

Análisis comparativo de técnicas metaheurísticas para resolver el planeamiento de la transmisión

CARLOS ADRIÁN CORREA FLÓREZ¹

GUILLERMO ALBERTO CAMACHO MUÑOZ²

RICARDO ANDRÉS BOLAÑOS³

MAURICIO GRANADA ECHEVERRI⁴

RESUMEN

Este artículo presenta la solución del problema de planeamiento de la expansión de la transmisión (PET) de sistemas eléctricos de potencia (SEP) usando el modelo DC, mediante diferentes técnicas de optimización metaheurísticas (TOM). El problema operativo de flujo de carga es resuelto mediante el método de puntos interiores. Las técnicas analizadas son: algoritmos genéticos, optimización por colonia de hormigas, enjambre de partículas y artificial *bee colony*. Se incluye, además, en cada algoritmo un esquema aleatorio controlado para inicialización de la población y una etapa de mejoramiento de la optimalidad. Los algoritmos modificados propuestos se prueban sobre los sistemas de Garver de 6 nodos y el sistema IEEE de 24 nodos. Se realiza un análisis comparativo de los métodos desde el punto de vista del esfuerzo computacional.

Palabras clave: planeamiento de la transmisión, optimización, metaheurísticas, modelo DC.

¹ Magíster en Ingeniería Eléctrica por la Universidad Tecnológica de Pereira, Colombia. Ingeniero eléctrico por esta misma universidad. Correo electrónico: carcorrea@unisalle.edu.co

² Magíster en Sistemas Mecatrónicos por la Universidad de Brasilia, Brasil. Ingeniero en Automática Industrial por la Universidad del Cauca, Popayán, Colombia. Correo electrónico: gacamacho@unisalle.edu.co

³ Magíster en Ingeniería Eléctrica por la Universidad Tecnológica de Pereira, Colombia. Ingeniero eléctrico por esta misma universidad. Correo electrónico: rabolanos@xm.com.co

⁴ Doctor en Ingeniería Eléctrica por la Universidade Estadual Paulista Júlio de Mesquita Filho, São Paulo, Brasil. Magíster en Ingeniería Eléctrica por la Universidad Tecnológica de Pereira, Colombia. Ingeniero eléctrico por esta misma universidad. Correo electrónico: magra@utp.edu.co

FECHA DE RECEPCIÓN: 17 DE ENERO DEL 2014 • FECHA DE APROBACIÓN: 17 DE MAYO DEL 2014

Cómo citar este artículo: Correa Flórez, C. A., Camacho, G., Bolaños, R. A. y Granada, M. (2014). Análisis comparativo de técnicas metaheurísticas para resolver el planeamiento de la transmisión. *Épsilon* (22), 81-105.

Comparative Analysis of Metaheuristic Techniques for Solving Transmission Planning

ABSTRACT

This paper presents the solution to the problem of transmission expansion planning (TEP) for electric power systems (EPS), using the DC model through different metaheuristic optimization techniques (MOT). The operational load flow problem is solved by the interior point method. The techniques discussed are: genetic algorithms, ant colony optimization, particle swarm, and artificial bee colony. Additionally, a controlled randomized scheme is included in each algorithm for population initialization, and a phase of improving optimality. The proposed modified algorithms are tested on Garver's 6-node system and on the IEEE 24-node system. A comparative analysis of methods is carried out from the point of view of computational effort.

Keywords: Transmission planning, optimization, metaheuristic, DC model.

Análise comparativa de técnicas meta-heurísticas para resolver o planejamento da transmissão

RESUMO

Este artigo apresenta a solução do problema de planejamento da expansão da transmissão (PET) de sistemas elétricos de potência (SEP) usando o modelo DC, através de diferentes técnicas de otimização meta-heurísticas (TOM). O problema operativo de fluxo de carga se resolve através do método de pontos interiores. As técnicas analisadas são: algoritmos genéticos, otimização por colônia de formigas, enxame de partículas e artificial *bee colony*. Inclui-se, além do mais, em cada algoritmo um esquema aleatório controlado para inicialização da população, e uma etapa de melhoramento da otimalidade. Os algoritmos modificados propostos se provam sobre os sistemas de Garver de 6 nodos e o sistema IEEE de 24 nodos. Realiza-se uma análise comparativa dos métodos desde o ponto de vista do esforço computacional.

Palavras chave: planejamento da transmissão, otimização, meta-heurísticas, modelo DC.

Introducción

El problema del planeamiento de la expansión de la transmisión (PET) consiste en determinar el plan de expansión de mínimo costo, teniendo en cuenta las restricciones operativas de la red. Dicho plan permite calcular la cantidad, la ubicación y el tiempo para la instalación de los diferentes equipos. Esto permite abordar el problema desde diversos puntos de vista: el modelo estático (Villasana *et al.*, 1985; Hemmati *et al.*, 2013), que presenta un único horizonte de planeamiento, y el modelo dinámico para varias etapas independientes (Zhanget *et al.*, 2012; Rocha y Saraiva, 2013). El modelamiento se puede clasificar también de acuerdo con el grado de complejidad: modelos de transportes (Garver, 1970), DC (Escobar *et al.*, 2004; Sánchez *et al.*, 2005), híbrido (Romero *et al.*, 2002) y AC (Rider *et al.*, 2007).

En este artículo se aborda el problema de la expansión de la transmisión mediante el modelo DC, que resulta en un problema de optimización del tipo no lineal entero mixto (PNLEM). Existen múltiples propuestas para resolver dicho problema empleando programación lineal, programación dinámica, programación no lineal, programación entera mixta, *benders* y técnicas de descomposición como *branch and bound* (Villasana *et al.*, 1985; Kim *et al.*, 1988; Bahiense *et al.*, 2001; Haffner *et al.*, 2000; Romero *et al.*, 2007; (Romero y Monticelli, 1994; Binato *et al.*, 2001). Además de las técnicas clásicas, los últimos años se han caracterizado por el uso de técnicas metaheurísticas para resolver este problema. Dentro de dichas técnicas se encuentran el recocido simulado, búsqueda tabú, algoritmos genéticos y enjambre de partículas (Escobar *et al.*, 2004; Gallego *et al.*, 1998; Hemmati *et al.*, 2013; Karaboga y Basturk, 2007).

El problema del PET se puede resolver teniendo en cuenta restricciones de mercado, a través de múltiples escenarios de generación, costos del despacho, costo anual de la transmisión, precios nodales, costos de congestión, entre otros. También puede tener en cuenta restricciones de seguridad a través del análisis de contingencias $N-\alpha$ (determinístico) o a través de enfoques probabilísticos usando tasas de falla para asegurar confiabilidad.

En este artículo se desarrolla un análisis comparativo de las siguientes técnicas para resolver el PET: algoritmos genéticos (AG), optimización por colonia de hormigas (OCH), enjambre de partículas o *particles swarm optimization* (PSO) y *artificial bee colony* (ABC). Cada propuesta de inversión generada por la metaheurística es

evaluada operativamente mediante un Método de Punto Interior Lineal (MPIL). Este determina el nivel de infactibilidad o corte de carga, para generar un esquema iterativo entre el problema de inversión y el problema operativo.

Se presentan los resultados obtenidos al implementar las diferentes técnicas en el sistema de seis nodos propuesto por Garver y sobre el sistema IEEE de 24 barras. El desempeño de cada técnica se mide a través del número de subproblemas operativos que debe resolver para obtener su mejor respuesta en un número determinado de pruebas. Para todas las técnicas bajo prueba, se incluye una metodología para inicialización aleatoria controlada y una etapa de mejoramiento de la factibilidad.

El artículo está organizado de la siguiente manera: la segunda sección presenta la formulación matemática del problema. Después, la sección tercera hace referencia a cada una de las técnicas utilizadas y su adaptación para el problema de PET. Los resultados y el respectivo análisis se muestran en la sección cuarta. Finalmente, se enuncian las conclusiones en la sección quinta.

Formulación del problema

El modelo matemático usado para el planeamiento de la expansión de la transmisión, basado en el modelo DC (Romero *et al.*, 2002), asume la siguiente forma para minimizar costos de inversión:

$$\min v = \sum_{ij \in \Omega} c_{ij} n_{ij} + \alpha \sum_{i \in N_b} r_i \quad (1)$$

s.t.

$$Sf + g + r = d \quad (2)$$

$$f_{ij} - \gamma_{ij} (n_{ij}^0 + n_{ij}) (\theta_i - \theta_j) = 0 \quad (3)$$

$$|f_{ij}| \leq (n_{ij}^0 + n_{ij}) \bar{f}_{ij} \quad (4)$$

$$0 \leq g \leq \bar{g} \quad (5)$$

$$0 \leq r \leq d \quad (6)$$

$$0 \leq n_{ij} \leq \bar{n}_{ij} \quad (7)$$

$$ij \in \Omega, n_{ij}^i \text{ integer} \quad (8)$$

En el conjunto de ecuaciones (1) a (8), los subíndices ij representan las barras i y j del SEP, y las variables que hacen referencia a estas barras ij se describen a continuación: c_{ij} es el costo de instalación de equipos, f_{ij} es el flujo de potencia activa, γ_{ij} es la susceptancia eléctrica, n_{ij} es el número de circuitos adicionales, n_{ij}^0 es el número de circuitos del caso base, \bar{f}_{ij} es el flujo máximo de potencia activa y \bar{n}_{ij} representa el número máximo de circuitos que deben ser adicionados por el corredor ij . S es la transpuesta de la matriz de incidencia elemento nodo, f es un vector cuyos elementos son los f_{ij} , g , d y r son vectores con las generaciones, las demandas y los cortes de carga nodales, respectivamente, θ_i es el ángulo en el nodo i , y Ω es el conjunto de ramas candidatas y N_b el número de nodos. Finalmente, α es un factor de penalización para asegurar planes de expansión factibles, es decir, sin corte de carga.

La ecuación (2) representa la conservación de potencia en cada nodo, es decir, la primera ley de Kirchhoff de la red DC equivalente. La ecuación (3) modela la segunda ley de Kirchhoff. El esquema de planeamiento mediante el modelo DC suele dividirse en dos subproblemas, donde un algoritmo combinatorial realiza una propuesta de inversión, número de líneas por ser adicionadas, y el subproblema operativo es convertido en un problema de programación lineal (PL). Este PL se resuelve mediante el MPIL, que es explicado ampliamente en Correa (2008).

Adaptación de las metaheurísticas al problema de PET

A continuación se presentan los conceptos básicos de cada una de las técnicas utilizadas y su adecuación para resolver el problema de PET.

Es importante mencionar que para todas las metaheurísticas se utilizó una codificación entera decimal, donde cada bit o posición de una solución representa el número de líneas que se van a instalar. Así, para un sistema básico de tres nodos y tres ramas, la propuesta de inversión 1-0-2 indica que se deben construir una, cero y dos líneas de transmisión, respectivamente, en cada una de las tres ramas. Además, otra característica que comparten las alternativas analizadas es la penalización en la función objetivo, donde el costo de una propuesta de inversión es penalizada de forma proporcional a su corte de carga. La función resultante es denominada *fitness*. Lo anterior asegura que las propuestas infactibles sean descartadas a través del proceso iterativo.

Vale la pena resaltar que uno de los aportes más importantes del enfoque presentado es la manera en la que es generada la población inicial. Esta etapa es crítica debido a que puede guiarse el algoritmo hacia zonas donde se encuentren soluciones de buena calidad. En el presente enfoque, la inicialización se desarrolla de acuerdo con la siguiente metodología:

- Un número predefinido ρ_{ini} de ramas es considerado para ubicar equipos de transmisión.
- Para cada individuo, estas posiciones se seleccionan de forma aleatoria para asignar un número de líneas entre cero y \bar{n}_{ij} .
- Después de que la población es generada, se realiza verificación de diversidad. Este procedimiento se realiza al comparar todas las soluciones entre sí y forzando las soluciones para que sean diferentes en, por lo menos, ρ_{div} posiciones.

Esta inicialización asegura que la población inicial quede esparcida por el espacio de soluciones, para identificar circuitos de alta importancia.

Otro elemento transversal a los algoritmos implementados es la etapa de mejoramiento donde realiza una mejora de la optimalidad de las soluciones resultantes de cada iteración, retirando circuitos redundantes. Los circuitos se ordenan de acuerdo con su costo de forma decreciente, y se retiran temporalmente para verificar el corte de carga de la configuración temporal. Si alguna de las configuraciones sigue siendo factible, dicho circuito se retira de forma permanente. Este procedimiento se describe también en el siguiente pseudocódigo:

Algoritmo 1. Etapa de Mejoramiento

if Q factible **then**

$Ordered \leftarrow$ Ordenar circuitos en forma descendente

$Q_{original} \leftarrow Q$

for $k = k \in \Omega$ **do**

$Q(Ordered(k)) \leftarrow Q_{original}(Ordered(k)) - 1$

if Q infactible **then**

$Q(\text{Ordered}(k)) \leftarrow Q_{\text{original}}(\text{Ordered}(k))$

end if

$k \leftarrow k + 1$

end for

end if

Algoritmo genético

El algoritmo genético (AG) utilizado en este trabajo es el propuesto por Chu y Beasley (1997), originalmente creado para resolver el problema de asignación generalizada. El AG de Chu-Beasley (CBGA) presenta varias ventajas al compararse con el tradicional, como el mantener un tamaño de población constante y así analizar solo un descendiente en cada ciclo generacional. De esta manera, se reduce el número de veces que debe ser evaluada la función objetivo en cada iteración.

Selección

En el proceso de selección, dos torneos son realizados para seleccionar dos padres. En cada torneo existe un número k de padres por ser elegido, que está directamente relacionado con el grado de elitismo del CBGA. En cada torneo, la configuración con menor *fitness* gana un derecho como padre. Así, después de dos torneos, dos padres están disponibles para la etapa de cruzamiento.

Cruzamiento

Este operador genético es implementado de la manera tradicional, a través del intercambio de información genética en un único punto. Es importante anotar que en el AGCB no es necesario definir una probabilidad de cruzamiento, dado que la población no cambia, y un individuo es incluido en cada ciclo solo si cumple con criterios de optimalidad y factibilidad. Este hecho difiere del AG tradicional, donde toda la población se cambia en cada ciclo, siendo interesante dejar algunos

individuos intactos. Después del cruzamiento, dos hijos son generados, y se descarta uno de ellos de manera aleatoria.

Mutación

En esta etapa, ρ_{mut} caminos son elegidos aleatoriamente para remover o adicionar un circuito. La decisión de eliminar un circuito se toma con un 80% de probabilidad. La probabilidad de mutar se considera alta, debido a que ello permite aumentar el nivel de diversidad de los individuos y, por tanto, de la población en general para que el algoritmo de búsqueda explore en regiones más diversas, nuevas y posibles soluciones; mientras que para los casos en que son menores las probabilidades, el algoritmo tiende a homogeneizar la población en torno a las soluciones de mejor calidad, lo que eventualmente podría atraparlos en soluciones mínimas locales sin alcanzar el óptimo global del problema.

Promoción de un individuo

Una de las características que hace el AGCB es la preservación de la diversidad. Este hecho asegura una búsqueda en diferentes lugares del espacio de solución y aumenta las posibilidades de convergencia del algoritmo a óptimos de buena calidad. Por lo anterior, es importante controlar la inclusión de un nuevo individuo en la población para evitar la homogeneización y la convergencia a óptimos locales. Para la inclusión de un individuo se exige que este sea diferente en más de ρ_{prom} bits, al compararse con cada uno de los miembros de la población. Si el individuo es diverso y su función *fitness* es mejor que la peor de la población actual, el peor individuo es reemplazado por el candidato. El esquema general del AGCB propuesto se muestra en la figura 1.

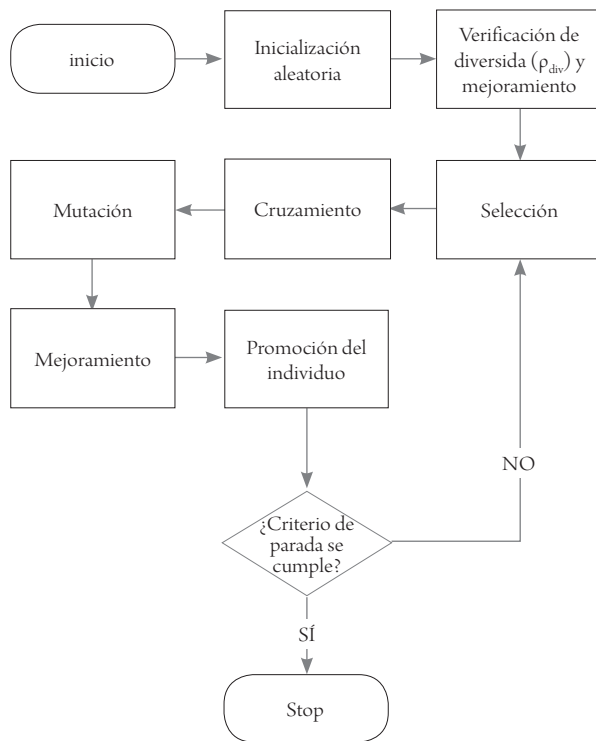


Figura 1. Esquema general de AGBC

Fuente: elaboración propia.

Optimización por colonia de hormigas

Haciendo referencia a la técnica de colonia de hormigas, en Colorni *et al.* (1994) se plantean diversos métodos para inicializar la matriz de feromona. En este artículo se utiliza la siguiente expresión:

$$\forall (i, j) \tau_{ij} = \tau_0 = \frac{\# \text{ hormigas}}{C^{nm}} = \frac{m}{C^{nm}} \quad (9)$$

$$C^{nm} = \begin{cases} \text{Longitud de la mejor alternativa} \\ \text{encontrada por una heurística en} \\ \text{la iteración actual} \end{cases}$$

Donde m es el número de hormigas usado en el proceso de optimización. La estructura de esta matriz debe permitir almacenar información de caminos y número de líneas. Así, por ejemplo, para un sistema como el mostrado en la figura 2, donde

las líneas continuas representan el sistema de transmisión actual o caso base y las punteadas las posibles modificaciones indican el sistema o coárbol, la matriz de feromonas posee la configuración allí mostrada.

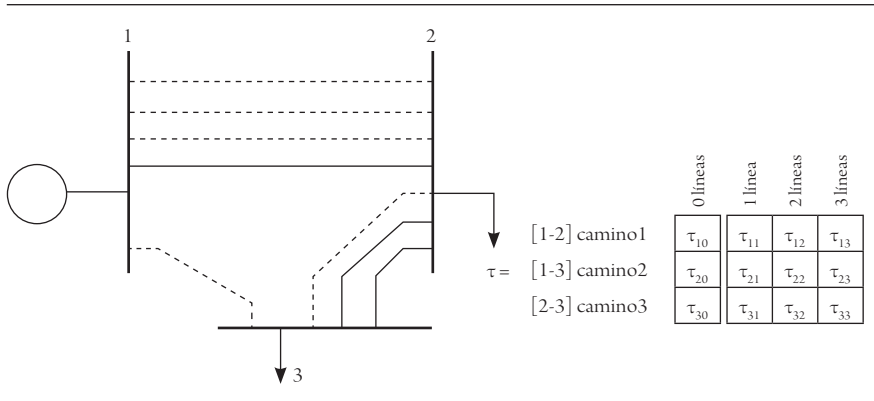


Figura 2. Matriz de feromona

Fuente: elaboración propia.

Otra matriz importante, que debe ser calculada una sola vez al inicio del proceso, es la de sensibilidades que relaciona los cambios en el valor de la función objetivo frente a la colocación de líneas partiendo del caso base. Esta matriz permitirá darle un valor asociado a cada arco relacionado fuertemente con el valor de la función objetivo. Su estructura es idéntica a la de la mostrada en la figura 2 y se llena calculando el valor de la función objetivo para cada caso posible. Como el problema es de minimización, se debe considerar que $S_{ij} = 1 / Fobj_{ij}$.

Para calcular la probabilidad con la cual una hormiga escoge un arco cualquiera de un grafo completo se utiliza lo planteado en Colorni *et al.* (1994):

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} \text{ si } j \in N_i^k \quad (10)$$

Donde p_{ij}^k es la probabilidad de que la hormiga k escoja el arco comprendido entre los vértices i y j ; τ_{ij} corresponde a la feromona depositada por la hormiga en el arco comprendido entre los vértices i y j ; η_{ij} es el valor de la importancia relativa de cada arco frente al valor de la función objetivo total. Para la selección de los parámetros se suelen realizar varias corridas del algoritmo, con el fin de identificar parámetros de buen desempeño y se muestran en el capítulo de resultados.

Construcción de las alternativas de solución

Para la construcción de las alternativas de solución se realizan dos etapas, una horizontal y otra vertical explicadas a continuación:

Etapla horizontal. Una vez la hormiga se encuentra sobre un camino, se escoge el número de líneas por instalar en este camino, usando la expresión (10) y teniendo en cuenta que $\eta_{ij} = S_{ij}$. Este procedimiento se aplica sobre un número de caminos predeterminado, lo cual implica que se obtiene un conjunto de caminos, y en cada uno de ellos se da una sugerencia de un número de líneas por instalar. El siguiente paso consiste en escoger de este conjunto un único camino, lo cual se realiza en la etapa vertical.

Etapla vertical. En esta etapa se usan, para cada uno de los caminos escogidos en la etapa anterior, los costos asociados a la instalación del número de líneas sugerido. Nuevamente, se utiliza la expresión (10), haciendo η_{ij} igual a los costos de instalación de las líneas. Se selecciona, usando hormigas artificiales, un camino y un número de líneas por ser instalado. Este camino se considera que ya ha sido visitado por la hormiga y no participa de los siguientes procesos de selección. La estrategia se repite hasta que todos los caminos hayan sido visitados.

Búsqueda local

El proceso de intensificación consiste en realizar un recorrido por todos los caminos de la alternativa de solución actual, haciendo de forma aleatoria incrementos o decrementos en la instalación de una línea. La búsqueda local se activa cada vez que se mejore la incumbente (mejor solución conocida hasta el momento) y puede ser realizada más de una vez.

Actualización de la matriz de feromonas

El primer paso consiste en evaporar una cierta cantidad de feromona sobre todos los elementos de la matriz:

$$\begin{aligned} \tau_{ij} &\leftarrow (1-\rho) \tau_{ij} \quad \forall (i, j) \\ \rho &= \text{factor de evaporación } 0 < \rho \leq 1 \end{aligned} \quad (11)$$

La actualización de los caminos de feromona se realiza teniendo en cuenta la calidad de la función objetivo; es decir, mientras mejor sea la función objetivo (menor F_{obj}), más feromona se debe depositar en los arcos que conforman la k -ésima alternativa.

$$\tau_{ij} \leftarrow \tau_{ij} + \frac{1}{fobj^k} \quad \forall (i, j) \in Alternativa^k \quad (12)$$

Se debe mencionar que una vez se generan las alternativas de solución, cada una es sometida al proceso de mejoramiento, tal como se describe en el algoritmo 1.

Enjambre de partículas (PSO: particle swarm optimization)

El algoritmo PSO es una técnica de optimización estocástica basada en el comportamiento de poblaciones. Fue propuesto por Eberhart y Kennedy (1995). El algoritmo asume la existencia de una población de S partículas. Estas partículas se mueven en un espacio de búsqueda N -dimensional, donde la i -ésima partícula puede ser representada por un vector $x_i = (x_{i1}, \dots, x_{iN})$. La velocidad de cada partícula se denota por $v_i = (v_{i1}, \dots, v_{iN})$. Adicionalmente, se mantiene el registro de la mejor posición individual para la partícula i [$pbest_i = (p_{i1}, \dots, p_{iN})$] y la primera mejor solución global ($gbest$) explorada hasta el momento. La velocidad de la partícula y su posición son actualizadas de acuerdo con (13) y (14):

$$v_{ij}(t+1) = w * r(v_{ij}(t)) + c_1 * r_1 * r(pbest_{ij} - x_{ij}(t)) + c_2 * r_2 * r(gbest_{ij} - x_{ij}(t)) \quad (13)$$

$$x_{ij}(t+1) = x_{ij}(t) + r(v_{ij}(t+1)) \quad (14)$$

Donde i es el índice de la partícula para $i = 1, \dots, S$; j es una dimensión específica para $j = 1, \dots, N$; w es la inercia; r_1 y r_2 son números aleatorios de distribución uniforme en el rango $(0,1)$; c_1 es un parámetro cognitivo; c_2 es un parámetro social; y finalmente, $r(\cdot)$ es una función de redondeo usada para adaptar el PSO original al problema de optimización discreta (Khushaba *et al.*, 2007; Jin *et al.*, 2008).

Considerando un problema de minimización, la mejor posición individual de cada partícula es actualizada de acuerdo con la ecuación (15), donde $f(\cdot)$ es la función objetivo o *fitness function*. Finalmente, la mejor partícula de todo el enjambre se actualiza a partir de la ecuación (16):

$$pbest_i(t+1) = \begin{cases} pbest_i(t) & \text{if } f(pbest_i(t)) \leq f(x_i(t)) \\ x_i(t) & \text{if } f(pbest_i(t)) > f(x_i(t)) \end{cases} \quad (15)$$

$$gbest(t+1) = \arg \min_{p \in pbest_i} f(p(t+1)) \quad (16)$$

Los parámetros del algoritmo son resumidos en la tabla 1, y los pasos básicos son presentados en la figura 2. El algoritmo termina su búsqueda al satisfacer al menos uno de los dos criterios de parada: 1) máximo número de iteraciones (C_{max}), o 2) valor óptimo de la función costo $f(\cdot)$ (Camacho *et al.*, 2012).

Tabla 1. Parámetros por configurar en el algoritmo PSO

PARÁMETRO	SÍMBOLO	OBSERVACIÓN
Parámetro cognitivo	c_1	Un valor alto indica partículas con alta autoconfianza
Parámetro social	c_2	Un valor alto incrementa la confianza de las partículas en las decisiones del enjambre
Peso de inercia	w	Controla la capacidad de exploración de las partículas. Valores grandes estimulan búsquedas globales. Valores pequeños estimulan la búsqueda local
Tamaño del enjambre	S	Indica el número de partículas utilizadas en la búsqueda
Límites de velocidad	$[v_{min}, v_{max}]$	Límites de velocidad
Límites de posición	$[x_{min}, x_{max}]$	Límites de posición
Máximo número de iteraciones	C_{max}	Criterio de parada del algoritmo

Fuente: elaboración propia.

Colonia artificial de abejas (ABC: artificial bee colony)

El algoritmo ABC es una técnica de optimización propuesta por Karaboga (2005). La técnica fue inspirada en el comportamiento social de las colonias de abejas durante la recolección de alimentos; por tanto, cada solución al problema de optimización es llamada *fuentes de alimento* y es representada por un vector de valores reales de N dimensiones. La colonia artificial de abejas contiene tres grupos de abejas: operarias (*employed bees*), seguidoras (*onlookers*) y *scouts*. La primera mitad de la colonia está compuesta de abejas operarias, y la segunda mitad, de abejas seguidoras. Se considera la existencia de S fuentes de alimento. El algoritmo establece la existencia de una operaria por cada fuente de alimento; por tanto, el número de operarias es igual al número de fuentes de alimento.

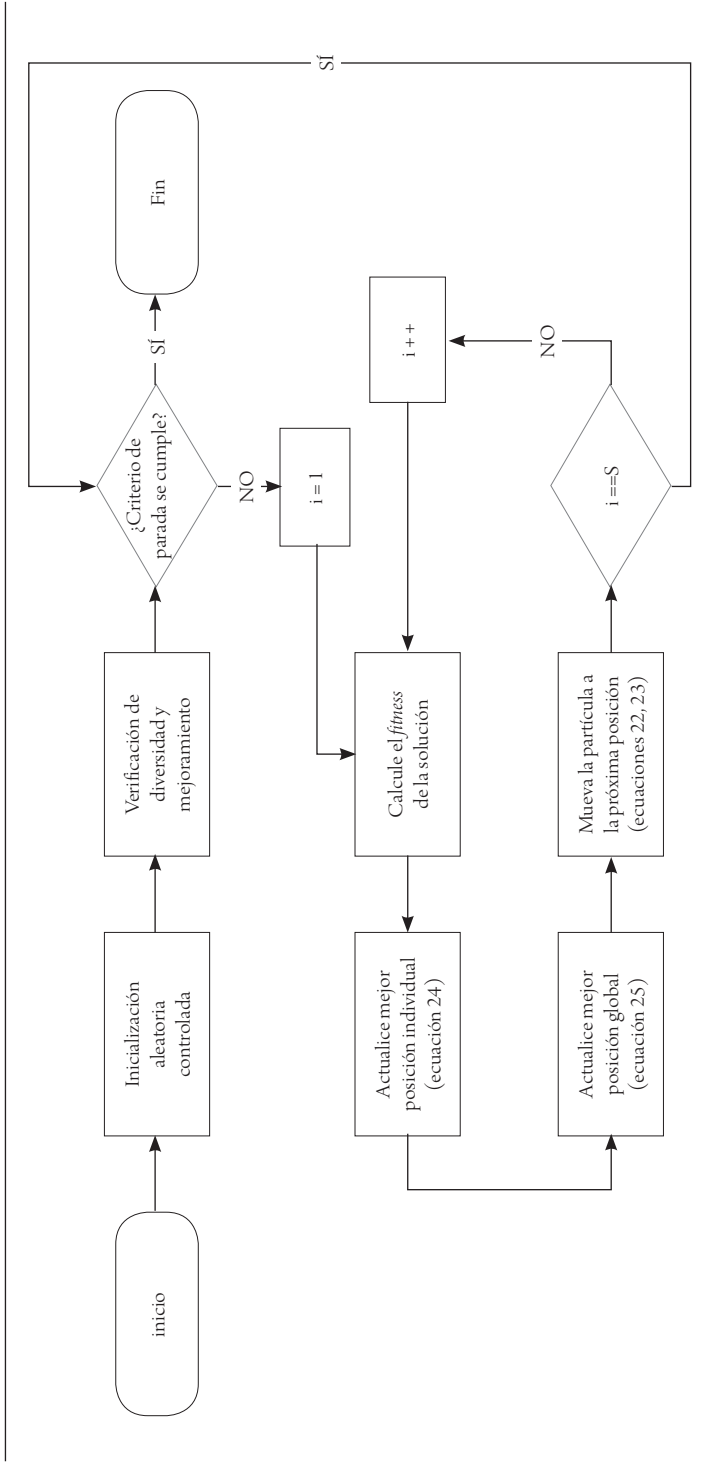


Figura 3. Flujoograma del algoritmo PSO discreto implementado

Fuente: elaboración propia.

En términos genéricos, el algoritmo puede ser dividido en dos etapas (Camacho *et al.*, 2012): 1) inicialización, y 2) búsqueda iterativa. En la primera etapa, las abejas seleccionan fuentes de alimento de forma aleatoria y determinan la cantidad de néctar. Posteriormente, estas abejas retornan a la colmena y comparten la información recolectada con las abejas seguidoras. En la segunda etapa se desarrollan tres fases iterativas: 1) se envían abejas operarias a fuentes de alimento candidatas, 2) cada abeja seguidora selecciona con cierta probabilidad una fuente de alimento por explorar (*exploitation*), 3) se determina la cantidad de operarias que se convierten en *scouts*. Estos nuevos *scouts* son enviados a buscar nuevas fuentes de alimento. A continuación se presenta una descripción más detallada de cada etapa del algoritmo.

Inicialización

El algoritmo genera una solución inicial de longitud S , donde cada solución x_p ($i = 1, 2, \dots, S$) es un vector de N dimensiones: ($x_{ij} = [x_{i1}, x_{i2}, \dots, x_{iN}]$). Las abejas operarias miden la cantidad de néctar de cada solución, retornan a la colmena y comparten la información medida con las abejas seguidoras.

Búsqueda iterativa

El conjunto de soluciones es sometido a búsquedas iterativas, con el propósito de encontrar las mejores fuentes de alimento. Durante estos ciclos, las abejas modifican el contenido de su memoria; cada abeja solo puede recordar una localización de fuente de alimento x_i . Cada ciclo es ejecutado en tres fases.

1) *Fase de las operarias*. En esta fase cada operaria calcula una nueva fuente de alimento v_i en la vecindad de la posición x_i almacenada en su memoria. En Karaboga *et al.* (2007), el cálculo de v_i se realiza mediante comparación de posiciones, incluyendo componentes aleatorios como se indica en la ecuación (17). Si la cantidad de néctar en la fuente de alimento v_i es mayor que la cantidad asociada con x_i , entonces, la abeja memoriza la nueva posición y olvida la antigua. De otra forma, la abeja conserva la posición de la fuente de alimento x_i . El procedimiento descrito es denominado *actualización condicional de la memoria de la abeja*, y se muestra en la figura 4. En el momento en que cada operaria ha terminado esta fase, se calcula un factor de probabilidad p_i utilizando la ecuación (18). Este factor indica el nivel relativo de néctar en cada fuente de alimento visitada.

2) *Fase de seguidoras*. En esta fase, las abejas seguidoras utilizan el factor p_i asociado a cada operaria para seleccionar una fuente de alimento x_i . Posteriormente, cada seguidora calcula una fuente de alimento en la vecindad del x_i seleccionado. Este cálculo es hecho con la ecuación (17). Finalmente, cada seguidora mide el néctar de la fuente de alimento v_i y decide si el néctar de v_i supera el néctar de x_i ; entonces, se actualiza la localización de la fuente de alimento en la memoria de la seguidora.

3) *Fase de las scout*. Cada vez que la exploración de una fuente de alimento x_i no termina con un incremento del néctar, entonces se actualiza el contador encargado de registrar el número de intentos de mejora de la solución. Si el valor de este contador es mayor que un umbral predefinido (C_{lim}), entonces la fuente de alimento es abandonada y se selecciona una nueva fuente de alimento de forma aleatoria. La abeja que realiza esta selección aleatoria se denomina *abeja scout*.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (17)$$

Donde $i, k \in (1, 2, \dots, S)$ y $j \in (1, 2, \dots, N)$ son índices seleccionados aleatoriamente, con $k \neq i$; ϕ_{ij} es un número aleatorio en el rango $[-1, 1]$ que controla la producción de fuentes de alimento alrededor de x_i :

$$p_i = \frac{f(x_i)}{\sum_{n=1}^S f(x_n)} \quad (18)$$

Donde $f(.)$ es el valor de la función objetivo y S es el número de fuentes de alimento. Los parámetros configurables del algoritmo son resumidos en la tabla 2, y los pasos básicos son ilustrados en las figuras 4 a 7.

Tabla 2. Parámetros por configurar en el algoritmo ABC

PARÁMETROS	SÍMBOLO	OBSERVACIÓN
Número de fuentes de alimento	S	Define el número de abejas operarias y seguidoras
Valores límite	$[v_{min}, v_{max}]$	
Número máximo de ciclos	C_{max}	Indica el número máximo de iteraciones permitidas durante la búsqueda de la solución óptima
Ciclos límite para el mejoramiento de una solución	C_{lim}	

Fuente: elaboración propia.

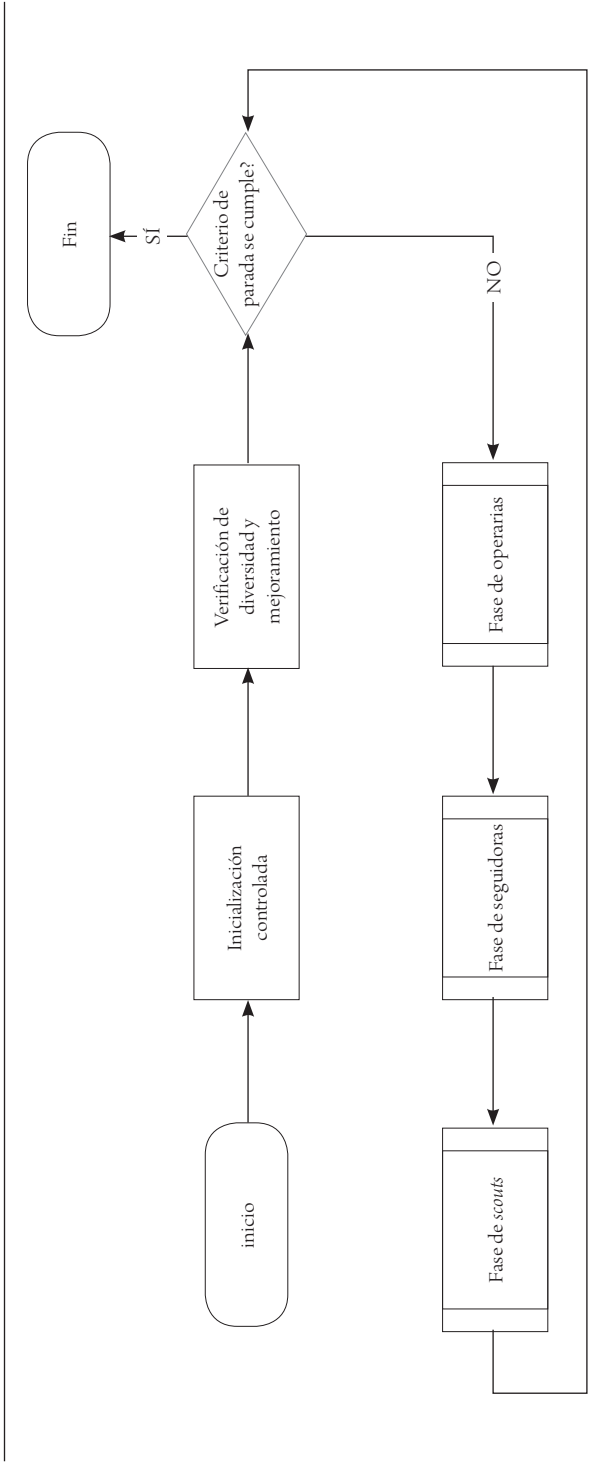


Figura 4. Flujoograma del algoritmo ABC implementado

Fuente: elaboración propia.

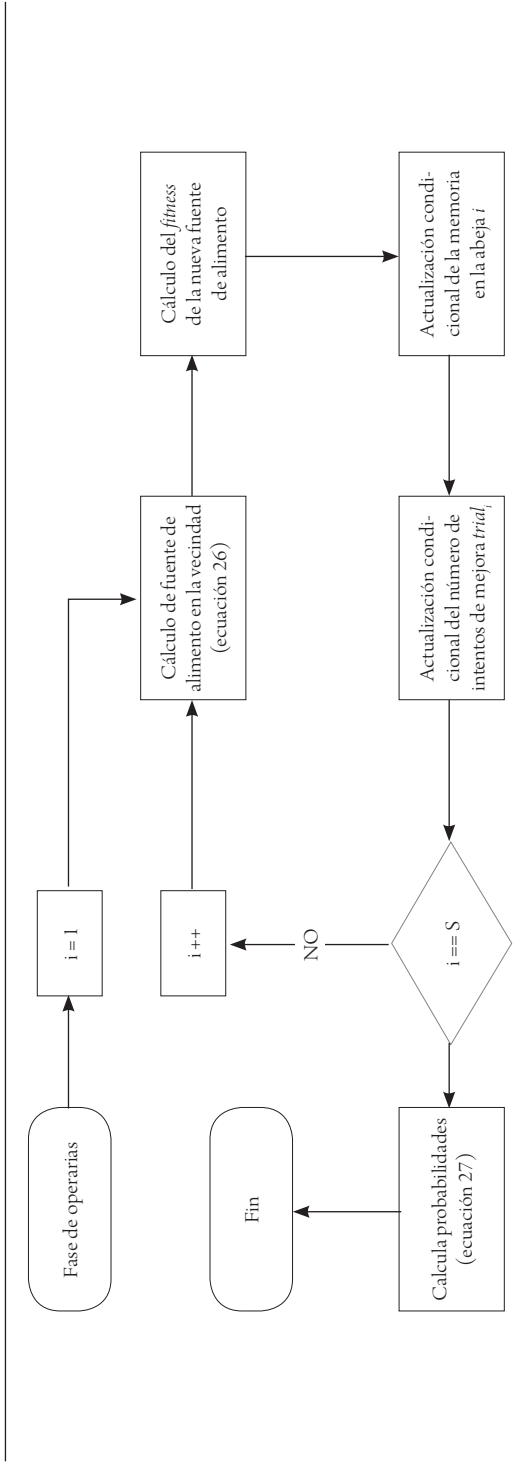


Figura 5. Flujograma de la fase *operarias* en el algoritmo ABC

Fuente: elaboración propia.

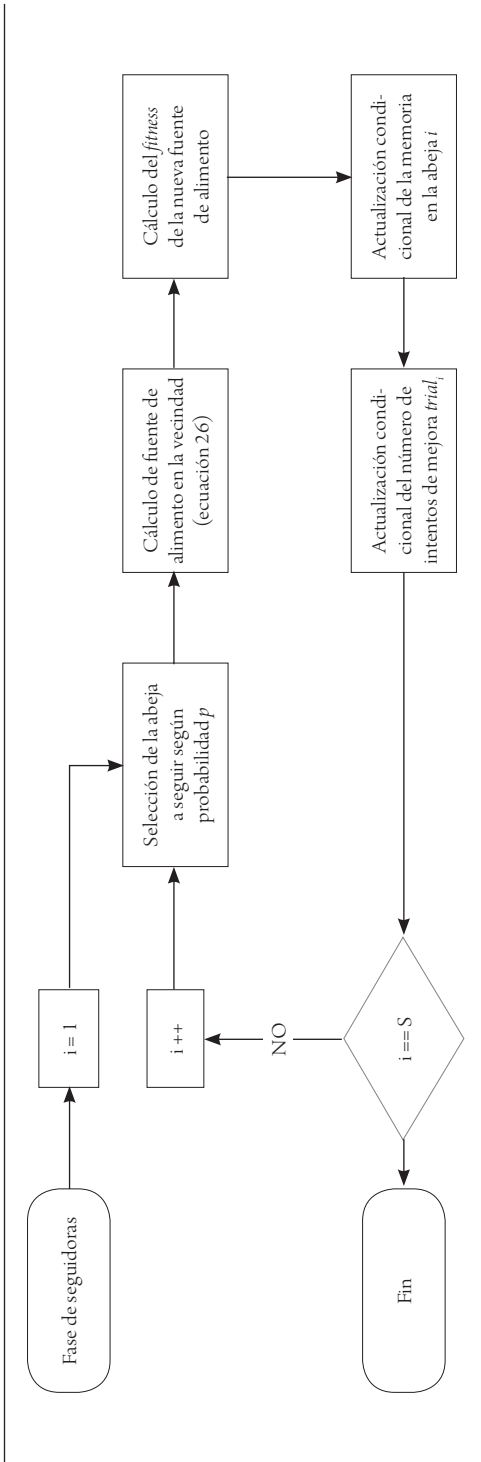


Figura 6. Flujograma de la fase seguidoras en el algoritmo ABC

Fuente: elaboración propia.

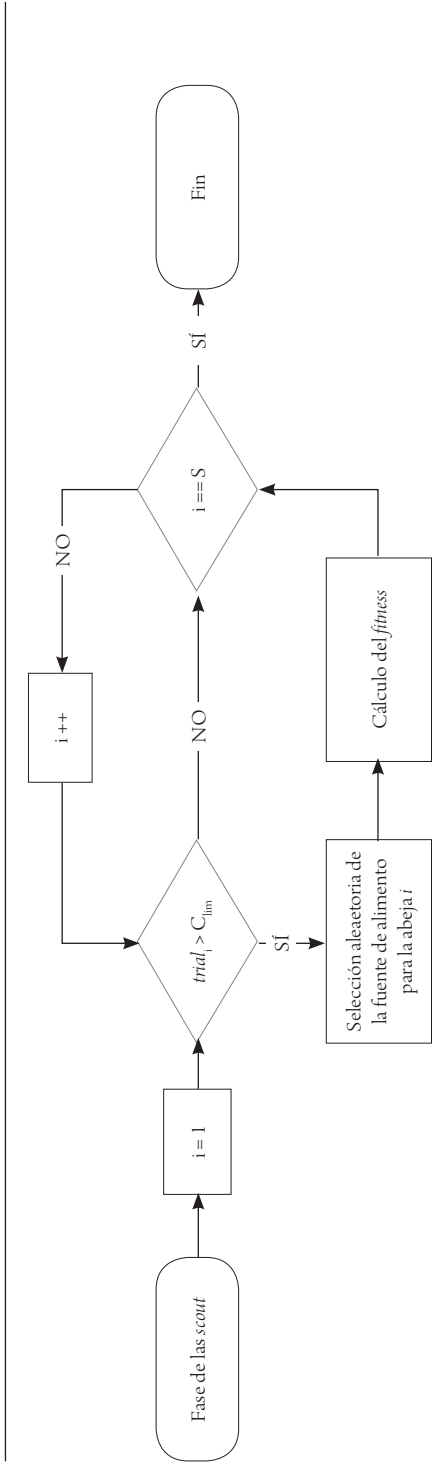


Figura 7. Flujograma de la fase *scout* en el algoritmo ABC

Fuente: elaboración propia.

Resultados

Para validar las técnicas explicadas anteriormente sobre el problema formulado en (1)-(8), se emplean dos sistemas de prueba ampliamente estudiados en la literatura especializada: el sistema de seis barras propuesto por Garver (1970) y el sistema IEEE de 24 nodos (Fang y Hill, 2003). Para lo anterior, se desarrolla un *software* en ambiente Matlab.

Para determinar el desempeño de los algoritmos, se realiza una comparación de la cantidad de los PL que cada uno debe resolver para alcanzar la mejor solución. Cada algoritmo se corre diez veces, y se muestra la información asociada con la mejor solución obtenida, los PL promedio y la respectiva desviación estándar. El criterio de parada en todos los casos se ajustó a 500 PL sin mejorar la incumbente.

Sistema de Garver

Esta red tiene seis barras, quince ramas, una demanda total de 760 MW y un máximo de cinco líneas en paralelo por cada corredor. En la etapa de inicialización, los parámetros utilizados fueron los siguientes: $\rho_{mi} = 5$ y $\rho_{div} = 3$.

La mejor solución obtenida implica una inversión de USD 200×10^3 , con las siguientes adiciones de circuitos: $n_{2-6} = 4$, $n_{3-5} = 1$, $n_{4-6} = 2$. Para los cuatro algoritmos utilizados, se encuentra la mejor respuesta en todas las corridas. La tabla 3 consigna los parámetros utilizados y la información relevante del desempeño de los algoritmos, de donde se puede observar que el CBGA y el ACO tienden a converger con mayor facilidad cuando se comparan con los algoritmos PSO y ABC. Además, se concluye que el ABC tiene que realizar un considerable mayor esfuerzo que los tres restantes algoritmos.

Tabla 3. Parámetros y desempeño de los cuatro algoritmos para el sistema Garver

	CBGA	ACO	PSO	ABC
Parámetros	30 individuos, $k = 2$, $\rho_{mut} = 3, \rho_{prom} = 1$	30 hormigas, $\alpha = \beta = 1, \rho = 0,65$	30 partículas, $w = 0,9$, $v_{max} = 2, c_1 = c_2 = 2$	50 fuentes, $C_{lim} = 5$
Mejor respuesta (USD $\times 10^3$)	200			

Continúa

	CBGA	ACO	PSO	ABC
Peor respuesta (USD $\times 10^3$)		200		
Promedio (PL)	433	348	686	3299
Desviación estándar (PL)	261	139	272	2380

Fuente: elaboración propia.

Es importante mencionar, además, que la mejor respuesta obtenida coincide con la reportada en la literatura especializada (Gallego *et al.*, 1998; Silva *et al.*, 2005).

Sistema IEEE de 24 nodos

Este sistema tiene 24 barras, 41 corredores, una demanda total de 8550, 10 barras de generación y un máximo de 3 líneas para ser adicionadas en cada corredor. En la etapa de inicialización, los parámetros utilizados fueron los siguientes: $\rho_{ini} = 10$ y $\rho_{div} = 5$. Para este sistema, la mejor solución está asociada a una inversión de USD 152×10^6 y requiere los siguientes refuerzos en la red: $n_{06-10} = 1$, $n_{07-08} = 2$, $n_{10-12} = 1$, $n_{14-16} = 1$. En la tabla 3 se puede observar la información de los parámetros utilizados y la comparación del esfuerzo computacional de cada algoritmo.

Tabla 4. Parámetros y desempeño de los cuatro algoritmos para el sistema IEEE 24 nodos

	CBGA	ACO	PSO	ABC
Parámetros	50 individuos, $k = 2$, $\rho_{mut} = 3, \rho_{prom} = 1$	50 hormigas, $\alpha = \beta = 1, \rho = 0,65$	50 partículas, $w = 0,9$, $v_{max} = 2, c_1 = c_2 = 2$	80 fuentes, $C_{lim} = 5$
Mejor respuesta (USD $\times 10^6$)	152			
Peor respuesta (USD $\times 10^6$)	152			
Promedio (PL)	1333	1062	1524	6801
Desviación es- tándar (PL)	826	559	386	4224

Fuente: elaboración propia.

En este caso se observa nuevamente un mejor comportamiento de la técnica ACO, y un desempeño comparable en el CBGA y el PSO. De nuevo, el esfuerzo computacional del ABC resulta demasiado alto cuando se compara con el resto de técnicas.

Conclusiones

Se presenta un estudio comparativo de cuatro técnicas metaheurísticas para solucionar el problema del PET, utilizando el modelo DC. Cada técnica es adaptada para resolver el problema entero mixto del planeamiento, y la comparación del desempeño de las técnicas se realiza con base en el número de problemas operativos que se deben resolver para obtener la mejor respuesta.

Se incluyen dos aspectos novedosos en la implementación de todas las metaheurísticas. Primero, la inicialización aleatoria controlada, que permite obtener un conjunto de configuraciones diversas al inicio del proceso iterativo, mejorando las capacidades de exploración de cada técnica. Además, se incluye también una etapa de mejoramiento de la optimalidad, con el fin de retirar circuitos redundantes de las configuraciones resultantes de cada iteración.

De los resultados obtenidos, se observa que en todas las ejecuciones de los algoritmos, estos encuentran las mejores soluciones reportadas en la literatura especializada. Además, el ACO presenta un comportamiento bastante interesante, al tener el mejor desempeño promedio en ambos sistemas de prueba, al compararse con las demás técnicas utilizadas. Por otro lado, el algoritmo ABC presenta el mayor número de PL para encontrar respuestas óptimas; comportamiento esperado, ya que en cada iteración del ABC se realizan entre dos y tres evaluaciones de la función objetivo.

Referencias

- Bahiense, L., Oliveira, G., Pereira, M. y Granville, A. S. (2001). A mixed integer disjunctive model for transmission network expansion. *IEEE Transactions on Power Systems*, 16(3), 560-565.
- Beasley, J. y Chu, P. C. (1997). A genetic algorithm for the generalized assignment problem. *Computers Operations Research*, 24(1), 17-23.
- Binato, S., Pereira, M. V. F. y Granville, S. (2001). A new benders decomposition approach to solve power transmission network design problems. *IEEE Transactions on Power Systems*, 13(3), 822-828.

- Camacho, G. A., LLanos, C. H. y Berger, P. A. (2012). *Avaliação dos efeitos de combinações função-custo/ algoritmo-de-busca sobre o erro de classificação dos sistemas de controle miolétrico com sintonização (IPCA)*. Brasília: Universidade de Brasília.
- Colorni, A. *et al.* (1994). Heuristics from nature for hard combinatorial optimization problems. *International Transactions in Operational Research*, 3(1), 1-21.
- Correa, C. A. (2008). *Planeamiento multiobjetivo de la expansión de la transmisión considerando múltiples escenarios de generación* (tesis de maestría). Pereira: Universidad Tecnológica de Pereira.
- Eberhart, R. y Kennedy, J. (1995). A new optimizer using particle swarm theory. En *Proceedings of the Sixth International Symposium on Micro Machine and Human Science* (pp. 39-43). Nagoya, Japón: IEEE.
- Escobar, A., Gallego, R. y Romero, R. (2004). Multistage and coordinated planning of the expansion of transmission systems. *IEEE Transactions on Power Systems*, 19(2), 735-744.
- Fang, R. y Hill, D. (2003). A new strategy for transmission expansion in competitive electricity markets. *IEEE Transactions on Power Systems*, 18(1), 374-380.
- Gallego, R., Monticelli, A. y Romero, R. (1998). Comparative studies on nonconvex optimization methods for transmission. *IEEE Transactions on Power Systems*, 13(3), 822-828.
- Garver, L. (1970). Transmission network estimation using linear programming. *IEEE Transactions on Power Apparatus and Systems*, 89(7), 1688-1697.
- Haffner, S. *et al.* (2000). Branch and bound algorithm for transmission system expansion planning using a transportation model. *IEE Proceedings on Generation, Transmission and Distribution*, 147(3), 149-156.
- Hemmati, R., Hooshmand, R. y Khodabakhshian, A. (2013). Comprehensive review of generation and transmission expansion planning. *Generation, Transmission Distribution (IET)*, 7(9), 955-964.
- Jin, J., Wang, X. y Zhang, J. (2008). Optimal selection of EEG electrodes via DPSO algorithm. *Proceedings of the 7th World Congress on Intelligent Control and Automation* (pp. 5095-5099). S. I.: IEEE.
- Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization*. Recuperado de http://mf.erciyes.edu.tr/abc/pub/tr06_2005.pdf
- Karaboga, D. y Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39, 459-471.

- Khushaba, R., Al-Ani, A. y Al-Jumaily, A. (2007). Swarm Intelligence based Dimensionality Reduction for Myoelectric Control. 3.rd *International Conference on Intelligent Sensors, Sensor Networks and Information* (pp. 577-582). Melbourne: IEEE.
- Kim, K., Park, Y. y Lee, K. (1988). Optimal long term transmission expansion planning based on maximum principle. *IEEE Transactions on Power Systems*, 3(4), 1494-1501.
- Rider, M., García, A. y Romero, R. (2007). Power system transmission network expansion planning using ac model. *IET Generation, Transmission and Distribution*, 1(5), 731-742.
- Rocha, M. C. y Saraiva, J. T. (2013). A discrete evolutionary fPSOg based approach to the multiyear transmission expansion planning problem considering demand uncertainties. *International Journal of Electrical Power and Energy systems*, 45(1), 427-442.
- Romero, R., Asada, E., Carreno, E. y Rocha, A. C. (2007). Constructive heuristic algorithm in branch-and-bound structure applied to transmission network expansion planning. *Generation, Transmission, Distribution (IET)*, 1(2), 318-323.
- Romero, R. y Monticelli, A. (1994). A zero-one implicit enumeration method for optimizing investments in transmission expansion planning. *IEEE Transactions on Power Systems*, 9(3), 1385-1391.
- Romero, R., Monticelli, A., García, A. y Haffner, S. (2002). Test systems and mathematical models for transmission network expansion planning. *IEE Proceedings on Generation, Transmission and Distribution*, 149(1), 27-36.
- Sánchez, I., Romero, R., Mantovani, J. y Rider, M. (2005). Transmission-expansion planning using the DC model and nonlinear programming technique. *IEE Proceedings on Generation, Transmission and Distribution*, 152(6), 763-769.
- Silva, I. et al. (2005). Transmission network expansion planning with security constraints. *IEE Proceedings on Generation, Transmission and Distribution*, 152(6), 828-836.
- Villasana, R., Garver, L. y Salon, S. (1985). Transmission network planning using linear programming. *IEEE Transactions on Power Apparatus and Systems*, 104(2), p. 349-356.
- Zhang, H., Vittal, V., Heydt, G. y Quintero, J. (2012). A mixed-integer linear programming Approach for multi-stage security-constrained transmission expansion planning. *IEEE Transactions on Power Systems*, 27(2), 1125-1133.

