

ALGORITMO GENÉTICO ESPECIALIZADO PARA LA LOCALIZACIÓN ÓPTIMA DE CAPACITORES EN SISTEMAS DE DISTRIBUCIÓN DE ENERGÍA ELÉCTRICA

RESUMEN

Algoritmo para la localización óptima de capacitores en sistemas de distribución, es un problema de programación no lineal, entera mixta (PNLIM). Entrega los puntos de localización, tipos y tamaños de estos, y el esquema de control para diferentes niveles de carga.

PALABRAS CLAVES: Algoritmos genéticos, optimización de sistemas.

ABSTRACT

A Genetic Algorithm for solving the optimal capacitor placement problem in radial distribution is presented. Is a non linear integer mixed problem (NLIMP). The algorithm obtains the placement points, types and sizes of the capacitor banks and the control scheme.

KEYWORDS: Genetic algorithms, optimization systems.

Ramón A. Gallego
Profesor Titular U.T.P.
ralfonso@utp.edu.co

Antonio H. Escobar Z.
Profesor Asociado U.T.P.
aescobar@utp.edu.co

Darío E. Rodas Rendón
Profesor titular U.T.P.
drodas@utp.edu.co

1. INTRODUCCION

En distribución se instalan bancos de capacitores para compensar la potencia reactiva, reducir las pérdidas de energía, mejorar el perfil de tensión, y en menor medida, aliviar la capacidad del sistema. Surge el problema de localización de bancos de capacitores para reducir los costos de operación disminuyendo pérdidas, la **localización** óptima de bancos de capacitores en sistemas de distribución radial, debe determinar: **donde** ser instalados, **tipos, tamaños** y también un **esquema de control** de esos bancos para el caso de operación en diferentes niveles de carga y cuando se considera bancos de capacitores variables. La función objetivo consiste en minimizar los costos de operación para un horizonte de funcionamiento especificado y las restricciones son las ecuaciones de flujo de carga y las restricciones de operación tales como los módulos de tensión de barra.

Este problema es interesante porque su modelo matemático es un PNLIM con función objetivo no diferenciable porque los costos de los bancos de capacitores varían de forma discreta. La carga de un sistema varía continuamente durante el día, entretanto esta variación es discretizada en un número determinado de niveles de carga para que sea posible resolverlo usando técnicas de optimización existentes. Por tanto, a cada nivel de carga, se debe determinar un esquema de control de los bancos de capacitores variables. Así mismo el tamaño y el punto de operación de los bancos de capacitores variables son variables discretas. Estos problemas pueden ser analizados adecuadamente usando algoritmos tales como Simulated Annealing, Algoritmos Genéticos y Tabu Search, como son presentados en [1,2,7]. La localización de bancos de capacitores ha sido

estudiado desde la década de los 60, tanto en su modelamiento matemático [5,6], así como el desarrollo de técnicas de solución. Así, fueron propuestas técnicas de solución usando optimización clásica como descomposición de Benders realizando un modelamiento matemático [5]. En los últimos años fueron propuestos algoritmos usados en problemas combinatorios y complejos, tales como Simulated Annealing, Algoritmos Genéticos [1] y Tabu Search [2,7]. En este trabajo se presenta un algoritmo genético con estrategias y operadores genéticos diferentes al algoritmo presentado en [1] y el desempeño del algoritmo propuesto es comparado con los obtenidos en [1,2,7].

2. FORMULACIÓN DEL PROBLEMA

La localización de capacitores en sistemas de distribución radial tiene como objetivo minimizar los costos relacionados con las pérdidas de energía y de inversión en la instalación de los bancos para un horizonte de operación de T años, manteniendo los módulos de tensión de las barras dentro de límites especificados. Así, el modelo matemático asume la siguiente forma:

$$\min v = k_e \sum_{i=0}^{nt} T p_i(x^i) + \sum_{k=1}^{nc} f(u_k^0) \quad (1)$$

s.a

$$G^i(x^i, u^i) = 0; \quad i = 0, 1, \dots, nt$$

$$H^i(x^i) \neq 0; \quad i = 0, 1, \dots, nt$$

$$0 \# u_k^i \# u_k^0; \quad k \in C_1 \quad \text{ó}$$

$$0 \# u_k^i = u_k^0; \quad k \in C_2$$

que es un PNLIM con una función objetivo no diferenciable y en la que nt representa un número de niveles de carga en que es discretizada una curva de duración de carga como se muestra en la figura 1, donde

también se asume que la carga varía de manera uniforme. Así, una carga Q_i es representada en la siguiente forma:

$$Q_i(\hat{\delta}) = Q_i^0(\hat{\delta}) S_i(\hat{\delta}) \quad (2)$$

en la que Q_i^0 representa el valor pico de carga y S_i es un factor de discretización, n_c es el número de barras candidatas a adición de bancos, $G^i(x^i, u^i) = 0$ es el conjunto de ecuaciones de flujo de carga para un nivel de carga i , x^i son las variables de estado y las u^i son las variables de control, esto es, la potencia reactiva de los bancos de capacitores que deben ser instalados; $H^i(x^i) \neq 0$ representa una restricción de operación para el nivel de carga i , u_k^0 es el tamaño de los bancos que deben ser instalados en la barra k , u_k^i es un punto de operación del banco instalado en la barra k y operando al nivel de carga i .

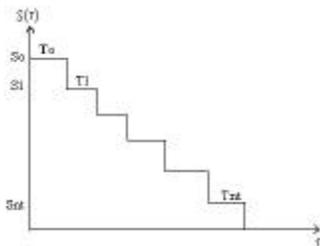


Figura 1. Curva de Duración de Carga

En (1) aparece una función objetivo con 2 partes: 1) la primera representa los costos debido a las pérdidas de energía en la que T_i es una fracción de T durante el cual el sistema opera al nivel de carga i con pérdidas $p_i(x^i)$, siendo k_c un parámetro que representa costos de energía (\$ US/KWh) y, 2) la segunda parte representa los costos de instalación de los bancos siendo $f(u_k^0)$ una función discreta no diferenciable.

El sistema (1) es difícil de resolver porque la función objetivo no es diferenciable y presenta restricciones no lineales, existen variables discretas y reales y adicionalmente, presenta un fenómeno de explosión combinatorial. El problema (1) se resuelve para dos tipos de operación: (1) usando bancos fijos que operan de forma permanente en el sistema y (2) usando bancos variables, operando con topología diferente para cada nivel de carga. Este último tipo de operación es mucho más complicado para resolver. Para la localización de bancos fijos, los capacitores son tratados como fuentes de potencia reactiva constantes, siendo válida la relación:

$$u^0 = u^1 = \dots = u^{n_t} \quad (3)$$

Para el caso de bancos variables, existe una topología para cada nivel de carga, por tanto para cada banco localizado en la barra k existen $n_t + 1$ formas de operación, u_k^i ; $i = 0, 1, k, \dots, n_t$ que deben ser encontradas,

una para cada nivel de carga. Los tamaños de los bancos y las capacidades de operación para cada nivel de carga deben satisfacer la siguiente relación:

$$0 \neq u_k^i \neq u_k^0 \quad (4)$$

Se puede resolver adecuadamente el sistema (1) usando un algoritmo genético. Una forma simple consiste en identificar a cada elemento de la población, una topología o configuración, a través de una codificación de variables discretas o enteras. Las variables reales son encontradas resolviendo un problema de flujo de carga radial. Así, por ejemplo, una configuración para el problema (1) con bancos fijos puede ser un vector de tamaño n_c , $\hat{U} = \{u_1, u_2, k, \dots, u_{n_c}\}$, en que u_i es una capacidad de generación reactiva de capacitor permanente instalado en la barra i . En el caso de bancos variables, una configuración puede ser representada por un vector $\zeta = \{u_1, u_2, k, \dots, u_{n_c}\}$ en la que $u_i = \{u_i^1, u_i^2, k, \dots, u_i^{n_t}\}$ es un vector de dimensión n_t y u_i^j es un punto de operación del banco instalado en la barra i y operando al nivel de carga j . En la propuesta de codificación presentada, las variables u son discretas porque representan una potencia reactiva generada por los bancos de capacitores. Se puede sustituir esas variables discretas por las variables n que representan un número de bancos que son colocados en cada barra. En este caso, los elementos del vector de codificación son números enteros. Así, una configuración es un espacio de configuraciones posibles del problema estando adecuadamente representadas y, una vez escogida una configuración, es posible encontrar las otras variables del problema (1) a través de los siguientes pasos: (1) resolver $n_t + 1$ problemas de flujo de carga radial, verificar si existen violaciones de los límites de tensión para algún de los $n_t + 1$ problemas de flujo de carga y encontrar las pérdidas para cada nivel de carga, (2) si en el paso anterior no existe violación de los límites de tensión en la totalidad de los $n_t + 1$ problemas de flujo de carga, entonces la configuración analizada es factible, y (3) si la configuración es factible, se puede encontrar un valor de la función objetivo de forma trivial.

Los algoritmos genéticos, encuentran un conjunto de configuraciones iniciales, factibles o no factibles, con una lógica de cada algoritmo hasta satisfacer un criterio de parada. Un gran esfuerzo computacional de esos algoritmos para el caso del problema de localización de bancos de capacitores se requiere en la solución de problemas de flujo de carga radial. En este algoritmo puede ser necesario resolver millares de problemas de flujo de carga, es deseable disponer de un algoritmo de flujo de carga que sea rápido y confiable (robusto).

3. EL ALGORITMO GENÉTICO

Se presenta las características del algoritmo genético usado en el problema de la localización de capacitores fijos y variables. Se discuten las características básicas del algoritmo propuesto y sus diferencias a otras propuestas que usan algoritmos combinatorios (Simulated Annealing, Algoritmo Genético, Tabu Search) para este tipo de problema. También serán presentadas propuestas diferentes en otras partes importantes del algoritmo, tales como la técnica de reducción de barras candidatas, en la determinación de la configuración inicial y en el uso de los llamados indicadores de sensibilidad. Posteriormente, es presentada una propuesta de algoritmo genético con sus principales características.

El algoritmo genético inicia el proceso a partir de un conjunto de configuraciones (población inicial) que puede ser obtenida aleatoriamente o usando algoritmos heurísticos constructivos simples y rápidos. En cada iteración es obtenido un nuevo conjunto de configuraciones (nueva población) a partir de la población corriente usando los operadores de selección, recombinación y mutación. En cada nueva iteración son encontradas configuraciones de mejor calidad y, eventualmente, en este proceso iterativo se puede encontrar una solución (configuración) óptima global. En otras palabras, un algoritmo genético realiza una búsqueda usando un conjunto de soluciones (configuraciones) y a través de un proceso iterativo son encontradas nuevas configuraciones candidatas. Lógicamente, un número de configuraciones visitadas en este proceso de búsqueda debe ser un número muy reducido de configuraciones del espacio de configuraciones y debe existir una estrategia adecuada para visitar las configuraciones más atractivas. Se presenta en forma separada, las principales características del algoritmo genético propuesto y en cada caso se realiza un análisis comparativo con otras investigaciones que usan algoritmos combinatorios para el problema de localización de capacitores.

3.1 Representación y Codificación del Problema.

Para una propuesta de solución o configuración, el problema (1) se reduce a verificar la factibilidad y calidad de la solución encontrada. La factibilidad es verificada a través de la solución de un problema de flujo de carga radial que adicionalmente calcula las pérdidas del sistema eléctrico. Típicamente el sistema eléctrico es discretizado en 3 niveles de operación, por tanto, analizar una propuesta de solución consiste en resolver 3 problemas de flujo de carga radial. Una función objetivo puede ser fácilmente encontrada conociendo las pérdidas en los 3 niveles de carga obtenidas de la solución del flujo de carga radial, así como la propuesta de solución que especifica el número de barras con adición de bancos de capacitores y el número de bancos que debe operar en cada barra para cada nivel de carga. Sin embargo, aún

debe ser especificada una codificación usada para representar una propuesta de solución o configuración.

Una propuesta para el problema de localización de capacitores debe especificar las barras escogidas para la localización de capacitores, el número que deben ser colocados en cada barra seleccionada y el número de ellos que deben estar operando en cada nivel de carga. La literatura especializada analiza solamente 3 niveles de carga. Así, sea n_n el número de barras candidatas para localización de capacitores, entonces una configuración es representada por un vector de tamaño $3 \times n_n$. Un elemento de una configuración indica el número de bancos de capacitores que deben estar operando en una barra a un determinado nivel de carga.

La codificación más adecuada para representar una configuración del problema es una codificación decimal o entera, esto es, respetando la naturaleza de variable entera que representa los bancos de capacitores. Por tanto, cada elemento de una configuración es un número entero representando un número de bancos de capacitores que está operando en una barra para un determinado nivel de carga.

Con la codificación decimal propuesta resulta simple determinar una función objetivo de cada configuración, pero se debe modificar la forma de trabajo de los operadores de recombinación y mutación. Así, en este trabajo es descartado el uso de la codificación binaria, muy usada en otras aplicaciones de algoritmos genéticos.

32 Funciones del Algoritmo Genético

3.2.1 Población

Existen 2 aspectos que deben ser especificados. El tamaño de la población n_p y la forma en que es determinada la población inicial. El tamaño debe ser escogida tomando en cuenta el tamaño del problema y otros parámetros genéticos, como la tasa de recombinación y tasa de mutación. En el caso del problema de localización de capacitores, el tamaño del problema está relacionado con el número de barras del sistema o el número de barras candidatas a localización de capacitores.

La población inicial puede ser encontrada de varias formas. Totalmente aleatoria, aleatoriamente controlada o usando algoritmos heurísticos constructivos rápidos. Aleatoriamente significa escoger para cada barra candidata un número de bancos de capacitores entre cero y un número máximo de bancos permitidos. Una estrategia de este tipo colocaría muchos bancos de capacitores en muchas barras candidatas. Típicamente, en problemas de localización de capacitores, las mejores soluciones son obtenidas con localización de capacitores en un número reducido de barras. Por tanto una estrategia aleatoriamente controlada es potencialmente más atractiva. En esta estrategia, para cada configuración, se

debe escoger primero un número reducido de barras y solamente en esas barras son localizados un número de bancos de capacitores, entre los límites permitidos.

Otra propuesta: encontrar una población inicial usando un algoritmo heurístico constructivo rápido. Las investigaciones sobre la utilización de algoritmos combinatoriales aplicados a problemas de ingeniería generan una población inicial con estrategias diferentes de esquema aleatorio [6,15]. Entretanto, en la presente investigación la generación de la población de manera aleatoriamente controlada presenta excelentes resultados.

3.2.2 Selección

Consiste en determinar el número de descendientes que debe generar cada una de las configuraciones candidatas de población actual. Una selección es realizada tomando en cuenta el valor de la función objetivo, o su equivalente, de las configuraciones de población actual. Sin embargo la selección puede ser realizada de varias formas diferentes y constituye uno de los dos operadores o funciones más importantes de algoritmo genético [9]. Algunas de las propuestas más importantes para implementar la selección, presentadas en la bibliografía especializada son: (1)selección proporcional, (2) selección usando escalonamiento, (3) selección basada en ordenamiento y (4) selección basada en juego. En este trabajo fue escogida una selección basada en juego debido a su facilidad de implementación y al excelente desempeño presentado en otras investigaciones.

En selección basada en el juego, el módulo de la función objetivo de las configuraciones no son cruciales, siendo importante apenas los valores relativos. La selección es realizada en n_p juegos para una población de tamaño n_p . En cada juego son escogidas aleatoriamente n_k configuraciones que deben participar siendo la ganadora la configuración que posea una función objetivo de mejor calidad. Así después de n_p juegos el proceso de selección es terminado. Lógicamente, las mejores configuraciones deben recibir un mayor número de descendientes y deben ser ganadores de los juegos en que fueron escogidos aleatoriamente para participar. Esta estrategia elimina las desventajas de la selección proporcional como es la aparición de superconfiguraciones (al inicio del proceso) o configuraciones uniformes (final del proceso), y también el problema debido a un número de descendientes no entero. La selección basada en juego encuentra un número entero de descendientes para cada configuración de la población actual.

3.2.3 Recombinación

Terminada la selección se debe proceder a la recombinación o cruce. Una recombinación consiste en escoger dos de las configuraciones con derecho a generar descendientes y proceder a cruzar esas configuraciones para producir dos nuevas configuraciones candidatas.

Una recombinación puede ser realizada de varias formas diferentes: (1) recombinación simple o de un punto, (2) recombinación en dos puntos, (3) recombinación multipunto y (4) recombinación uniforme

La recombinación usada en esta investigación fué de punto simple. Sin embargo, fue implementada una forma especializada de recombinación. El vector que representa una configuración fue separado en tres partes, una para cada nivel de carga. Así, en la recombinación, las partes corresponden a cada barra y una configuración debe ser revisada totalmente para un único descendiente como muestra la figura 2. En la figura son mostradas dos configuraciones de un sistema de cinco barras y tres niveles de operación que deben ser recombinadas. Así, se debe generar un número aleatorio entre 1 y 4 para encontrar el punto de recombinación. En la figura fue escogido el número 2 y, por tanto, los bancos localizados en las barras 1 y 2 para los 3 niveles de carga deben ser analizados de nuevo para un único descendiente y lo mismo acontece con las barras 3,4 y 5 como se muestra en la figura 2. Esta implementación se debe al hecho de que existe una relación directa entre el número de bancos en cada barra para cada nivel de carga.

El número de bancos disminuye para cada nivel de carga inferior. Si la propuesta presentada no lo toma en cuenta pueden aparecer muchas configuraciones no factibles.

También, para implementar la recombinación, se debe escoger una tasa de recombinación, \tilde{n}_c

*: Punto de Recombinación.

Nivel 1 de Carga					Nivel 2 de carga					Nivel 3 de Carga				
Barra					Barra					Barra				
1	2	3*	4	5	1	2	3*	4	5	1	2	3*	4	5
4	3	2	2	0	4	3	1	1	0	3	3	1	0	0
3	1	1	1	4	3	1	0	1	2	2	1	0	1	2

Después de la recombinación

4	3	1	1	4	4	3	0	1	2	3	3	0	1	2
3	1	2	2	0	3	1	1	1	0	2	1	1	0	0

Figura 2. Una Recombinación Genética.

3.2.4 Mutación

Una vez implementada la recombinación, se debe realizar una mutación en las configuraciones candidatas para constituir una nueva población. Una mutación en codificación binaria, consiste en escoger un elemento de la configuración y cambiar el valor de 0 a 1 o viceversa. Para otros tipos de codificación generalmente es posible definir otras alternativas de mutación. Así, por ejemplo,

en codificación decimal, se puede definir una mutación simple o múltiple. Por ejemplo, si una configuración está en forma de codificación decimal y fue escogida para mutación con un valor actual de 4, entonces una mutación simple produce una mutación del valor 4 hacia el 3 o el 5. De manera similar, se puede definir una mutación doble o triple que debe producir una variación mayor del valor actual 4 para 6 o 7 en caso de incremento o hacia 2 o 1 para el caso de decremento.

En esta investigación es usada una codificación decimal y una mutación simple presenta excelentes resultados. Asociado con la mutación, se debe escoger una tasa de mutación \tilde{n}_m que determina el número de elementos de una configuración que debe ser sometida a mutación. La literatura especializada recomienda valores de \tilde{n}_m en el intervalo de [0.001 - 0.050] mutaciones por bit.

Para la codificación propuesta, una mutación debe ser realizada tomando en cuenta problemas de no factibilidad de las configuraciones que puedan ser encontradas. Por ejemplo, suponga que el máximo número de bancos que pueden ser localizados en una barra es igual a 4. Usando la primera configuración generada después de la recombinación de la figura 2 puede acontecer casos especiales como los siguientes: (1) si la posición 11 del vector con $n_1^3 = 3$ (3 bancos localizados en la barra 1 con nivel de carga 3) fue escogida para mutación y una decisión aleatoria para reducir un banco entonces tenemos que $n_1^3 = 2$ y esta mutación sería considerada normal, (2) si una posición número 2 del vector con $n_2^1 = 3$ (3 bancos en la barra 2 con nivel de carga 1) fuera escogida para mutación y la decisión es reducir un banco entonces $n_2^1 = 2$ y, simultáneamente, se debe hacer $n_2^2 = 2$ y $n_2^3 = 2$ en las posiciones 7 y 12 porque los niveles de carga inferiores no pueden tener un número de bancos mayor que en el nivel de carga más elevado, (3) si la posición número 1 fuera escogida entonces la única opción es reducir el número de bancos porque se encuentra en el límite superior, así tenemos que $n_1^1 = 3 \Rightarrow n_1^2 = 3$, (4) de manera parecida, si fuera escogida una posición 8 del vector la única opción es aumentar bancos para $n_3^2 = 1$. Existen además otras opciones que pueden ser consideradas. El aspecto crucial de esta propuesta de mutación es que se debe evitar, en la medida de lo posible, una generación de configuraciones candidatas no factibles.

También, se debe escoger un criterio de parada. En el problema de localización de capacitores, una parte del algoritmo que precisa de mayor esfuerzo computacional es determinado por la función objetivo que exige una solución de 3 problemas de flujo de carga radial para evaluar una función objetivo de una configuración de la población. Así, el número máximo de problemas de flujo de carga resuelto debe hacer parte del criterio de parada.

En el algoritmo no fueron implementadas otras estrategias muy usadas por otros investigadores para resolver el problema de localización óptima de bancos de capacitores en sistemas de distribución radial. Esas estrategias son las siguientes: (1) generación de población inicial de forma aleatoria o usando un algoritmo heurístico constructivo, (2) uso de indicadores de sensibilidad y (3) una solución del Atamaño del problema @. Generación aleatoria de población inicial es usada en [3] pero esa propuesta genera configuraciones de pobre calidad y, como consecuencia, el esfuerzo computacional debe ser mayor para encontrar soluciones de buena calidad. De otro lado, el uso de algoritmos heurísticos constructivos es una idea muy interesante, sin embargo será necesario un mayor esfuerzo computacional para generar esas configuraciones de calidad. En [10] fue usada esa idea, pero para un algoritmo Tabu Search que aprovecha mejor ese tipo de información lo que no ocurre con el algoritmo genético. Los indicadores de sensibilidad, usando relaciones matemáticas aproximadas para identificar barras atractivas para la localización de bancos, fueron usadas en [2,3] para reducir el número de barras candidatas a localización de bancos de capacitores. Esta propuesta pretende disminuir la complejidad del problema disminuyendo el número de variables y, por tanto, el espacio de configuraciones del problema, mas cuando se calcula usando únicamente una configuración base y un único nivel de carga, esta propuesta puede llevar a gruesos errores eliminando del conjunto de barras candidatas, barras que hacen parte de la topología óptima, como acontece en [2]. El Asizing problem@ consiste en determinar una topología óptima de localización de bancos para los niveles de carga intermedios cuando es conocido el número de bancos para un nivel de carga elevado. Esta dificultad aparece cuando se resuelve un problema usando métodos exactos de optimización y también cuando se usa, por ejemplo, un algoritmo Tabu Search como ocurre en [10]. En el algoritmo genético este problema se resuelve de una forma trivial. Por otro lado, un algoritmo de Simulated Annealing, como se presenta en [1], generalmente precisa de un esfuerzo computacional mayor para encontrar soluciones de calidad equivalente a las encontradas usando algoritmos combinatorios.

4 ALGORITMO GENETICO PROPUESTO

El algoritmo genético desarrollado puede ser resumido en los siguientes pasos:

4.1. Determinación de la Población Inicial

Generar una población inicial de tamaño n_p . Cada configuración es generada de forma aleatoria controlada, esto es, se escoge un número reducido de barras, n_k , donde son localizados bancos, típicamente alrededor del 10% de las barras. Para cada barra seleccionada es escogida, también de forma aleatoria, un número de

bancos entre $[1, n_{\max}]$ siendo n_{\max} el número máximo de bancos que puede ser localizado en cada barra. Esta propuesta toma en cuenta el hecho que en las topologías óptimas apenas en un número reducido de barras son localizadas bancos de capacitores.

4.2. Cálculo de la Función Objetivo.

Para la población actual, se encuentra una función objetivo de cada una de las configuraciones de población. En este paso, para cada configuración de población, se debe resolver 3 problemas de flujo de carga radial. Se debe actualizar la probable solución.

4.3. Selección

Usando una **selección** basada en juego, implementarla, esto es, encontrar un número de descendientes de cada configuración. En este proceso las configuraciones no factibles (violación de límites de operación) son descartadas. Es implementada esta estrategia porque raramente aparecen configuraciones no factibles. Esto acontece porque el problema es de operación, esto es, una topología inicial sin bancos ya es factible y la localización de bancos, casi siempre mejora esas condiciones de operación.

4.4. Cruzamiento

Implementar un cruzamiento simple tomando en cuenta la estrategia propuesta anteriormente (ver Figura 2), esto es, los bancos localizados en una barra para todos los niveles de carga deben ser revisados para un único descendiente.

4.5. Mutación

Implementar una mutación simple, esto es, se escoge aleatoriamente una posición para mutación y, también aleatoriamente, se decide aumentar o disminuir el número de bancos en una unidad. Adicionalmente, se debe tomar en cuenta los límites del número de bancos que pueden ser adicionados en una barra y que el número de bancos instalados para un nivel de carga intermedio no puede ser superior a los bancos instalados en el nivel de carga más elevado, como fue anteriormente analizado.

4.6. Criterio de Parada

Parar el proceso cuando uno de los dos siguientes criterios se satisface: (1) si el número de problemas de flujo de carga sobrepasa una cifra previamente especificada o (2) si una probable solución no mejora en un número especificado de generaciones.

5. PRUEBAS EXPERIMENTALES

Por razones de espacio no se adjuntan resultados pero la metodología se probó en un sistema de 9 barras [7], en otro sistema de 69 barras [5], otro mas de 33 barras [7]

6. CONCLUSIONES

El algoritmo genético presentado encuentra fácilmente las mejores soluciones para los sistemas probados. Particularmente el algoritmo presentado encuentra soluciones de mejor calidad que el algoritmo genético presentado en [2]. En relación al algoritmo Tabu Search presentado en [10] el algoritmo genético encuentra prácticamente las mismas soluciones con la ventaja de un menor esfuerzo computacional y una mayor diversidad de soluciones sub - óptimas. Es sorprendente verificar la gran cantidad de soluciones sub - óptimas que presentan todos los sistemas probados. El uso de indicadores de sensibilidad para reducir el número de barras candidatas con la finalidad de reducir la complejidad del problema puede frecuentemente eliminar soluciones óptimas o sub - óptimas en el problema de localización de capacitores en sistemas de distribución radial. Esta estrategia, típica en investigaciones de este tipo de problema fue evitado en el presente trabajo

7. BIBLIOGRAFIA

- [1] S. Sundhararajan, A. Pahwa: AOptimal Selection of Capacitors for Radial Distribution Systems Using a Genetic Algorithm@, IEEE Transactions on Power Systems, Vol. 9, N° 3, pp 1499-1505, August 1994.
- [2] Y.C. Huang, H.T. Yang, C.L. Huang: ASolving the Capacitor Placement Problem in a Radial Distribution System Using Tabu Search Approach@ IEEE Trans. on Power Systems, Vol. 11, N° 4, pp 1868-1873, Nov. 1996.
- [3] C.R. Reeves: AModern Heuristic Techniques for Combinatorial Problems@, McGraw-Hill Company, 1995.
- [4] M.E. Baran, F.F. Wu: ANetwork Reconfiguration in Distribution Systems for Loss Reduction and Load Balancing@, IEEE Transactions on Power Delivery, Vol. 4, N° 2, pp 1401-1407, April/89.
- [5] M.E. Baran, F.F. Wu: AOptimal Capacitor Placement on Radial Distribution Systems@, IEEE Transactions on Power Delivery, Vol. 4, N° 1, pp 725-733, January 1989.
- [6] J.J. Grainger.: ACapacite Release by Shunt Capacitor Placement on Distribution Feeders: A Voltage Dependent Model@, IEEE Transactions on Power Apparatus and Systems, Vol.PAS-101, N° 15, pp 1236-1244, May 1982.
- [7] R. Romero, R. Gallego: AAlgoritmo de Búsqueda Tabú en la Localización de Bancos de Condensadores en Sistemas de Distribución Radial@, XIII Congreso Chileno de Ingeniería Eléctrica, Noviembre 1999.