución. El estudio de los algoritmos evolutivos inicio en la década de los 60´s. En estos años, un grupo de investigadores concibieron la idea de imitar el mecanismo de la evolución biológica y desarrollaron tres principales áreas de estudio en estos algoritmos. Los algoritmos evolutivos iniciales de estudio fueron los algoritmos genéticos, los algoritmos de programación evolutiva y los algoritmos de estrategias evolutivas. En las últimas dos décadas han surgido una nueva clase de algoritmos evolutivos, denominados algoritmos de enjambre como son el algoritmo de colonia de hormigas, el algoritmo de abejas y el algoritmo de enjambre de partículas. Los algoritmos de enjambre imitan el comportamiento social de animales o de poblaciones humanas.

Las técnicas de computación evolutivas (programación evolutiva, algoritmos genéticos, estrategias evolutivas, y programación genética) están basadas en el concepto de evolución natural. Al inicio del proceso, se crea una población inicial de individuos (padres), entonces estos padres son manejados de acuerdo a variaciones aleatorias para crear nuevas posibles soluciones (hijos). Estas variaciones aleatorias (operadores genéticos) usualmente incluyen mutación (un operador que permite que varios atributos de los individuos sean modificados), y recombinación (operador por medio de cual los atributos de dos o más individuos son combinados para crear un nuevo individuo). Enseguida, los individuos son evaluados a través de una función de aptitud, (cuyo objetivo es proporcionar una medida de qué tan bien adaptados se encuentran los individuos de la población al medio). Finalmente, un operador de selección determina cuál individuo se mantendrá como padre para la siguiente iteración (generación). Este proceso iterativo continúa hasta satisfacer el criterio de convergencia.

A pesar de compartir la misma estructura, la diferencia más notable entre las técnicas está relacionada con el tipo de representación usado por las posibles soluciones (individuos), del tipo de operadores genéticos utilizados, y de la forma en la cual se implementa cada operación.

- Programación evolutiva.

El concepto se propuso en [1], destacando la relación entre la herencia y el comportamiento entre los padres e hijos. En este paradigma, la adaptación es concebida como un tipo de inteligencia.

La programación evolutiva simula la evolución a nivel de las especies. Por lo tanto, no hay operadores de recombinación, (en la naturaleza no es posible el cruce de dos diferentes especies). La selección por torneo se utiliza para evaluar la aptitud de los individuos, y el único operador utilizado es la mutación.

- Estrategias evolutivas

Las estrategias evolutivas fueron desarrolladas en Alemania [2]. Éstas simulan la evolución a nivel de un individuo, entonces hay un operador de recombinación, el cual actúa como un operador secundario, siendo el operador de mutación el operador principal. El operador de mutación varía sobre el tiempo y es auto-adaptivo. La representación en las estrategias evolutivas es a nivel de fenotipo (no se requiere de codificación), y a diferencia de la programación evolutiva, el proceso de selección es determinístico y los peores individuos tienen una probabilidad nula de supervivencia.

- Algoritmos genéticos

Los algoritmos genéticos fueron denominados originalmente "Planes reproductivos" [3]. El método fue desarrollado para resolver un problema de naturaleza mecánica. Los algoritmos genéticos son técnicas de búsqueda basados en el mecanismo de selección natural y genética natural. Trabajan a nivel de genotipo y el operador de recombinación es su principal operador ya que los algoritmos genéticos emulan la evolución a nivel de individuos; la mutación es el operador secundario. La selección es probabilística basada en la aptitud. Generalmente no son auto adaptivos.

- Otras técnicas

Además de las ya descritas, existen otras técnicas de programación evolutivas que no se basan en los paradigmas de selección, recombinación, y mutación, pero que hoy en día son comúnmente usadas en problemas de optimización [4]. Entre estas encontramos la programación genética, evolución diferencial, sistemas de colonia de hormigas, inteligencia artificial, inteligencia de enjambres, en el que se incluye al optimizador por enjambre de partículas. Por sus características se eligió éste último como herramienta de optimización para resolver el problema de optimización planteado en éste trabajo.

CAPÍTULO 2

CAPÍTULO 2

EL OPTIMIZADOR DE ENJAMBRE DE PARTÍCULAS

En este capítulo se presenta el principio de operación del optimizador de enjambre de partículas; así como también se describen los parámetros de control que determinan el desempeño del optimizador.

2.1 Introducción

El optimizador por enjambre de partículas (PSO) es una técnica de optimización estocástica, adaptiva, basada en poblaciones, e introducida por Kennedy y Eberhart en 1995 como una alternativa a los algoritmos genéticos [5-8]. Kennedy y Eberhart 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 línea con trabajos previos en el ámbito de la biología y de la sociología, que concluyen que el comportamiento, inteligencia y movimiento de estas agrupaciones está relacionado directamente con la capacidad de los individuos para compartir información y aprovecharse de la experiencia acumulada por sus congéneres, en [6] se modela dicho comportamiento de forma matemática utilizando expresiones simples que revelan su potencial como método de optimización. La primera versión de PSO se desarrolló para manejar problemas de optimización no lineales con variables continuas. Sin embargo, sus avances han incrementado su capacidad, permitiendo manejar una variedad de problemas de optimización complejos [5].

2.2 Principio de operación del PSO.

El PSO usa el término multitud o enjambre, para referirse a cualquier conjunto de agentes o individuos que interactúan entre sí y con el medio que los rodea. El ejemplo clásico de enjambre lo representan las abejas en las inmediaciones del panal, aunque esta analogía se hace extensible a cualquier otro sistema con una arquitectura y comportamiento social como grupo similar. Para entender el funcionamiento del PSO como algoritmo de optimización supongamos, a modo de ejemplo, el comportamiento que exhibe un enjambre de abejas en su movimiento sobre un campo cubierto con diferentes concentraciones de flores. Sin ningún conocimiento a priori del espacio de búsqueda, las abejas inician su movimiento desde posiciones aleatorias y con velocidades aleatorias. En su desplazamiento, el objetivo del enjambre se centra en encontrar el emplazamiento con la mayor densidad de flores. Cada abeja tiene memoria y puede recordar la posición visitada con mayor densidad de flores y también conoce, por mecanismos de comunicación con sus congéneres, la localización donde otras abejas encontraron una densidad de flores significativa. Esta dupla de información es utilizada por la abeja para modificar continuamente su trayectoria, acelerando en ambas direcciones y volando hacia un punto espacial intermedio que dependerá de su posición actual y de cómo influyan sobre su decisión las así denominadas nostalgia o memoria y cooperación o conocimiento social. De esta forma, las abejas se encuentran permanentemente sobrevolando el campo en busca de posiciones con mayor densidad de flores, redirigiendo en parte la trayectoria del enjambre cada vez que se encuentran configuraciones de mayor calidad.

Con el transcurso del tiempo, una vez que ha sido explorado el espacio de soluciones en su totalidad, el conjunto del enjambre se encontrará volando alrededor de la zona con la mayor concentración de flores de todo el campo. En esta situación, las abejas, incapaces de encontrar posiciones alternativas mejores, son permanentemente atraídas hacia dicha posición. Este comportamiento social se ilustra en la Figura 2.1.

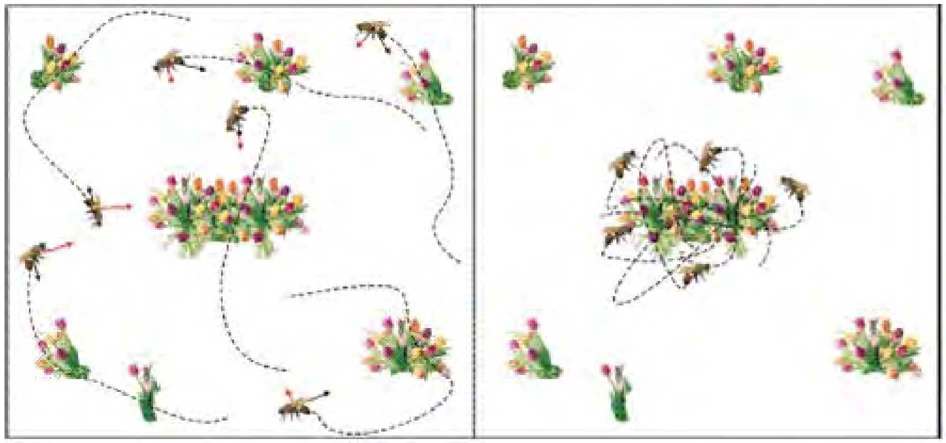


Figura 2.1. Analogía del comportamiento del PSO con el movimiento de un enjambre de abejas sobre un campo con flores.

El comportamiento social que exhiben éste y otros organismos se puede entender como un método de optimización en el que el espacio de búsqueda se puede extender a las N dimensiones del problema a optimizar, y donde cada partícula, abeja en este caso, se identifique como una potencial solución al problema. Cada partícula es caracterizada por un vector velocidad y un vector posición, ambos en N dimensiones. De esta forma, el problema se reduce a establecer la ecuación que dicte cómo debe moverse cada partícula de la población en el espacio N-dimensional para mimetizar la inteligencia de estas comunidades y evitar a su vez caer en soluciones locales.

La idea básica del PSO es el modelado matemático y la simulación del proceso de búsqueda de comida de un enjambre (población). En un espacio multidimensional donde se busca la solución óptima, cada individuo (partícula) en el enjambre se mueve hacia la solución óptima, adicionándole un término de ajuste a la posición actual (velocidad de ajuste). En el PSO clásico, la velocidad se compone por tres términos denominados: inercial, social, y cognitivo.

El componente inercial simula el comportamiento inercial de las abejas de volar en direcciones previas. El término cognitivo modela la memoria de las abejas para recordar la mejor posición previa encontrada, y el término social modela la memoria de las abejas para grabar la mejor posición entre las partículas (interacción entre las partículas). Las partículas se mueven alrededor del espacio de búsqueda multidimensional hasta que encuentran la comida (solución óptima).

El PSO primero produce una población inicial, cuyos miembros representan una posible solución al problema de optimización; esta tiene su propia posición y velocidad inicial. Para un problema N-dimensional, la posición y velocidad pueden ser especificados por matrices de tamaño M x N como se describe en (1) y (2).

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ & & \ddots & \\ x_{M1} & x_{M2} & \cdots & x_{MN} \end{bmatrix} \quad (1)$$

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1N} \\ v_{21} & v_{22} & \cdots & v_{2N} \\ & & \cdots & \\ v_{M1} & v_{M2} & \cdots & v_{MN} \end{bmatrix} \quad (2)$$

donde X y V son la matriz de posiciones y velocidades, respectivamente. M, representa el número de partículas que constituyen una población. Cada fila en la matriz de posiciones representa la posición de cada partícula en el espacio de búsqueda, a través de la cual se calcula el valor de aptitud de la partícula.

En cada iteración, cada partícula memoriza y sigue el rastro de su mejor población (Pbest), y el vector con la mejor posición global (Gbest) para actualizar la velocidad de la matriz. La mejor población (Pbest) es la posición con el mejor valor de aptitud que se ha encontrado, y se puede definir por (3):

$$Pbest = \begin{bmatrix} pbest_{11} & pbest_{12} & \cdots & pbest_{1N} \\ pbest_{21} & pbest_{22} & \cdots & pbest_{2N} \\ & & \cdots & \\ pbest_{M1} & pbest_{M2} & \cdots & pbest_{MN} \end{bmatrix} \quad (3)$$

La mejor posición global (Gbest), es la mejor posición de entre todas las partículas de la mejor población (Pbest) y es definida como:

$$\mathrm{Gbest} = \left[\mathrm{gbest}_1 \quad \mathrm{gbest}_2 \quad \cdots \quad \mathrm{gbest}_N \right] \tag{4}$$

Conociendo estas dos mejores posiciones, las partículas pueden modificar velocidades y posiciones usando las siguientes expresiones:

$$v_{ij}^{iter+1} = w.v_{ij}^{iter} + C_1 * rand() * (pbest_{ij} - x_{ij}^{iter}) + C_2 * Rand() * (gbest_i - x_{ij}^{iter}) \tag{5}$$

$$x_{ij}^{iter+1} = x_{ij}^{iter} + v_{ij}^{iter+1} \qquad i = 1,2,.....,M \qquad j = 1,2,.....,N \tag{6}$$

donde, *iter*, es la iteración (generación) actual; C_1, C_2, son coeficientes positivos de aceleración, que ajustan el paso máximo de aproximación de las partículas hacia Pbest y Gbest; rand(), Rand(), son variables generadas aleatoriamente, uniformemente distribuidas en el rango [0, 1], cuyo objetivo es emular el comportamiento estocástico y un tanto impredecible que exhibe la población del enjambre; *w*, es conocido como el peso inercial, y cumple la función de balancear la búsqueda local y la búsqueda global, [6-8]. Con pequeñas modificaciones, pueden manejarse variables discretas en (5) y (6).

Es importante mencionar que la ecuación 5 que describe el cálculo de la velocidad de cada partícula consiste de tres términos:

- La velocidad previa, $v_{i,j}^{iter}$, le sirve al PSO como una memoria de la dirección del vuelo previo, es decir, el movimiento en el pasado inmediato. Este termino de memoria se puede considerar como un momento, el cual previene que la partícula cambie drásticamente de dirección y afecte la dirección actual. A este parámetro se le conoce también como la componente inercial del PSO.

- La componente cognitiva, $C_1 * rand() * (p_{best} - x_{i,j})$, la cual cuantifica el desempeño de la partícula *i,j* respecto a desempeños anteriores. En cierta medida, la componente cognitiva se asemeja a la memoria individual de la mejor posición que ha tenido la partícula. El efecto de este parámetro es que las partículas son dirigidas hacia sus mejores posiciones, lo cual se parece a la tendencia de los individuos de retornar a las situaciones o lugares que los han dejado muy satisfechos anteriormente. Kennedy y Eberhart se refieren a esta componente cognitiva como la nostalgia de la partícula.
- La componente social, $C_2 * Rand() * (g_{best} - x_{i,j})$, la cual cuantifica el desempeño de la partícula *i,j* respecto a las partículas vecinas. De manera conceptual, esta componente social es parecida a las normas o estándares que los individuos deben respetar en una sociedad. El efecto de la componente social es que cada partícula es dirigida hacia le mejor posición encontrada por las partículas en una región de búsqueda.

La Figura 2.2 esquematiza el concepto de modificación del punto de búsqueda.

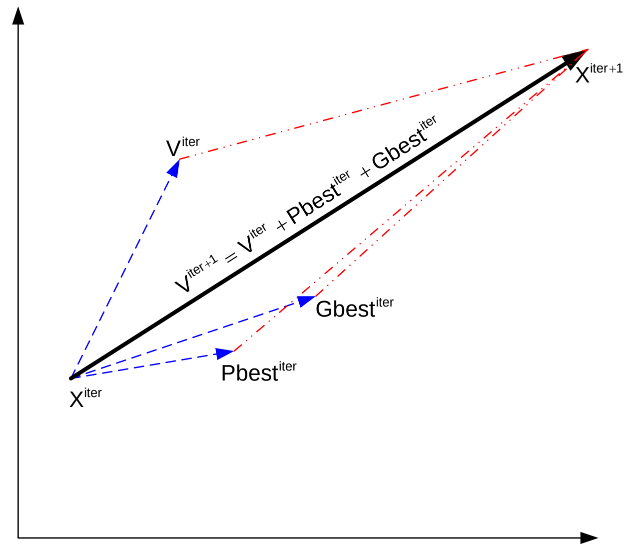


Figura 2.2 Movimiento de una partícula del PSO.

La Figura 2.3 muestra el detalle del movimiento de cada partícula del PSO, influenciada por su componente cognitiva y social.

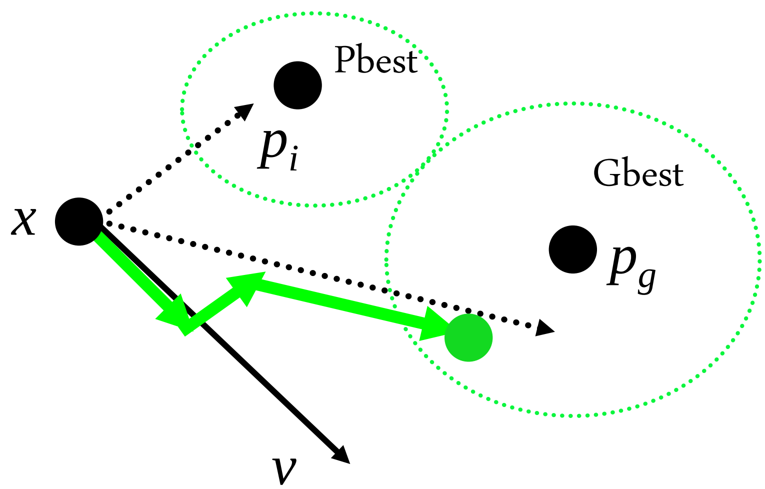


Figura 2.3 Contribución de la componente congnitiva y social en la partícula x del PSO.

A partir de (5) y (6), se observa que cuando una partícula coincide con la mejor posición global (Gbest), la partícula saldrá de este punto si el peso inercial w y su velocidad actual son diferentes de cero. Si las velocidades actuales de las partículas son cercanas a cero, entonces estas partículas no se moverán y quedarán atrapadas, lo que significa que todas las partículas convergen a la mejor posición (Gbest) encontrada por la partícula. Para este caso, se presenta un problema de convergencia prematura si la aptitud de la mejor posición encontrada no corresponde a un óptimo global. Para resolver este problema la literatura proporciona alternativas basadas en adicionar variantes de operadores de mutación al PSO.

El uso de un operador de mutación en PSO no es nuevo. Higashi e Iba [4], Secrest y Lamond [5], proponen incorporar una mutación con distribución de probabilidad Gaussiana al PSO convencional o a una modificación de éste.

2.3 Parámetros del PSO.

Los parámetros de control del algoritmo de optimización PSO son la dimensión del problema, el tamaño del enjambre o población, los coeficientes de aceleración, el peso inercial, el tamaño del vecindario, el número de iteraciones y los valores aleatorios que escalan la contribución de las componentes cognitiva y social. Además, si la velocidad de las partículas se limita o si se utilizan restricciones, los coeficientes de velocidad máxima y de restricción también influyen en el desempeño del PSO.

2.3.1 Tamaño del enjambre o población.

Este parámetro se refiere al número de partículas que componen el enjambre. A mayor número de partículas en el enjambre, mayor es la diversidad inicial del enjambre con lo que se obtiene un buen esquema uniforme de inicialización para las partículas. Con un gran número de partículas en el enjambre, se puede explorar mayores regiones en el espacio de búsqueda a ser cubierto en cada iteración del PSO, así como también puede permitir que el PSO alcance la solución óptima en menos iteraciones. Sin embargo, con un mayor número de partículas también aumenta la complejidad computacional y la búsqueda aleatoria se degrada a una búsqueda aleatoria paralela. No existe un fundamento matemático formal que indique cuál debe ser el número de partículas a emplear en el enjambre del PSO. Existen estudios empíricos que demuestran que el PSO tiene un mejor desempeño en encontrar la solución óptima con enjambres pequeños de 10 a 30 partículas [8], aunque también se han reportado aplicaciones en que se han utilizado enjambres menores a 10 partículas. Sin embargo, el tamaño óptimo del enjambre de partículas del PSO es dependiente del problema a resolver. Un espacio de búsqueda suave necesitará menos partículas que una superficie agreste para localizar soluciones óptimas. De esta forma, aunque se puede seleccionar el tamaño del enjambre de partículas basado en el conocimiento heurístico reportado en la literatura abierta, es mejor determinar el número óptimo de partículas para cada problema a partir de métodos de validación cruzada.

2.3.2 Tamaño del vecindario.

El tamaño del vecindario define el grado de interacción social dentro del enjambre. Si se usan vecindarios pequeños, la interacción entre partículas será menor. Mientras que vecindarios pequeños son más lentos en converger, ellos son más confiables en converger en las soluciones óptimas. Los vecindarios pequeños son menos susceptibles a ser atrapados en mínimos locales. Por esta razón, se recomienda iniciar con pequeños vecindarios de búsqueda e ir incrementando el tamaño del vecindario conforme se aumente el número de iteraciones del PSO. Esta estrategia asegura una alta diversidad inicial con una convergencia más rápida conforme las partículas se muevan hacia una región de búsqueda prometedora. La Figura 2.4 muestra el vecindario geográfico y social de las partículas del PSO.

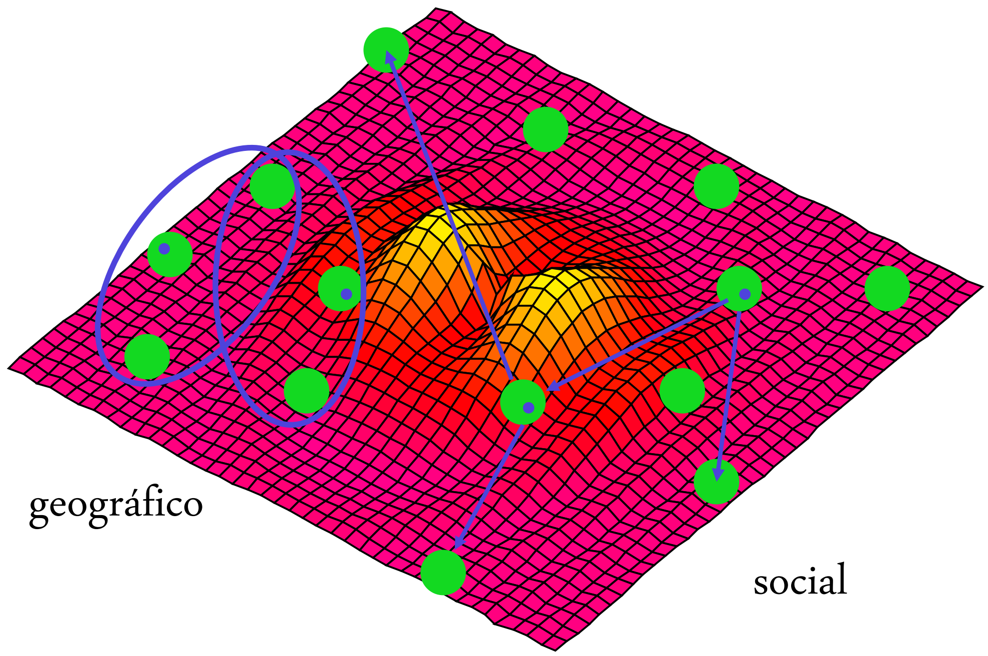


Figura 2.4 Vecindarios de búsqueda de las partículas del PSO.

2.3.2 Número de iteraciones.

Este parámetro de control del PSO también es dependiente de la complejidad del problema. Si se utiliza un pequeño número de iteraciones, el PSO terminará prematuramente. Por el contrario, si se utiliza un número grande iteraciones provocará que la complejidad computacional del PSO aumente.

2.3.3 Coeficientes de aceleración.

Los coeficientes de aceleración, c_1 y c_2 así como de términos aleatorios rand(), controlan la influencia estocástica de las componentes sociales y cognitivas en la velocidad de cada partícula del PSO. Las constantes de aceleración c_1 y c_2 también son conocidas como parámetros de confiabilidad de las partículas, ya que la constante c_1 expresa que tanto confía una partícula en si misma, mientras que

c_2 expresa que tanto la partícula confía en los resultados obtenidos por sus vecinos. Si c_1=c_2=0, entonces las partículas se mantendrían volando en su velocidad actual hasta llegar a la frontera del espacio de búsqueda. Si c_1 > 0 y c_2 =0, todas las partículas se moverían independientemente, ignorando la información de las partículas vecinas. Cada partícula encontraría la mejor posición en su vecindario de búsqueda reemplazando la mejor posición actual por la nueva posición si es que esta es mejor; por lo que las partículas estarían realizando una búsqueda local exclusivamente. Por otra parte, si c_2>0 y c_1=0, entonces el enjambre en su total sería atraído a un punto en singular y el PSO se convertiría en un algoritmo de optimización estocástico. Con base en el comportamiento de las partículas bajo estas condiciones en particular, se fortalecería la característica de búsqueda de la partícula si la naturaleza cooperativa de las partículas se prioriza haciendo que la nostalgia y envidia de cada partícula coexistan balanceadamente y $c_1 \approx c_2$. Aunque la relación de los valores de c_1 y c_2 también son dependientes de la complejidad del problema, en la mayoría de las aplicaciones de PSO se emplea que c_1= c_2. Si c_1>> c_2, cada partícula es mayormente atraída hacia su mejor posición personal, lo que resulta en un excesivo movimiento errante. Por otras parte si c_2>> c_1, las partículas son atraídas con mayor fuerza hacia la mejor posición global, lo que provoca que las partículas se precipiten prematuramente hacia un óptimo. En problemas uni-modales con espacios de búsqueda suave, es más conveniente resaltar la componente social del enjambre, mientras que en problemas multi-modales con superficies altamente agrestes la componente individual de cada partícula produce mejores resultados.

Por otra parte, valores pequeños para c_1 y c_2 provocan que las trayectorias de las partículas sean suaves y que vaguen lejos de buenas regiones del espacio de búsqueda para explorar. En caso contrario, valores altos de c_1 y c_2 ocasiona que las partículas se aceleren en mayor cantidad y tengan movimientos más abruptos. De esta forma, una mala selección de los valores de c_1 y c_2 pueden originar que el PSO tenga un comportamiento divergente y nunca encuentre la solución óptima al problema. En las aplicaciones de PSO se ha encontrado de manera experimental que $c_1 + c_2 = 4$.

2.3.4 Peso inercial.

El peso inercial, w, fue introducido en el algoritmo del PSO como un mecanismo para controlar las habilidades de exploración y explotación del enjambre de partículas, así como también como un mecanismo para eliminar la necesidad para limitar las velocidades de las partículas. Aunque el parámetro del peso inercial ha sido exitoso en su primer objetivo, no eliminó la necesidad de limitar la velocidad de las partículas.

El peso inercial controla el momento de la partícula ponderando la contribución de la velocidad de la partícula en la iteración inmediata anterior, por lo que básicamente controla la influencia de recordar la dirección previa de vuelo en la nueva velocidad de la partícula.

El valor del peso inercial es muy importante para asegurar la convergencia del PSO. Para valores de $w \geq 1$, las velocidades de las partículas aumentan con el tiempo, por lo que las partículas se aceleran hacia su máxima velocidad (asumiendo que la velocidad de las partículas se ha restringido) por lo que el enjambre diverge. En esta condición, las partículas se ven impedidas en cambiar de dirección y moverse hacia áreas más prometedoras. Para valores de $w<1$, las partículas se desaceleran hasta que sus velocidades alcanzan el cero. Al igual que los otros parámetros de control del PSO, el peso inercial también es dependiente de la complejidad del problema.

El valor del peso inercial deber ser seleccionado en conjunto con los valores de los coeficientes de aceleración, c_1 y c_2. En las primeras aplicaciones del PSO el valor del peso inercial era un valor estático; sin embargo en las últimas investigaciones se ha encontrado que el PSO presenta un mejor desempeño si w cambia dinámicamente con respecto al número de iteraciones, como es descrito por:

$$w(t) = \left(w(0) - w(n_t)\right)\frac{n_t - t}{n_t} + w(n_t) \quad (7)$$

Donde n_t es el número total de iteraciones del PSO, $w(0)$ es el valor inicial del peso inercial, $w(n_t)$ es el valor final para el peso inercial, mientras que $w(t)$ es el valor del peso inercial en la iteración *t*. La ecuación (7) modifica el valor del peso inercial durante el tiempo de simulación del PSO, decrementándolo linealmente. Existen variantes de PSO en que el peso inercial es dinámicamente modificado en forma no lineal, aleatoria o difusa; por lo que también la selección de la modificación de este parámetro es dependiente de la complejidad del problema.

2.4 Algoritmo del PSO

El algoritmo del PSO básicamente consiste en ir calculando iterativamente la velocidad de cada partícula a partir de las mejores posiciones personales y globales de las partículas. De esta forma, el algoritmo del PSO está descrito por el diagrama de flujo de la Figura 2.5.

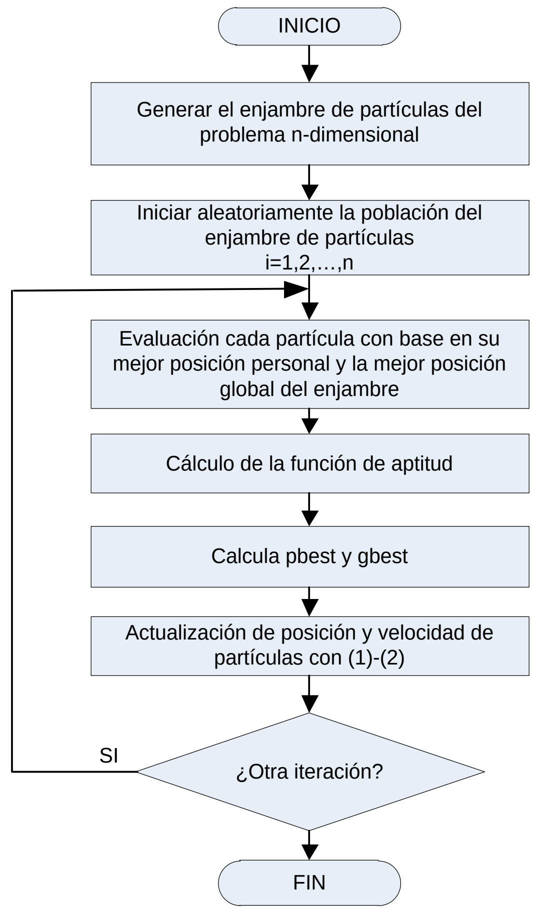


Figura 2.5 Diagrama de flujo del optimizador de enjambre de partículas (PSO).

El proceso de optimización del PSO es iterativo. El algoritmo es repetido iterativamente hasta que la condición de terminación es satisfecha. Cada iteración consiste en determinar la mejor posición personal y global y de esta forma determinar la nueva velocidad de la partícula. De igual forma, en cada iteración un determinado número de funciones de evaluación se realiza. La función de evaluación se refiere al cálculo de una función aptitud que caracteriza al problema de optimización. Para el algoritmo básico del PSO se analizan n_s funciones de evaluación, dónde n_s es el número total de partículas del PSO.

El primer paso del algoritmo del PSO consiste en inicializar las partículas del enjambre y los parámetros de control del algoritmo. Por lo que se requiere que se especifiquen los valores de las constantes de aceleración, las velocidades iniciales, así como las posiciones iniciales y las mejores posiciones personales.

Usualmente, las posiciones de las partículas se inicializan de tal forma que cubran uniformemente el espacio de búsqueda, como se muestra en la Figura 2.6.

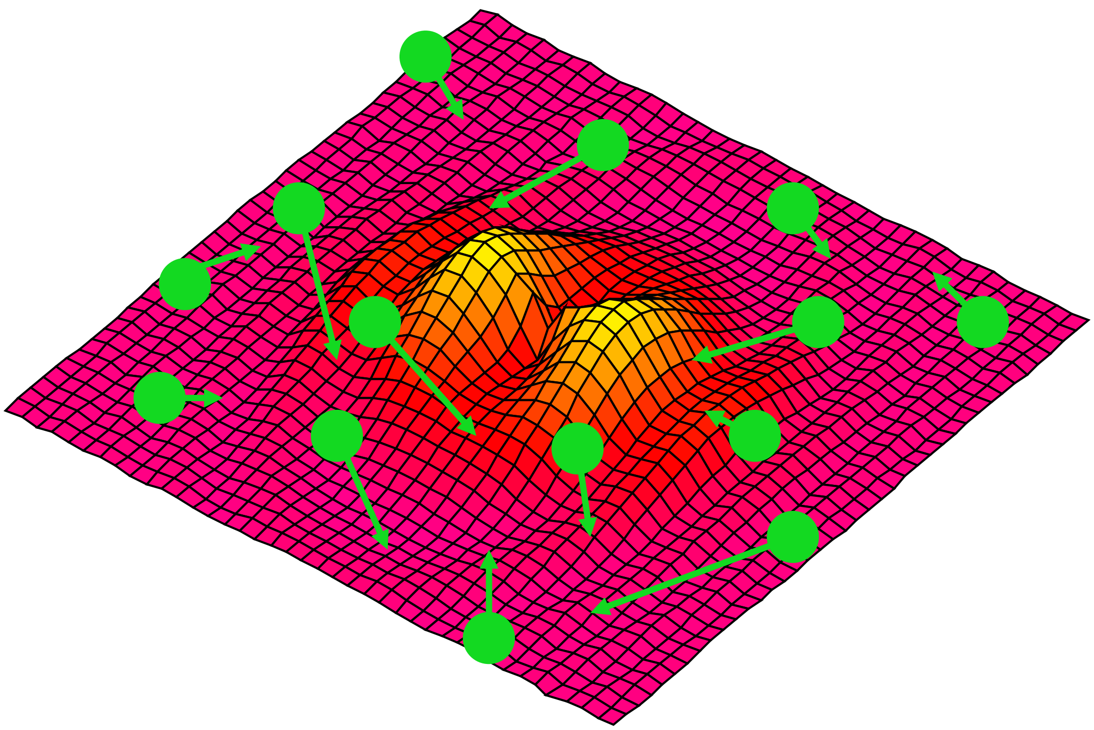


Figura 2.6 Distribución de partículas del PSO en una superficie de búsqueda.

Es importante resaltar que la eficacia del PSO depende en gran medida del área cubierta inicial del espacio de búsqueda y la forma en que las partículas cubrieron esa área inicial. Si quedan regiones del espacio de búsqueda que no son cubiertas por la población inicial del enjambre de partículas, el PSO difícilmente encontrará la solución óptima si esta se encuentra en esta área no cubierta. El PSO sólo encontrará la solución óptima si el momento de alguna o algunas de las partículas lo lleva a esa zona no cubierta del espacio de búsqueda. La Figura 2.7 muestra la interacción y movimiento de dos partículas del PSO.

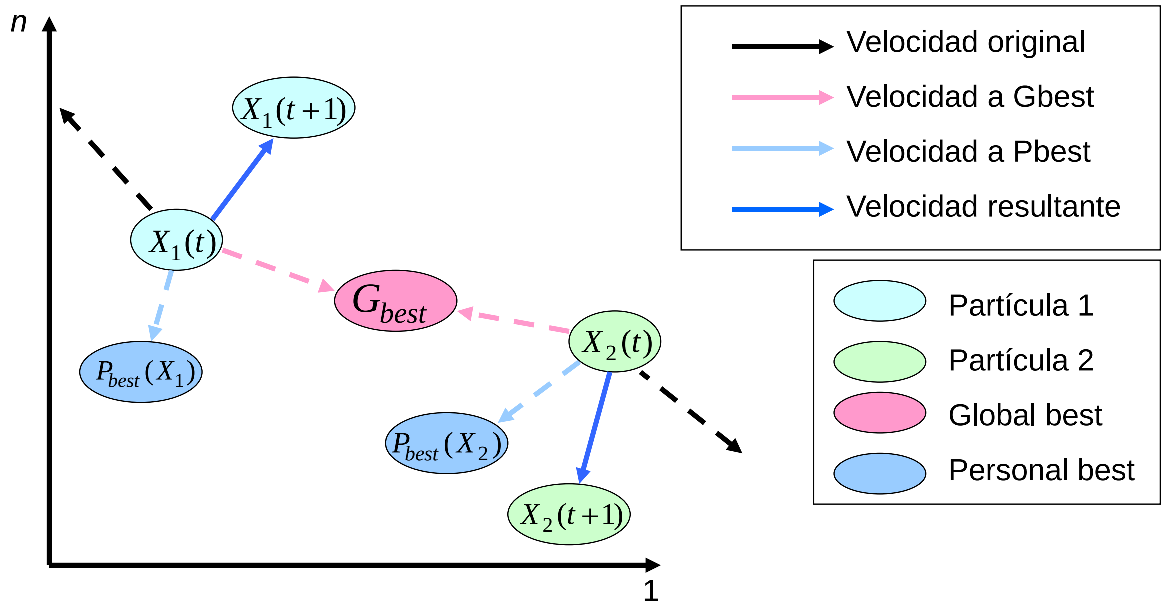


Figura 2.7 Movimiento de dos partículas del PSO a partir de su mejor personal y su mejor global.

Por el contrario, la terminación del algoritmo del PSO está relacionada con una condición de alto al proceso iterativo del algoritmo. En la literatura abierta se han utilizado diferentes criterios de terminación del algoritmo del PSO; sin embargo se deben tener en cuenta los siguientes aspectos:

1. La condición de terminación no debe provocar que el PSO converja prematuramente, ya que la solución encontrada por el PSO será muy probablemente una solución sub-óptima.
2. La condición de terminación debe evitar una sobre-evaluación de la función de aptitud del PSO, ya que esto incrementará en gran medida la complejidad computacional del algoritmo.

Las condiciones de terminación para el procesamiento del algoritmo del PSO principalmente utilizadas son:

- Terminación por un número máximo de iteraciones.
- Terminación cuando el PSO ha encontrado una solución aceptable a cierto criterio.
- Terminación cuando no se observa una mejora en la solución durante un determinado número de iteraciones.

CAPÍTULO 3

CAPÍTULO 3

APLICACIONES DEL PSO

En este capítulo se presenta un estudio de las diferentes aplicaciones de PSO en la solución de problemas de optimización en el área de los sistemas eléctricos de potencia. En el capítulo se destacan las características claves del PSO y las ventajas sobre otros algoritmos de optimización. Además, se exploran las recientes en materia de desarrollo del PSO en esta área. Este capítulo también discute las posibles aplicaciones futuras del PSO en el área de los sistemas eléctricos de potencia y sus posibles estudios teóricos.

3.1 Introducción

Los problemas de optimización son ampliamente encontrados en varios campos de la ciencia y la tecnología. A veces este tipo de problemas pueden ser muy complejo debido a la naturaleza real y práctica de la función objetivo o debido también a las limitaciones de los modelos matemáticos utilizados. Tradicionalmente, los métodos de optimización utilizados en este tipo de problema requieren del cálculo de derivadas de funciones. Estas técnicas son robustas y han demostrado su eficacia en el manejo de muchas clases de problemas de optimización. Sin embargo, estas técnicas pueden tener dificultades tales como quedar atrapado en mínimos locales, producir el aumento de la complejidad computacional, o en ocasiones no son aplicables a ciertas clases de funciones objetivo. Esto llevó a la necesidad de desarrollar una nueva clase de métodos de optimización que pueden superar estas deficiencias. Las técnicas de optimizaciones heurísticas son herramientas que pueden superar la mayor parte de las limitaciones encontradas en las técnicas basadas en derivadas y que en las últimas décadas se ha incrementado su aplicación en la solución a problemas de optimización complejos.

Por otra parte, en los sistemas de eléctricos de potencia, hay muchos problemas de optimización tales como el flujo óptimo de potencia, despacho económico, control de potencia reactiva y reducción de pérdidas de potencia, la generación y planificación de la transmisión de energía, la programación de mantenimiento de unidades, estimación del estado de la red eléctrica, la identificación del modelo, la previsión de carga y de control. Diversas técnicas de optimización se han aplicado a resolver estos problemas de optimización, tales como programación no lineal, programación cuadrática, programación lineal, el método de punto interior, algoritmo genético, y programación evolutiva. Sin embargo, en el uso práctico, todas estas técnicas tienen desventajas en cierta medida, debido a la complejidad de los problemas de optimización

Recientemente, el PSO ha recibido mucha atención respecto a su potencial como técnica de optimización global. En PSO, a cada solución potencial se le asigna una velocidad aleatorizada, y estas posibles soluciones potenciales, llamada partículas, vuelan a través del espacio de búsqueda del problema siguiendo a las mejores partículas actuales. A diferencia de los métodos matemáticos de programación, el PSO no es sensible a los puntos de inicio o a la forma de la función objetivo. Con respecto a otros algoritmos evolutivos, el PSO es capaz de evolucionar hacia el óptimo global con una velocidad aleatoria y tiene un mejor rendimiento de la búsqueda global con una convergencia más rápida. Debido a estas ventajas del PSO, los investigadores lo han aplicado para resolver problemas de optimización en sistemas eléctricos de potencia y en dónde se han obtenido resultados exitosos en diversas aplicaciones en esta área de la ingeniería.

El PSO se ha aplicado con éxito en problemas de optimización multi-objetivo, optimización en problemas con restricciones, en el entrenamiento de redes neuronales artificiales, o en la optimización de parámetros de modelos, por mencionar algunos ejemplos de aplicaciones. La Figura 3.1 muestra el empleo del PSO en trabajos de investigación reportados en la base de datos de la Asociación de Ingenieros Eléctricos y Electrónicos (IEEE por sus siglas en inglés), durante los últimos 15 años, en las diferentes áreas de ingeniería de la IEEE.

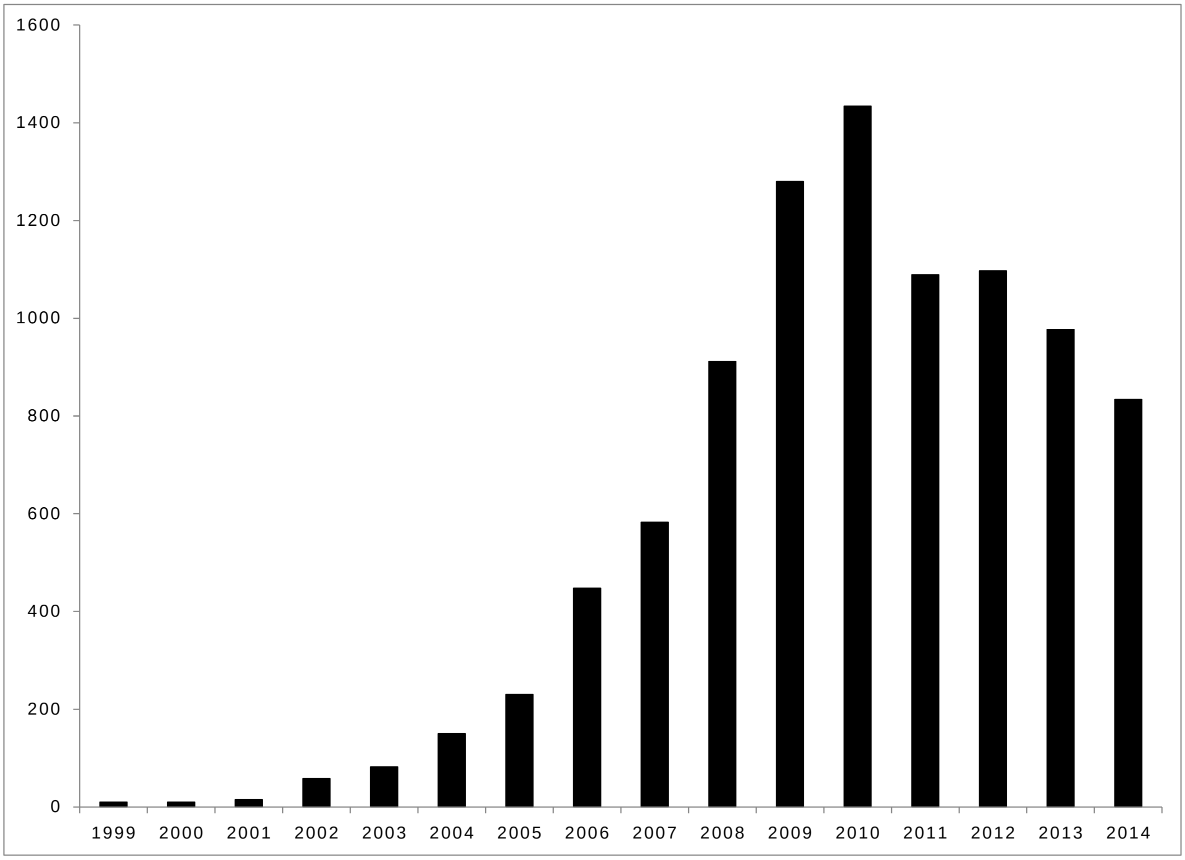


Figura 3.1 Número de artículos publicados sobre el PSO en la base de datos de la IEEE durante los últimos quince años.

Como se observa en la Figura 3.1, en el 2010 se tuvo el mayor número de publicaciones sobre el PSO y a partir de este año se observa una disminución. Esta reducción de trabajos del PSO en las diferentes áreas de la IEEE se puede explicar en el sentido de las áreas de computación y computación evolutiva han llegado a la madurez de estudio del algoritmo del PSO, cubriendo diversos aspectos de análisis y estudio, por lo que en estas áreas principalmente han disminuido el número de artículos al respecto del PSO. Sin embargo, esto no significa que el PSO esté obsoleto, sino que el área de estudio que está creciendo ahora en el número de trabajo corresponde a las aplicaciones del PSO en la solución de problemas de ingeniería, como se observa en la Figura 3.2.

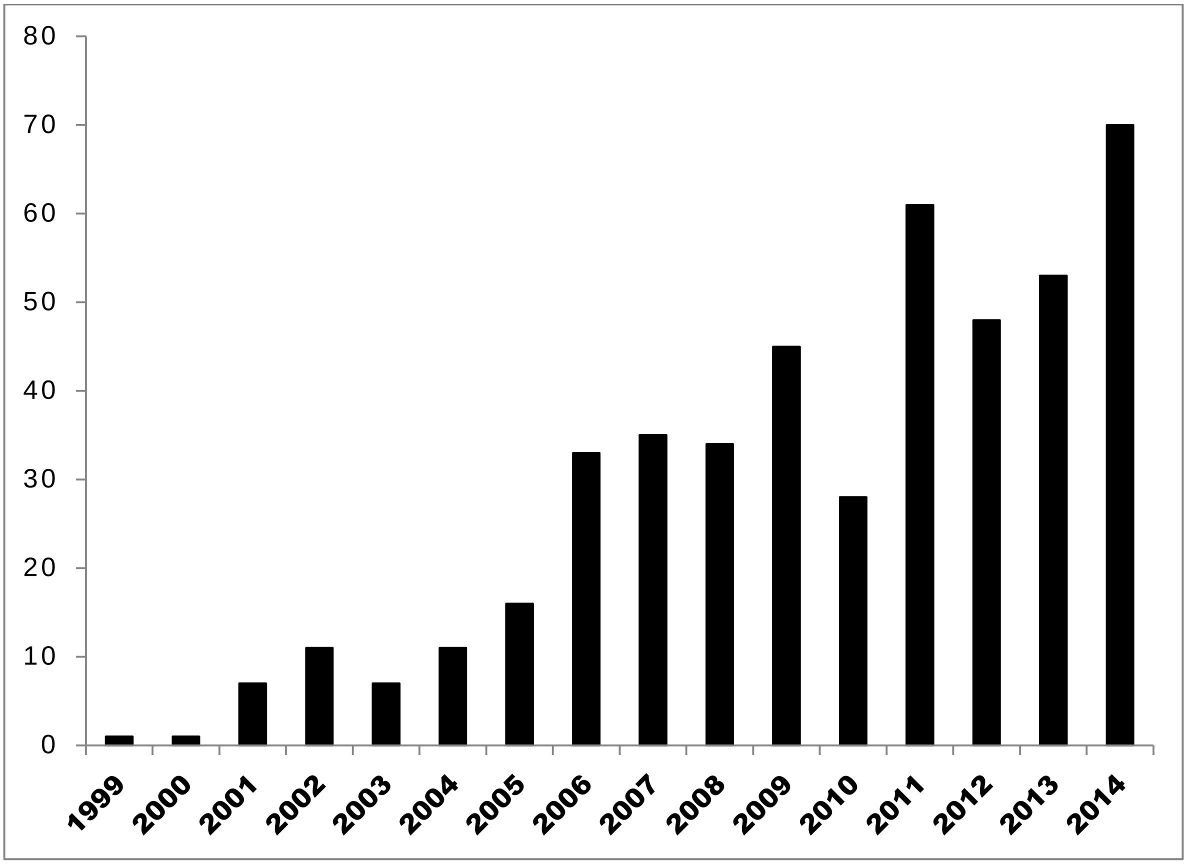


Figura 3.2 Número de artículos publicados sobre el PSO en el área de sistemas eléctricos de potencia.

La Figura 3.2 muestra la tendencia del número de publicaciones sobre la aplicación del PSO en el área de sistemas eléctricos de potencia. Como se observa, la tendencia de los últimos años es un incremento en el número de artículos publicados, lo cual es un indicativo que se está incrementando el número de aplicaciones del PSO en la solución de problemas de optimización del área de sistemas eléctricos de potencia. Cabe mencionar que en la Figura 3.2, el número de artículos sobre el PSO disminuye en ciertos años (2010, 2012 por ejemplo), pero esto se explica porque algunos congresos se realizan cada dos años, lo cual impacta en el número de publicaciones del año en que no se realizan estos eventos.

En la siguiente sección se presenta un estudio sobre las aplicaciones de PSO en sistemas eléctricos de potencia.

3.2 Aplicaciones del PSO en sistemas eléctricos de potencia

En esta sección se presenta un resumen del uso del algoritmo del PSO para la solución de problemas de optimización en el área de sistemas eléctricos de potencia.

3.2.1 Flujo de potencia óptimo.

El flujo de potencia óptimo (OPF por sus siglas en inglés) es una problema de optimización no-lineal, de gran escala, que calcula los valores óptimos de las variables eléctricas bajo la condición de las limitaciones impuestas por datos operacionales y las limitaciones físicas de la red eléctrica. Para resolver los problemas de OPF, se deben hacer algunas modificaciones al algoritmo PSO básico [9-13].

De esta forma en [9] se presenta la solución de flujo de potencia óptimo mediante la optimización con enjambre de partículas con penalización en la función objetivo. En este trabajo se utiliza una función de penalización no estacionaria de múltiples etapas para convertir el problema de optimización con restricciones en un problema de optimización sin restricción. De esta forma, el desempeño del PSO es probado en un sistema eléctrico estándar IEEE de 30-buses y los resultados obtenidos demuestran que el PSO obtuvo soluciones de mayor calidad y de manera más eficiente en el problema de OPF en comparación con los algoritmos de programación lineal y algoritmo genético. Sin embargo, este estudio también demostró que si los coeficientes de penalización son inadecuados pueden hacer que el algoritmo del PSO converja lentamente o de forma prematura. Debido a esto, una de las conclusiones a la que llegaron los autores es que la selección de un valor óptimo de coeficientes de penalización por sí mismo un problema de optimización difícil.

Por otra parte, los autores del trabajo [10] mejoran el algoritmo del PSO mediante la incorporación de un concepto de la biología denominado "congregación pasiva" para resolver el problema de flujo de potencia óptimo. En la referencia [11] se presenta un PSO que utiliza un esquema de mutación en sus partículas para resolver el flujo de potencia óptimo mediante una mezcla de variables de control continuas y discretas y funciones de costo de combustible discontinuas. Por su parte, los autores de [12] presentan una mejora al PSO al hacer que las partículas no tan sólo aprendan de su mejor desempeño, o de la partícula de mejor desempeño, sino que también aprendan de otros individuos del enjambre. Estas modificaciones al algoritmo básico del PSO evitar de manera efectiva la convergencia prematura del PSO, permitiéndole obtener resultados satisfactorios en la solución de problemas de OPF.

En la referencia [13] se presenta un estudio de la aplicación del PSO como una herramienta para el análisis de reducción de pérdidas. El análisis se lleva a cabo en dos pasos. En primer lugar, mediante el uso de la técnica de vector tangente, el área crítica del sistema de potencia se identifica bajo el punto de vista de la inestabilidad de voltaje. En segundo lugar, una vez que se identifica esta área, el algoritmo del PSO calcula la cantidad de compensación de potencia reactiva de derivación que debe haber para cada bus. El enfoque propuesto ha sido examinado y probado con resultados numéricos prometedores utilizando el sistema 118-bus de la IEEE.

3.2.2 Despacho económico.

El Despacho Económico (ED por sus siglas en inglés) se define como el proceso de asignación de niveles de producción de energía de las unidades de generación disponibles, de manera que la demanda de energía de la carga eléctrica conectada a la red eléctrica sea cubierta al menor costo [14]. Matemáticamente, el problema del despacho económico es encontrar la combinación óptima de los generadores de energía que reduzcan al mínimo el costo total de generación al tiempo que satisfacen las restricciones del problema. El Despacho económico se puede clasificar como un problema de optimización estática, en la no se consideran los costos asociados con el hecho de cambiar la energía producida por los generadores. Por otra parte, también se puede clasificar como un problema de optimización dinámica en la que se consideran los cambios en los costos de producción.

Las referencias [15, 16] presentan una mejora al algoritmo del PSO para resolver problemas de despacho económico con funciones de costos complejas. Para ello, los autores propusieron el diseñó de un mecanismo de tratamiento de restricciones de tal manera que se conserva el proceso dinámico inherente al PSO convencional. Además, se concibió una estrategia dinámica de reducción del espacio de búsqueda para acelerar el proceso de optimización. Los resultados obtenidos por los autores demostraron que el PSO modificado demostró un desempeño mejor comparado con métodos de optimización convencionales.

Por otra parte, en las referencias [17, 18] se presenta la aplicación del PSO para resolver el despacho económico de carga con flujos de línea y limitaciones de tensión. En este trabajo los autores concluyeron que el PSO es computacionalmente más rápido que la programación lineal, programación cuadrática y el algoritmo genético en la solución de este tipo de problemas de optimización. En la referencia [19] se propone la incorporación del algoritmo-EP con mutación de Gauss en el PSO, para resolver los problemas de despacho de carga económica no convexos. Los autores encontraron que el algoritmo del PSO modificado supera a los algoritmos del PSO y EP en términos de velocidad de convergencia, tiempo de solución y la calidad de la solución. Por su parte, en la referencia [20] presenta el desempeño eficaz y fiable del PSO para resolver los problemas de optimización de emisión y despacho económico, y concluye que el

PSO encuentra la solución óptima global en comparación con algoritmos de optimización convencionales.

En la referencia [21] se propone un método de optimización de enjambre de partículas para resolver el problema del despacho económico con características no lineales del generador eléctrico, como los límites dela rampa de velocidad, la zona de operación prohibida, y las funciones de costos no-lineales. Los autores de este trabajo reportaron un desempeño eficiente del PSO. Por otra parte, en la referencia [22] se propone utilizar el PSO para resolver el problema de despacho económico dinámico limitado en la operación de un sistema de potencia eléctrico. En la referencia [23] se aborda una metodología de solución híbrida para integrar el algoritmo de optimización de enjambre de partículas con el método de programación cuadrática secuencial para la reserva restringida para el problema del despacho económico dinámico de las unidades de generación, considerando sus dinámicas no-lineales. En la referencia [24] se propone un modelo de despacho económico dinámico basado en la oferta y considerado restricciones, como rampas de velocidad, la capacidad de la línea de transmisión y las limitaciones de emisiones, con el objetivo de maximizar el beneficio social en un mercado competitivo de la electricidad. Los autores de este trabajo concluyeron que el PSO es capaz de obtener la solución de mayor calidad y de manera eficiente al problema de despacho económico.

3.2.3 Despacho de energía reactiva.

El Despacho de Energía Reactiva en sistemas eléctricos de potencia es un problema complejo de optimización combinatoria con funciones no lineales, así como de restricciones no-lineales y discontinuas. En la referencia [25] se presenta la aplicación del PSO basado en sistemas de agentes múltiples para resolver el problema del despacho de potencia reactiva. En la referencia [26] se presenta una mejora al PSO con la operación de mutación y adaptación estratégica óptima para el envío de potencia reactiva y control de voltaje en los sistemas eléctricos de potencia; los resultados obtenidos de este trabajo demostraron que el enfoque propuesto es superior a los métodos convencionales de optimización para encontrar la mejor solución del problema, tanto en términos de calidad de la solución como en la robustez del algoritmo. En la referencia [27] se presenta unla aplicación del PSO para tratar el problema de optimización de potencia reactiva en un sistema de energía provincia en China, los resultados de este trabajo indican la posibilidad de utilizar al PSO como una herramienta práctica para los diversos problemas de optimización en sistemas eléctricos de potencia.

3.2.4 Programación de mantenimientos

La Programación de mantenimientos es una planificación de interrupciones preventiva para las unidades generadoras en un sistema de potencia eléctrica dentro de un horizonte de tiempo determinado. La programación de mantenimientos se convierte en un problema complicado cuando el sistema de alimentación contiene una serie de unidades de generación con diferentes especificaciones, y cuando existen numerosas limitaciones que deben tenerse en cuenta para obtener una solución práctica y viable. En la Referencia [28] se propone al algoritmo de optimización de enjambre de partículas con el mecanismo de selección para la solución de un problema multi-objetivo de programación de mantenimientos del generador con muchas limitaciones. Los autores concluyen en este trabajo que el enfoque basado optimización enjambre de partículas es eficaz en la obtención de horarios posibles de mantenimiento en un tiempo razonable.

3.2.5 Control

El control de potencia reactiva así como el control de voltaje pueden ser formulados como problemas de optimización con variables de estado continuas tales como la regulación automática de los voltajes de operación o de variables de estado discretas como el caso de reguladores de voltaje basados en transformadores con derivaciones (conocidos en inglés como taps).

De esta forma en la referencia [29] se presenta una optimización mediante PSO para el control de potencia reactiva y de tensión considerando el suministro seguro de tensión a la carga. En este trabajo los autores formularon el problema de optimización como un problema de optimización no lineal entera mixta. Por otra parte, en la referencia [30] se aplica el PSO para resolver el problema del control de tensión y potencia reactiva en parques eólicos. Los resultados de este trabajo muestran que la aplicación del PSO es factible.

En la referencia [31] se presenta el uso del PSO para la optimización de los parámetros del sistema de control de generación de energía (AGC por sus siglas en inglés). Los resultados de este trabajo validan la eficacia del optimizador de enjambre de partículas en la sintonización de los parámetros de AGC. En las referencias [32 y 34] se propone el uso del PSO para buscar los valores óptimos de los parámetros de un estabilizador de potencia (PSS) parámetros. En la referencia [35] se presenta el uso del PSO para entrenar una red neuronal artificial multicapa para la discriminación entre la corriente de entrada de magnetización y la corriente de falla interna en un transformador de potencia.

En la referencia [36] se propone al PSO para buscar los parámetros óptimos de los controladores de un compensador síncrono estático de potencia (STATCOM por sus siglas en inglés) para un sistema eléctrico de potencia linealizado. La figura 3.3 muestra el diagrama unifilar de la conexión de un STATCOM en un sistema eléctrico de potencia.

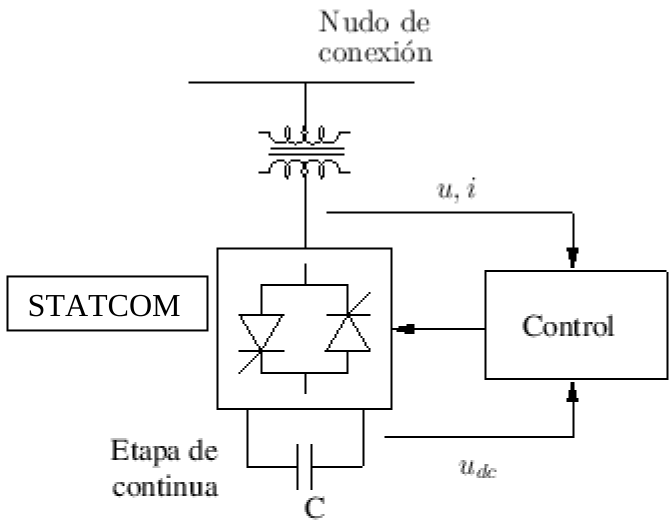


Figura 3.3 Conexión de un STATCOM en un sistema eléctrico de potencia.

Por su parte en las referencias [37 - 38] se propone el uso del PSO para diseñar un controlador difuso que se emplea en un STATCOM de tiristores para mejorar la estabilidad transitoria en sistemas de transmisión de CA flexibles (FACTs por sus siglas en inglés).

3.3 Ejemplo de aplicación del PSO en problema de optimización de despacho económico.

El problema de despacho económico consiste en la generación de energía eléctrica al menor costo pero asegurando el suministro de energía a la carga eléctrica. Si se considera que la producción base de energía eléctrica proviene de generadores eléctricos que utilizan combustibles fósiles, el problema de despacho económico se puede formular como un problema de optimización en el que se requiere minimizar el costo del combustible utilizado por los generadores eléctricos pero manteniendo el nivel de generación que permita satisfacer la demanda de la carga eléctrica así como compensar las pérdidas de transmisión. Existen diferentes formas de formular el problema de despacho económico. La formulación básica del problema de despacho económico puede ser descrita matemáticamente como un problema de minimización del costo del combustible empleado por los generadores eléctricos y sujeto a ciertas restricciones, como es descrito por:

$$Minimizar \sum_{i=1}^{n} F_i(P_i) \qquad (8)$$

donde $F_i(P_i)$ es la ecuación del costo de combustible del i-ésimo generador eléctrico y la cual será la función objetivo del problema de despacho económico. La ecuación del costo de combustible se puede expresar como un polinomio de segundo orden como es descrito por:

$$F_i(P_i) = a_i P_i + b_i P_i + c_i,\ \ P_i{}^{min} \leq P_i \leq P_i{}^{max} \qquad (9)$$

La generación total de energía debe satisfacer la demanda de energía de la carga eléctrica así como también las pérdidas de transmisión. Esta restricción se puede formular matemáticamente como:

$$\sum_{i=1}^{n} P_i = P_{carga} + P_{transmisión} \qquad (10)$$

donde las pérdidas de potencia de transmisión en cada generador se modela como un conjunto de coeficientes B_{ijm} como es expresado por:

$$P_{transmisión} = \sum_{i}^{n} \sum_{j}^{n} B_{ij} * P_i * P_j \qquad (11)$$

De esta forma, la ecuación () será la función de aptitud para el problema de optimización del PSO, como se describe en el diagrama de flujo en la Figura 3.6. Finalmente, para la encontrar la solución al problema de minimización del costo de combustible se utilizará el método de penalización de la función objetivo para tener un problema de optimización sin restricciones, como es descrito por la ecuación (12).

$$Minimizar \sum_{i=1}^{n} F_i(P_i) + K * abs(\sum_{i=1}^{n}(P_i - P_{carga} - P_{transmisión}) \qquad (12)$$

donde K es la constante de penalización y cuyo valor es ajustado heurísticamente.

En este ejemplo de aplicación del PSO para el problema de despacho económico se considera que se emplean tres generadores diesel para satisfacer una carga eléctrica constante de 150 MW. El rango de generación de potencia para cada generador se muestra en la tabla 3.1.

Tabla *3.1* Límites de potencia de los generadores eléctricos para el problema de despacho económico.

Generador Eléctrico	Potencia Mínima de generación (MW)	Potencia Máxima de Generación (MW)
1	10	85
2	10	80
3	10	70

La tabla 3.2 muestra los coeficientes de las pérdidas de transmisión para cada generador.

Tabla *3.2* Coeficientes de pérdidas de transmisión por generador eléctrico.

Generador Eléctrico	B_1	B_2	B_3
1	$2.18x10^{-4}$	$9.3x10^{-5}$	$2.8x10^{-5}$
2	$9.3x10^{-5}$	$2.28x10^{-4}$	$1.7x10^{-5}$
3	$2.8x10^{-5}$	$1.7x10^{-5}$	$1.79x10^{-4}$

3.3.1 Programación del PSO

Aunque en este trabajo se ha mencionado que el algoritmo del PSO es muy sencillo y por lo tanto muy fácil de programar, es recomendable utilizar programas ya codificados por expertos en ciencias de la computación, ya que estos programas están optimizados para el uso de los recursos computacionales disponibles. Al utilizar programas ya codificados se reduce el tiempo de aplicación del PSO en problemas de optimización.

En internet se pueden encontrar de forma gratuita diferentes programas del PSO codificados en diferentes lenguajes como C, C++, Python, Matlab, por mencionar algunos ejemplos. Al utilizar estos programas ya codificados, el usuario sólo se tiene que enfocar en la formulación del problema de optimización para que el PSO encuentre la solución óptima. De esta forma, en este trabajo se utiliza el programa del PSO para ser ejecutado en Matlab, desarrollado por el profesor Brian Birge, uno de los pioneros en el ámbito del PSO. El código del PSO es libre y puede ser encontrado en la dirección electrónica de la referencia [39].

Es conveniente señalar que en el programa del PSO codificado en Matlab requiere que el conjunto de datos de variables de entrada sean introducidos de forma vectorizada, debido a que Matlab tiene un mejor desempeño al trabajar con datos en forma vectorizada. Cómo un recurso adicional, el programa del PSO genera una animación gráfica del movimiento dinámico de las partículas del PSO mientras se encuentra en la búsqueda de la solución óptima al problema de minimización.

Los parámetros de control para el PSO empleados en la solución del problema de despacho económico se muestran en la tabla 3.3.

Tabla *3.3* Límites de potencia de los generadores eléctricos para el problema de despacho económico.

Parámetro	Valor
Número de partículas	100
Constante de aceleración C_1	2
Constante de aceleración C_2	2
Valor inicial del peso inercial W	0.9
Valor final del peso inercial W	0.4
Número máximo de iteraciones	100000
Dimensión del problema	3

En la Figura 3.4 se muestra la información del desempeño del PSO en la solución del problema de despacho económico, durante las primeras 10 iteraciones.

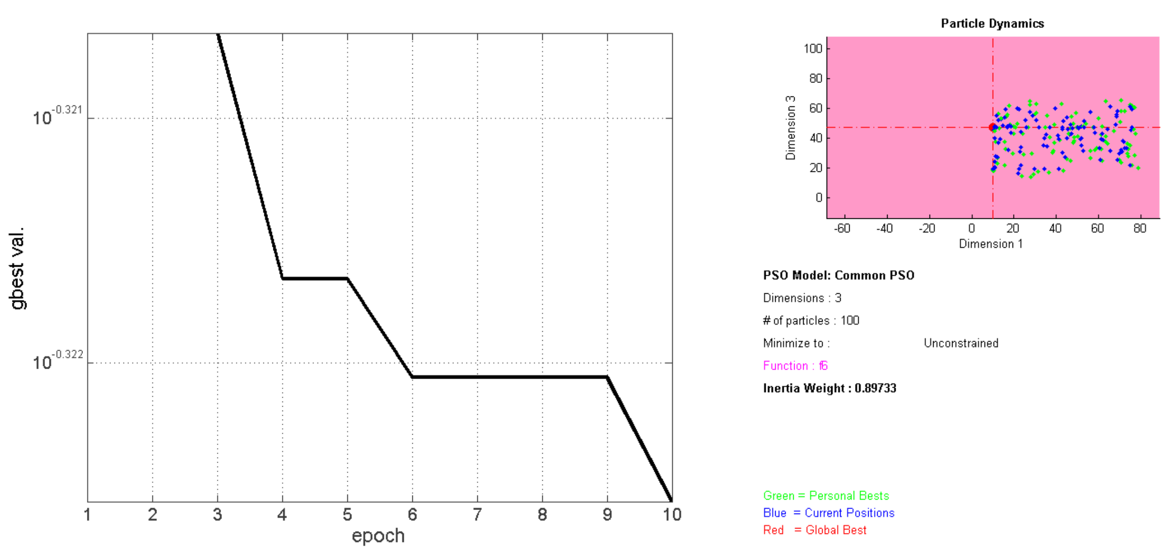
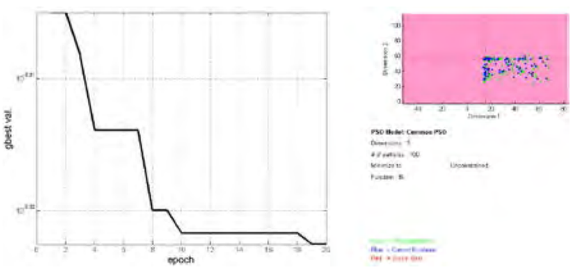


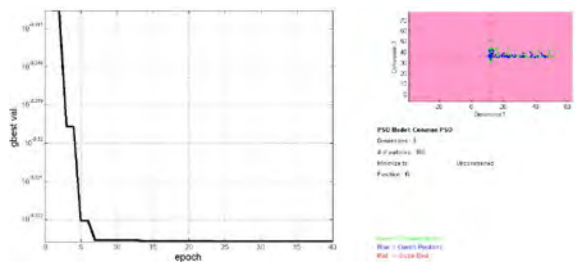
Figura 3.4 Desempeño del PSO durante 10 iteraciones.

En la Figura 3.4 se observa en su lado izquierdo, la gráfica del comportamiento del valor del mejor global de todas las partículas del PSO. En la esquina superior izquierda de la Fig. 3.4 se observa la gráfica de la posición de las partículas alrededor del valor del mejor global (el mejor personal de cada partícula). De igual forma, en la Figura 3.4 se informa al usuario de la dimensión del problema, que en este caso es de 3 porque corresponde a los 3 generadores eléctricos. El número de partículas utilizadas en el PSO (100 partículas) y el valor del peso inercial (*w*) que inicialmente tiene un valor de 0.9.

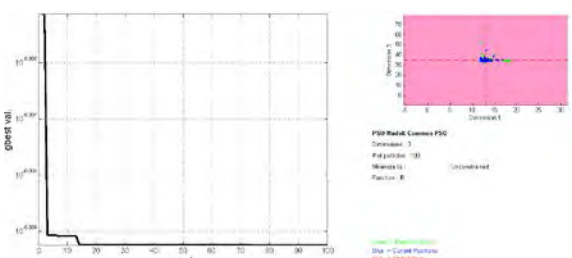
Por otra parte, en la Figura 3.5 se muestra una secuencia de la animación del movimiento de las partículas del PSO y su convergencia en la solución óptima al problema de despacho económico número de iteraciones.



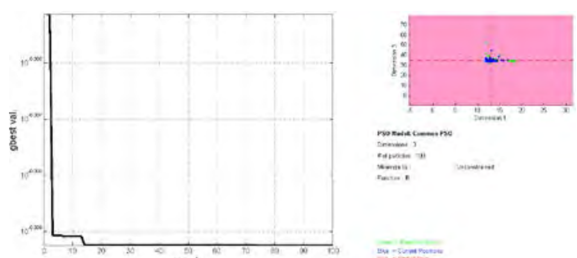
(a)



(b)



(c)



(d)

Figura 3.5 Desempeño del PSO. (a) 20 iteraciones. (b) 40 iteraciones. (c) 100 iteraciones. (d) 1000 iteraciones.

En la Figura 3.5 se observa como el PSO converge a la solución óptima prácticamente después de 30 iteraciones, posteriormente el valor del mejor global no disminuye en mayor medida aunque se tengan más iteraciones. Por otra parte, también se observa en la dinámica de las partículas que conforme el número de iteraciones aumenta, estas tienden a converger hacia la partícula que reporta el mejor global. Es importante hacer notar que el valor del peso inercial varía desde el valor inicial estipulado (0.9) hasta el valor final estipulado (0.4), con la finalidad de ir reduciendo el movimiento de las partículas conforme se acercan al mejor global.

La solución óptima al problema de despacho económico determinado por el PSO en valores por unidad se muestra en la tabla 3.4.

Tabla *3.3* Solución óptima determinada por el PSO para el problema de despacho económico.

Parámetro	Valor
Potencia Generador 1	32.88 MW
Potencia Generador 2	64.59 MW
Potencia Generador 3	55.10 MW
Costo combustible	1599.98

Con base en los valores determinados para la potencias que los generadores eléctricos deben producir, se observa como la suma de las potencias individuales de cada generador suman la potencia demandada por la carga eléctrica, con lo cual se demuestra que el PSO satisface la restricción de minimizar los costos de producción de energía eléctrica pero sin dejar a la carga eléctrica sin energía.

CAPÍTULO 4

CAPÍTULO 4

CONCLUSIONES

En este trabajo monográfico se ha presentado el principio básico de funcionamiento del optimizador de enjambre de partículas y su aplicación en la solución de problemas de optimización en el área de sistemas eléctricos de potencia.

En principio, los problemas de optimización están presentes en diversas áreas del conocimiento como las áreas económicas, finanzas, sociales, educación y el área de ingeniería no es la excepción. Debido a esto, en el trabajo monográfico se presentó la terminología básica y descripción de un problema de optimización con la finalidad de fijar las bases sobre las cuales el algoritmo de optimización debe de operar.

De igual forma, en el trabajo monográfico se aborda de forma general la descripción y clasificación de los algoritmos de optimización más comúnmente utilizados en la solución de problemas de optimización. Dentro de esta clasificación, los algoritmos de optimización evolutivos han ganado preferencia de uso en la solución de problemas de optimización complejos debido a que no requieren de un modelado matemático detallado de las relaciones que guardan las variables a optimizar, lo cual facilita su empleo por usuarios que no tengan un conocimiento profundo en el área de optimización. Debido a esta característica, el uso de los algoritmos de optimización evolutivos se ha generalizado en diversas áreas del conocimiento.

El optimizador de enjambre de partículas es un algoritmo evolutivo cuyo principio de funcionamiento está basado en el aprendizaje cooperativo de individuos como son los enjambres de insectos, las parvadas de aves o bancos de peces. Los individuos del optimizador se les denomina partículas y cada partícula se “mueve” en un espacio de búsqueda basado en tres componentes: la trayectoria inicial, la mejor solución individual encontrada por la propia partícula y la mejor solución global encontrada por alguna partícula del enjambre.

Aunque existen diferentes algoritmos de optimización evolutivos como los algoritmos genéticos, colonia de hormigas, evolución diferencial por mencionar algunos de ellos; el optimizador de enjambre de partículas presenta como ventaja con respecto a los algoritmos mencionados en que utiliza un menor número de parámetros de control para su operación así como también presenta una rápida convergencia hacia la solución óptima. Estas características han permitido el creciente uso del optimizador de enjambre de partículas en diversas áreas de las ingenierías. Debido a ello, en este trabajo se presentó de manera particular el uso del optimizador de enjambre de partículas en la solución de problemas de optimización comunes en el área de sistemas eléctricos de potencia por su estrecha relación con el perfil de la carrera de Ingeniería en Sistemas de Energía.

REFERENCIAS BIBLIOGRÁFICAS

[1] A.P. Engelbrecht. "Computational Intelligence, an Introduction". Editorial Wiley and sons. Segunda edición. USA 2007. pp. 597

[2] *A. Qing. "Differential evolution, Fundamentals ans applications in Electrical Engineering".Editorial Wiley and sons. USA 2009. pp. 405* .

[3] D. C. Walters and G. B. Sheblé, "Genetic algorithm solution of economic dispatch with valve point loading," IEEE Transactions on Power Systems, Vol.8, No. 3, August 1993, pp. 1325-1332.

[4] Thomas Back, "Evolutionary Algorithms in Theory and Practice," Oxford University Press, New York, 1996

[5] M. Clerc. "Particle Swarm Optimization". Editorial ISTE. Segunda edición. USA 2006. pp. 243.

[6]. J. Kennedy, y R. C. Eberhart, "Swarm intelligence," Morgan Kaufmann Publisher, 2001

[7] R. C. Eberhart, y J. Kennedy, "A new optimizer using particle swarm theory," Proc. Sixth International Symposium on Micro Machine and Human Science (Nagoya, Japan), 1995, IEEE Service Center, Piscataway, NJ, pp. 39-43.

[8] J. Kennedy y R. C. Eberhart, "Particle swarm optimization," Proc. IEEE International Conference on Neural Networks (Perth, Australia), 1995, IEEE Service Center, Piscataway, NJ, pp. IV: pp. 1942-1948.

[9] B. Chao, C. X. Guo, Y. J. Cao, "Improved particle swarm optimization algorithm for OPF problems," Proceedings of 2004 IEEE PES Power Systems Conference and Exposition, vol. 1, pp. 233-238, 2004.

[10] S. He, J. Y. Wen, E. Prempaint, Q. H. Wu, J. Fitch, S. Mann, "An improved particle swarm optimization for optimal power flow," Proceedings of 2004 International Conference on Power System Technology, vol. 2, pp. 1633-1637, 2004.

[11] Zwe-Lee Gaing, "Constrained optimal power flow by mixed-integer particle swarm optimization," Proceedings of 2005 IEEE Power Engineering Society General Meeting, vol. 1, pp. 243-250, 2005.

[12] Cui-Ru Wang, He-Jin Yuan, Zhi-Qiang Huang, Jiang-Wei Zhang, Chen-Jun Sun, "A modified particle swarm optimization algorithm and its application in optimal power flow problem,"

Proceedings of 2005 International Conference on Machine Learning and Cybernetics, vol. 5, pp. 2885-2889, 2005.

[13] Ahmed A. A. Esmin, Germano Lambert-Torres and C. Zambroni de Souza, "A hybrid particle swarm optimization applied to loss power minimization," IEEE Transactions on Power Systems, vol. 20, no. 2, pp. 859-866, 2005.

[14] Chowdhury B.H., Rahman S., "A review of recent advances in economic dispatch," IEEE Transactions on Power Systems, vol. 5, no. 4, pp. 1248-1259, 1990.

[15] Jong-Bae Park, Ki-Song Lee, Joong-Rin Shin, and Kwang Y. Lee, "A particle swarm optimization for economic dispatch with nonsmooth cost functions," IEEE Transactions on Power Systems, vol. 20, no. 1, pp. 34- 42, 2005.

[16] Jong-Bae Park, Ki-Song Lee, Joong-Rin Shin, and Kwang Y. Lee, "Economic load dispatch for nonsmooth cost functions using particle swarm optimization," Proceedings of 2003 IEEE Power Engineering Society General Meeting, vol. 2, pp. 938-943, 2003.

[17] Rohit Kumar Pancholi, "Particle swarm optimization for economic dispatch with line flow and voltage constraints," Proceedings of Conference on Convergent Technologies for Asia-Pacific Region, vol. 1, pp. 450-455, 2003.

[18] Rohit Kumar Pancholi, K.S.Swarup, "Particle swarm optimization for security constrained economic dispatch," Proceedings of International Conference on Intelligent Sensing and Information Processing, pp. 7-12, 2004

[19] Nidul Sinha, B. Purkayastha, "PSO embedded evolutionary programming technique for non-convex economic load dispatch," Proceedings of 2004 IEEE PES Power Systems Conference andExposition, vol. 1, pp. 66-71, 2004.

[20] A. Immanuel Selva Kumar, K.Dhanushkodi, J.Jaya Kumar, C.Kumar Charlie Paul, "Particle swarm optimization solution to emission and economic dispatch problem," Proceedings of Conference on Convergent Technologies for Asia-Pacific Region, vol. 1, pp. 435-439, 2003.

[21] Zwe-Lee Gaing,"Particle swarm optimization to solving the economic dispatch considering the generator constraints," IEEE Transactions Power Systems, vol. 18, no. 3, pp. 1187-1195, 2003.

[22] Zwe-Lee Gaing, "Constrained dynamic economic dispatch solution using particle swarm optimization," Proceedings of 2004 IEEE Power Engineering Society General Meeting, vol. 1, pp. 153-158, 2004.

[23] T. Aruldoss Albert Victoire, and A. Ebenezer Jeyakumar, "Reserve constrained dynamic dispatch of units with valve-point effects," IEEE Transactions on Power Systems, vol. 20, no. 3, pp. 1273-1282, 2005.

[24] Bo Zhao, Chuangxin Guo and Yijia Cao, "Dynamic economic dispatch in electricity market using particle swarm optimization algorithm," Proceedings of the 5th World Congress on Intelligent Control and Automation, vol. 6, pp. 5050-5054, 2004.

[25] B. Zhao, C. X. Guo, and Y. J. Cao, "A multiagent-based particle swarm optimization approach for optimal reactive power dispatch," IEEE Transactions on Power Systems, vol. 20, no. 2, pp.1070-1078, 2005.

[26] B. Zhao, C. X. Guo, and Y. J. Cao, "An improved particle swarm optimization algorithm for optimal reactive power dispatch," Proceedings of 2005 IEEE Power Engineering Society General Meeting, vol. 1, pp. 272-279, 2005.

[27] Wen Zhang,Yutian Liu,"Reactive power optimization based on PSO in a practical power system" Proceedings of 2004 IEEE Power Engineering Society General Meeting, vol. 1, pp. 239-243, 2004.

[28] Chin Aik Koay and Dipti Srinivasan, "Particle swarm optimizationbased approach for generator maintenance scheduling," Proceedings of the 2003 IEEE Swarm Intelligence Symposium, pp.167-173, 2003.

[29] Hirotaka Yoshida, Kenichi Kawata, Yoshikazu Fukuyama, Shinichi Takayama, and Yosuke Nakanishi, "A particle swarm optimization for reactive power and voltage control considering voltage security assessment," IEEE Transactions on Power Systems, vol. 15, no. 4, pp. 1232-1239, 2000.

[30] Genevieve Coath, Majid Al-Dabbagh, Saman K. Halgamuge, "Particle swarm optimization for Reactive power and voltage control with gridintegrated wind farms," Proceedings of 2004 IEEE Power Engineering Society General Meeting, vol. 1, pp. 303-308, 2004.

[31] Youssef L. Abdel-Magid, M.A. Abido, "AGC tuning of interconnected reheat thermal systems with particle swarm optimization," Proceedings of the 2003 10th IEEE International Conference on Electronics, Circuits and Systems, vol. 1, pp. 376-379, 2003.

[32] Ali Karimi,Amer Al-Hinai,Karl Schoder,Ali Feliachi, "Power system stability enhancement using backstepping controller tuned by particle swarm optimization technique," Proceedings of 2005 IEEE Power Engineering Society General Meeting, vol. 2, pp. 1388-1395, 2005.

[33] M. A. Abido, "Optimal design of power-system stabilizers using particle swarm optimization," IEEE Transactions on Energy Conversion, vol. 17, no. 3, 2002.

[34] M. A. Abido,"Particle swarm optimization for multimachine power system stabilizer design," Proceedings of 2001 IEEE Power Engineering Society Summer Meeting, vol. 3, pp. 1346-1351, 2001.

[35] A. I. El-Gallad, M. El-Hawary, A. A. Sallam, A. Kalas, "Swarmintelligently trained neural network for power transformer protection," Proceedings of Canadian Conference on Electrical and Computer Engineering, vol. 1, pp. 265-269, 2001.

[36] S. M. Bamasak,M. A. Abido,"Aseessment study of shunt FACTS-based controllers effectiveness on power system stability enhancement," Proceedings of the 39th International Universities

[37] Chun-Feng Lu,Chia-Feng Juang, "Control of flexible AC transmission system by evolutionary fuzzy controller," Proceedings of 2004 IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 2292-2296, 2004.

[38] Chun-Feng Lu and Chia-Feng Juang, "Evolutionary fuzzy control of flexible AC transmission system," IEE Proceedings of Generation, Transmission and Distribution, vol. 152, no. 4, pp. 441-448, 2005.

[39] http://www.mathworks.com/matlabcentral/fileexchange/7506-particle-swarm-optimization-toolbox.