

2011-06-01

Solución del flujo óptimo de potencia vía algoritmos genéticos

Édgar Rubén Muela Velásco

Universidad de La Salle, Bogotá, emuela@unisalle.edu.co

Clauco Andrés Latorre Balaguera

Ingetec S.A., caribbean_cool18@hotmail.com

Diego Armando León Valencia

Ingetec S.A., diegoleon@ingetec.com.co

Follow this and additional works at: <https://ciencia.lasalle.edu.co/ep>

Citación recomendada

Muela Velásco, Édgar Rubén; Latorre Balaguera, Clauco Andrés; and León Valencia, Diego Armando (2011) "Solución del flujo óptimo de potencia vía algoritmos genéticos," *Épsilon*: Iss. 16 , Article 4. Disponible en:

This Artículos de investigación is brought to you for free and open access by the Revistas descontinuas at Ciencia Unisalle. It has been accepted for inclusion in Épsilon by an authorized editor of Ciencia Unisalle. For more information, please contact ciencia@lasalle.edu.co.

Solución del flujo óptimo de potencia vía algoritmos genéticos

Optimal Power Flow Solution through Genetic Algorithms

ÉDGAR RUBÉN MUELA VELASCO*

CLAUCO ANDRÉS LATORRE BALAGUERA**

DIEGO ARMANDO LEÓN VALENCIA***

RESUMEN

Los algoritmos genéticos se han aplicado con éxito a una gran variedad de problemas de ingeniería. Este trabajo propone un algoritmo genético para resolver el problema del flujo óptimo de potencia teniendo en cuenta solamente la potencia activa del sistema eléctrico de potencia. A diferencia de las tradicionales técnicas de programación matemática, el algoritmo genético puede usar funciones objetivo sin tener conocimiento del problema que va a resolver, esta característica hace que el algoritmo sea sencillo de aplicar, además permite la inclusión de restricciones no lineales sin necesidad de obtener un equivalente lineal de ellas, lo cual acerca más la solución obtenida a la realidad del problema por resolver. Este método se probará utilizando el sistema de prueba IEEE 30 y 118 buses.

Palabras clave: flujo óptimo de potencia (FOP), algoritmos genéticos (AG), software Matlab, despacho económico.

ABSTRACT

Genetic algorithms have been successfully applied to a variety of engineering problems. This paper proposes using a genetic algorithm to solve the problem of optimal power flow, only taking the active power of the power system into account. Unlike traditional mathematical programming techniques, GA can use objective functions without any knowledge of the problem to be solved. This makes the algorithm easy to implement and also allows the inclusion of nonlinear constraints without the need of a linear equivalent, which brings the obtained solution much closer to the reality of the problem to be solved. This method will be tested with the IEEE 30 test system and 118 buses.

Keywords: optimal power flow, genetic algorithm, Matlab, economic dispatch.

FECHA DE ENVÍO: 21 DE OCTUBRE DE 2010 • FECHA DE ACEPTACIÓN: 12 DE DICIEMBRE DE 2010

* Profesor Asistente, Programa Eléctrica, Facultad Ingeniería, Universidad de La Salle. Correo electrónico: emuela@unisalle.edu.co.

** Ingetec, Bogotá. Correo electrónico: caribbean_cool18@hotmail.com.

*** Ingetec, Bogotá. Correo electrónico: diegoleon@ingetec.com.co.

Introducción

El propósito fundamental de la existencia de un sistema eléctrico de potencia es suministrar energía eléctrica a sus clientes. Esta labor debe realizarse de la manera más económica posible y con un nivel de confiabilidad aceptable. Los clientes conectados a las redes del sistema eléctrico esperan que el suministro de energía esté continuamente disponible, respondiendo en forma inmediata a las fluctuaciones de la demanda (Arguello, 1988; Wood y Wollenberg, 1996).

La solución al problema de optimizar la generación, en tanto se acatan los límites en las líneas de transmisión, consiste en combinar el despacho económico con los flujos de potencia. El resultado se conoce como flujo óptimo de potencia (FOP). Hay varios métodos para resolver el FOP, como son la programación lineal, el gradiente de base¹, la programación cuadrática entre otras (Pandya y Joshi, 2005).

No obstante, los métodos mencionados presentan tres problemas; en primer lugar no pueden proporcionar una solución óptima global sino un óptimo a nivel local; en segundo lugar, estos métodos se basan en hipótesis de continuidad y diferenciabilidad de la función objetivo y, por último, todos estos métodos no pueden aplicarse con variables discretas, como son los *taps* de los transformadores (Pandya y Joshi, 2005).

Este trabajo presenta un algoritmo genético aplicado a la solución del problema de flujo óptimo de potencia (FOP). Puesto que los algoritmos genéticos (AG) son un método adecuado para solucionar este problema, debido a que mejora los inconvenientes de los métodos convencionales de solución, expresados en el párrafo anterior. Este algoritmo genético se probará en los sistemas de prueba IEEE 30 buses e IEEE 118 buses, para compararlo con otro método de solución, como el software Gams Power, y así mostrar la eficacia del algoritmo genético.

Flujo óptimo de potencia

El flujo óptimo de potencia nace con el problema del despacho económico. Este problema empezó a ser analizado desde 1920 cuando se debió repartir la carga total

¹ Método el cual resuelve el FOP, mediante repetidas resoluciones de una aproximación.

de un sistema entre las unidades generadoras de dicho sistema (Arguello, 1988). El objetivo del problema del flujo de potencia óptimo consiste en determinar la producción de potencia de cada generador de modo que toda la demanda se satisfaga con un costo mínimo al tiempo que se respetan las restricciones propias de la red. Hacia 1930 se tomaron dos criterios para resolver el problema:

- El método de carga base en el que la unidad más eficiente era la primera en tomar la carga hasta su máxima capacidad; luego, la segunda unidad más eficiente hasta su máxima capacidad y así sucesivamente hasta satisfacer la demanda.
- El mejor punto de carga, en el que las unidades tomaban carga hasta alcanzar su punto de mínimo calentamiento empezando por la unidad más eficiente y cargando la unidad menos eficiente al final (Arguello, 1988).

Para la década del cincuenta aparece el flujo de potencia en las computadoras digitales². Con estas bases a principios de los sesenta, Squires y J. Carpentier formulan métodos más poderosos³ que no requieren suposiciones en las que se consideraban costos incrementales de combustible con las pérdidas incrementales, que se hacían en la matriz de coeficiente B , y que pueden incluir un modelo exacto de la red (Arguello, 1988; Wood y Wollenberg, 1996).

Además de satisfacer la demanda, los valores de tensión a lo largo de la red deben mantenerse en niveles aceptables. La potencia reactiva debe transmitirse a lo largo de la red, y su demanda debe ser satisfecha. La potencia activa neta (generación menos demanda) que llega a un nodo debe expresarse como función de todos los tensiones y ángulos en la red (Castillo et ál., 2002).

$$P_{Gi} - P_{Di} = v_i \sum_{k=1}^n y_{ik} v_k \cos(\delta_i - \delta_k - \theta_{ik}) \quad (1)$$

Donde:

P_{Gi} : potencia activa generada en el nodo i

P_{Di} : potencia activa demandada en el nodo i

² Para reducir las suposiciones hechas en la matriz de coeficientes.

³ Lo que se conoce como flujo óptimo de potencia.

v_i : magnitud de tensión

S_i : ángulo en el nodo i

Y_{ik} : módulo del elemento ij de la matriz Y barra

θ_{ik} : argumento de una constante compleja que depende de la topología y la estructura física de la red, y

n : número de nodos de la red

La magnitud de la tensión de todo nodo debe estar limitada superior e inferiormente

$$\underline{V}_i < v_i < \overline{V}_i \quad (2)$$

Donde:

\underline{V}_i : cota inferior para la magnitud de la tensión en el nodo i , y

\overline{V}_i : cota superior.

Los generadores pueden producir potencia activa por encima de cierta cota inferior y por debajo de cierta cota superior:

$$\underline{P}_{Gi} < P_{Gi} < \overline{P}_{Gi} \quad (3)$$

Donde:

\underline{P}_{Gi} : mínima potencia activa que puede salir del generador i , y

\overline{P}_{Gi} : potencia máxima

Definido C_i como el costo de producción en el generador i , el problema del flujo óptimo de potencia se puede representar matemáticamente como la minimización de la sumatoria de costos de todas las unidades de generación de un sistema de potencia, así:

$$Z = \sum_{i=1}^n C_i \quad (4)$$

Donde:

$$C_i = a_i^2 P_{gi} + b_i P_{gi} + c_i \quad (5)$$

En general, el problema del flujo óptimo de potencia consiste en determinar cómo suministrar la demanda de manera óptima, dentro de límites operativos y cumpliendo las restricciones de seguridad.

La formulación presentada anteriormente considera el caso de tener sistemas con generación puramente térmicos (costos de generación por consumo de combustible) (Castillo et ál., 2002).

Algoritmos genéticos

Los algoritmos genéticos (AG) surgen inspirados en el fenómeno evolutivo y selectivo, sucediendo en una dinámica de poblaciones y ciertas propiedades de los sistemas biológicos: la capacidad de autorreplicación de los ácidos nucleídos transmitiendo información genética entre generaciones, la mutación aumentando la diversidad, la competición de seres vivos por recursos limitados del entorno (Coley, 1999; Gopi, 2007).

Si bien esto sucede en nuestro planeta desde hace 3.500 millones de años, en 1944 fue cuando Oswald Avery et ál. conjeturaron que el soporte físico de la información genética transmitida entre generaciones es una molécula de ADN (Coley, 1999; Gopi, 2007). En 1975, John Holland (Coello, 2004) pone en escena los algoritmos genéticos, entre las capacidades de los algoritmos genéticos (AG) en optimización; se puede destacar las siguientes:

- Son métodos iterativos que convergen a partir de un conjunto inicial y arbitrario de puntos del espacio de búsqueda.
- La convergencia no está condicionada por requisitos de continuidad o diferenciabilidad del funcional por maximizar o minimizar.

- En su convergencia al óptimo global escapan de la atracción de múltiples óptimos locales.
- Son aplicables y eficientes en la optimización multicriterio.

Un algoritmo genético contiene una serie de procesos, similares a los observados en la evolución biológica de las especies, que permiten solucionar un problema; éstos se describen a continuación (Coley, 1999; Gopi, 2007).

Selección

La selección es el proceso en el cual se escogen los individuos que integrarán las parejas que van a reproducirse; todos los individuos pueden participar en la formación de dichas parejas. La selección se realiza al azar y los métodos más comunes son la selección por ruleta, selección basada en el rango y selección por torneo (Sivanandam y Deepa, 2008; Coello, 2004).

Cruce

El operador cruce realiza la reproducción de cada pareja de individuos; este operador permite crear individuos nuevos y mejor adaptados, combinando el material genético de los padres. Existen muchas formas de cruzar dos individuos, entre ellas se encuentran el cruce en un punto, el cruce en varios puntos, el cruce uniforme y el cruce aritmético (Sivanandam y Deepa, 2008; Coello, 2004).

Mutación

El operador mutación cambia el valor de uno o varios genes⁴ de un individuo, alterando su material genético. En la evolución biológica de las especies, las mutaciones son sucesos bastante poco comunes; en algunos casos producen la muerte del organismo, pero en promedio contribuyen a la diversidad genética. En los algoritmos genéticos las mutaciones tienen la misma función y su probabilidad de ocurrencia debe ser baja⁵ (Sivanandam y Deepa, 2008; Coello, 2004).

⁴ Cada una de las partículas dispuestas en un orden fijo a lo largo del ADN que determinan la aparición de los caracteres hereditarios en los organismos. En un algoritmo genético, cada uno de los elementos del cromosoma que puede tomar un valor numérico.

⁵ La sustitución de un gen para mejorar las habilidades de las especies; en la mayoría de las veces, la especie desaparece y en muy contadas ocasiones evoluciona.

Funcionamiento de un algoritmo genético

El funcionamiento de un algoritmo genético comienza generando la población inicial, la cual constituye la primera generación⁶ de individuos; también puede utilizarse como población inicial, un conjunto de individuos que representen soluciones aproximadas al problema, obtenidas por algún otro método de búsqueda (Gopi, 2007).

Cuando se tiene la población inicial, se evalúan todos los individuos por medio de una función de aptitud y se le asigna, a cada uno, su valor de fenotipo⁷; luego, utilizando un método de selección, se construyen las parejas de individuos que van a reproducirse; la cantidad de parejas es determinada por el diseñador del algoritmo genético (Gopi, 2007; Sivanandam y Deepa, 2008). Después de tener las parejas de individuos, se realiza la reproducción utilizando un operador de cruce; esto permite obtener un nuevo grupo de individuos, con diferentes características de los actuales.

Los descendientes de cada pareja pueden ser obtenidos de la aplicación de uno o varios operadores de cruce; la cantidad de descendientes la determina el criterio del diseñador del algoritmo genético; generalmente se obtienen dos descendientes de cada pareja y se aplica un operador de cruce, cambiando el orden de los padres, al aplicar el operador, para obtener cada descendiente (Gopi, 2007; Sivanandam y Deepa, 2008).

Después de la reproducción, se determina aleatoriamente, utilizando la probabilidad de mutación, si se aplica el operador mutación a cada uno de los descendientes de las parejas. Cuando se han aplicado los operadores genéticos, los descendientes de las parejas pueden sustituir a la población actual completa o a una parte de ella. También se puede evaluar a los nuevos individuos con la función de aptitud y que los mejores sean los que sustituyan a los peores de la población actual (Gopi, 2007; Sivanandam y Deepa, 2008). Esta nueva población, formada por los descendientes de la población anterior y tal vez por los mejores individuos de dicha población, constituye la segunda generación de individuos; todos los individuos de esta nueva

⁶ Serie de etapas en la que los descendientes rempazan a los progenitores.

⁷ El nivel de adaptación de un organismo con su medioambiente. En los algoritmos genéticos, el valor que toma un individuo de acuerdo con su aptitud para resolver el problema.

población son evaluados para determinar si se cumple el criterio de convergencia⁸ del algoritmo, el cual es definido por el diseñador del algoritmo; en caso contrario, se repite todo el proceso anterior sobre la nueva población, hasta que se cumpla el criterio (Gopi, 2007; Sivanandam y Deepa, 2008). El funcionamiento de un algoritmo genético se resume en el diagrama, mostrado en la figura 1.

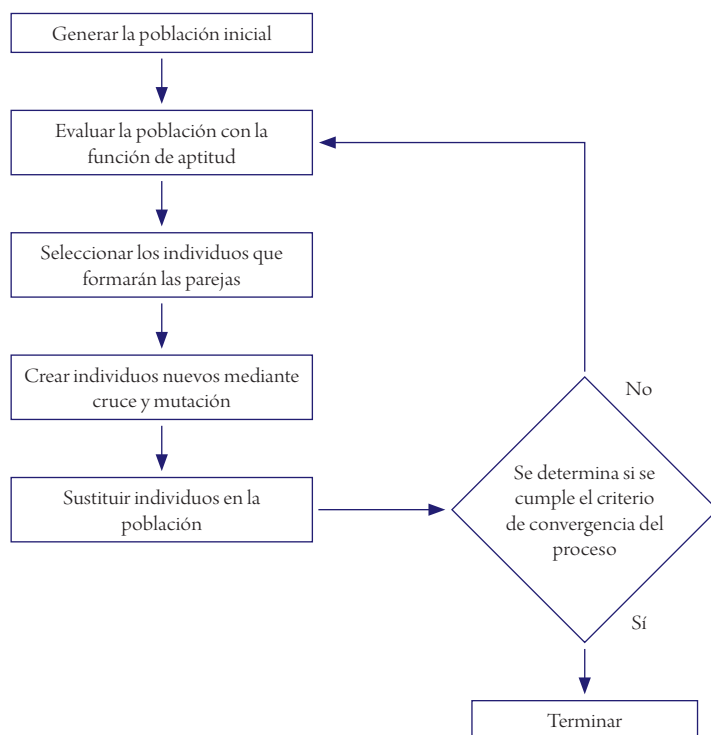


Figura 1. Diagrama de flujo de un algoritmo genético

Fuente: elaboración de los autores.

En resumen, el algoritmo recibe como entrada una población de individuos, y a partir de éstos genera nuevas poblaciones, donde algunos individuos desaparecen mientras que otros se cruzan con los mejores individuos posibles hasta que se encuentra una solución satisfactoria, o hasta que se cumple alguna otra condición de convergencia. A lo largo del algoritmo, los individuos de una iteración, serán llamados población (Gopi, 2007).

⁸ Aproximarse a un valor numérico.

Aplicación del algoritmo genético al flujo óptimo de potencia

El objetivo del algoritmo genético es solucionar el problema de flujo óptimo de potencia, el cual consiste en obtener los costos de despacho económico y la generación total del sistema teniendo en cuenta las restricciones técnicas y operativas del sistema eléctrico en estudio. Cada individuo de la población contendrá, en forma codificada, los valores de la potencia suministrada en las barras de generación; utilizando estos valores, se calculan nuevos valores de potencia suministrada que entreguen un menor despacho económico hasta encontrar el despacho económico más óptimo, y después de ejecutar el algoritmo se muestra un reporte con el despacho económico. La aptitud⁹ de un individuo para resolver el problema de flujo óptimo de potencia, se determina en una función que tiene como entrada el genotipo¹⁰ de un individuo y su salida es el fenotipo¹¹ de ese individuo. Un algoritmo genético puede realizarse mediante cualquier lenguaje de programación (Gopi, 2007; Sivanandam y Deepa, 2008). En este proyecto se utilizó el software Matlab (MathWorks, Inc., 2004-2009); este algoritmo se muestra en la figura 2.

Resultados obtenidos

Los algoritmos empleados en las simulaciones fueron programados en el lenguaje matemático de Matlab 7.6.0 (R2008a) (MathWorks, Inc., 2004-2009). Las simulaciones presentadas en el presente capítulo se llevaron a cabo en un computador con procesador AMD Athlon (TM) 64x2 Dual Core Processor 4000+2.09 GHz, 1,87 GB de memoria RAM y con sistema operativo Windows XP Profesional.

Primero se toma un sistema de prueba base, tomando el tamaño de la población aleatoriamente; puesto que los algoritmos genéticos no poseen una formulación matemática, son técnicas que se basan en parámetros probabilísticos, se aplica el operador cruce y finalmente el número de iteraciones, que en este caso son las generaciones; las pruebas —realizadas para la ejecución del algoritmo genético

⁹ Valor numérico asignado a un individuo de una población, el cual indica qué tan bueno es este individuo al resolver el problema.

¹⁰ Forma codificada de una posible solución a un problema, en los genes de un individuo.

¹¹ Nivel de adaptación de un organismo con su medioambiente. En los algoritmos genéticos, el valor que toma un individuo de acuerdo con su aptitud para resolver el problema.

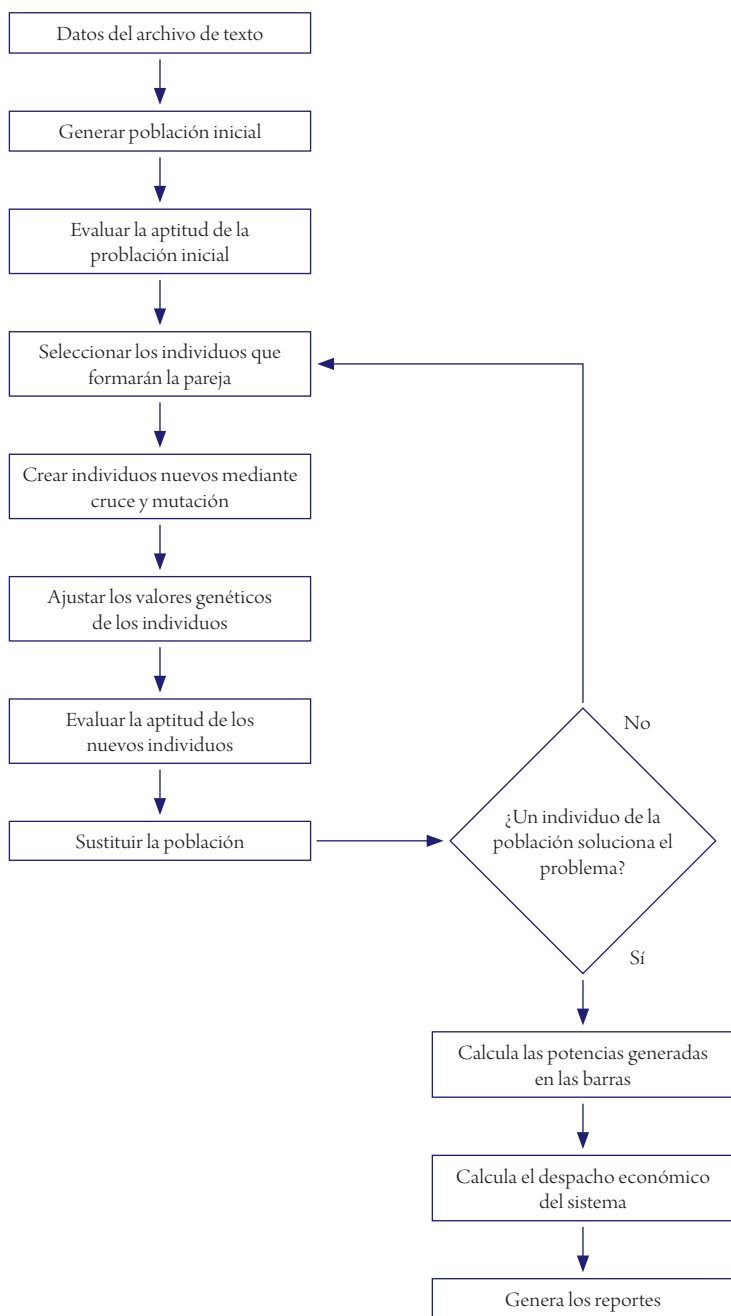


Figura 2. Diagrama del algoritmo de solución

Fuente: elaboración de los autores.

enfocado en la solución de flujo óptimo de potencia— fueron en los sistemas de prueba IEEE 30 buses e IEEE 118 buses.

Resultados simulación IEEE 30 buses

Para la simulación de este sistema de prueba se crearon dos códigos, los cuales fueron llamados “fop30bus” y “fopag30bus”. En el primer código se encuentran los datos de entrada del sistema, las restricciones de tensión y la solución del flujo de potencia, y en el segundo código, las funciones que va a ejecutar el algoritmo genético; con los resultados obtenidos, se ejecuta el programa Gams Power para comparar los resultados. Después de realizar diferentes pruebas y ver los resultados obtenidos, se llega a la conclusión que la mejor configuración para el sistema IEEE 30 buses, fue la siguiente:

- Número de generaciones de 200.
- Población de 100 individuos.
- Porcentaje de cruce del 90%.
- Tiempo computacional de ejecución no limitado.

En la tabla 1 se muestran los resultados obtenidos mediante el software de solución Fopag y el software de solución Gams Power.

Tabla 1. Comparación de potencias en generadores entre Matlab y Gams Power

	FOPAG	GAMS
Generador	P (MW)	P (MW)
1	213,222	212,088
2	36,309	36,204
5	28,364	29,384
8	11,761	13,285
11	5,566	4,878
13	0	0
Total	295,222	305,839

Fuente: elaboración de los autores.

En la figura 3 se observan las gráficas de las tensiones en buses de los dos programas para la solución de flujo óptimo de potencia. En la figura 4 se observan las gráficas obtenidas de la ejecución de los dos programas para la solución de flujo óptimo de potencia.

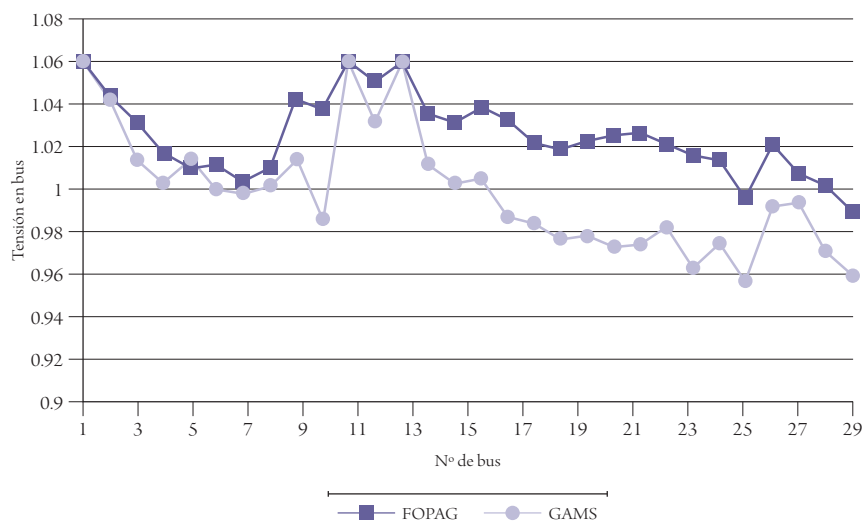


Figura 3. Comparación de tensiones en buses entre Matlab y Gams Power IEEE 30 buses

Fuente: elaboración de los autores.

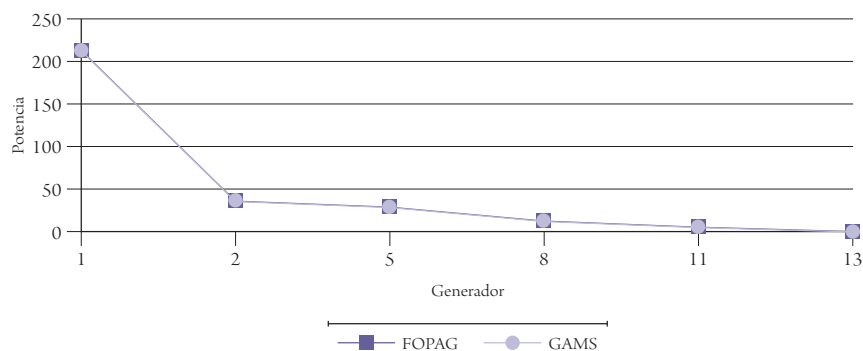


Figura 4. Comparación de potencias en generadores entre Matlab y Gams Power IEEE 30 buses

Fuente: elaboración de los autores.

El costo total por despacho económico obtenido con el algoritmo genético es de 8904,94 \$/h y el obtenido por Gams Power es de 8934.74 \$/h.

Resultados simulación IEEE 118 buses

Para la simulación de este sistema de prueba se crearon dos códigos, los cuales fueron llamados “fop118bus” y “fopag118bus”. En el primer código se encuentran los datos de entrada del sistema, las restricciones de tensión y la solución del flujo de potencia, y en el segundo código, las funciones que va a ejecutar el algoritmo genético; con los resultados obtenidos, se ejecuta el programa Gams Power para comparar los resultados.

Después de realizar diferentes pruebas y ver los resultados obtenidos, se llega a la conclusión que la mejor configuración para el sistema IEEE 118 buses, fue la siguiente:

- Número de generaciones de 1100.
- Población de 100 individuos.
- Porcentaje de cruce del 80%.
- Tiempo computacional de ejecución no limitado.

En la tabla 2 se muestran los resultados obtenidos mediante el software de solución Fopag y el software de solución Gams Power, para el sistema de prueba IEEE 118 buses. En la figura 5 se observan las gráficas de las tensiones en buses de los dos programas para la solución de flujo óptimo de potencia. En la figura 6 se observan las gráficas obtenidas de la ejecución de los dos programas para la solución de flujo óptimo de potencia. El costo total por despacho económico obtenido con el algoritmo genético es de $1,0682 \times 10^5$ \$/h, y el obtenido por medio de Gams Power es de $1,2973 \times 10^5$ \$/h.

Tabla 2. Comparación de potencias en generadores entre Matlab y Gams Power IEE 118 buses

GENE- RADOR	FOPAG	GAMS	GENE- RADOR	FOPAG	GAMS	GENE- RADOR	FOPAG	GAMS
	P (MW)	P (MW)		P (MW)	P (MW)		P (MW)	P (MW)
1	0	26,261	42	11,0369	40,935	80	236,7379	430,693
4	11,1236	0	46	29,4648	19,050	85	43,8981	0
6	25,613	0	49	15,0833	193,294	87	32,6008	3,626
8	14,4342	0	54	91,4861	49,538	89	108,3551	501,076
10	203,0057	401,862	55	68,626	32,047	90	41,1951	0
12	29,3743	85,776	56	57,9475	32,466	91	17,9131	0
15	46,1072	20,942	59	255	149,684	92	77,8583	0
18	67,2778	13,335	61	214,0777	148,394	99	30,8838	0
19	56,5866	21,655	62	33,1635	0	100	66,4686	231,172
24	15,9714	0	65	55,1254	352,190	103	38,774	38,225
25	66,9717	193,849	66	492	348,813	104	78,3443	0
26	224,0816	279,803	69	27,823	453,517	105	57,793	5,384
27	93,2743	10,059	70	100	0	107	100	29,395
31	14,6204	7,250	72	7,9294	0	110	93,9373	7,064
32	172,7076	15,001	73	100	0	111	54,4234	35,232
34	69,0169	4,593	74	53,1726	18,297	112	171,6832	36,671
36	71,4451	10,390	76	77,3969	24,358	113	1,1844	0
40	77,1153	49,155	77	34,2978	0	116	70,1619	0
Total							4304,57	4321,05

Fuente: elaboración de los autores.

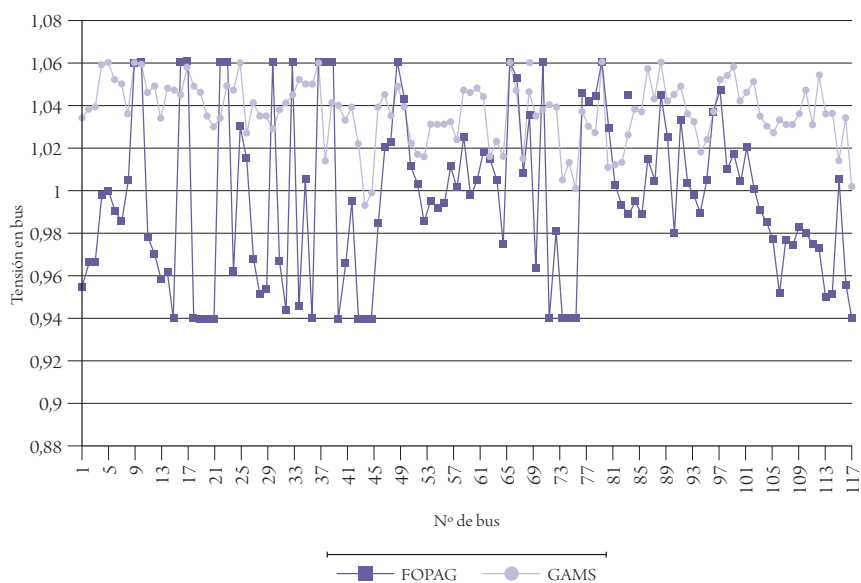


Figura 5. Comparación de tensiones en buses entre Matlab y Gams Power IEEE 118 buses

Fuente: elaboración de los autores.

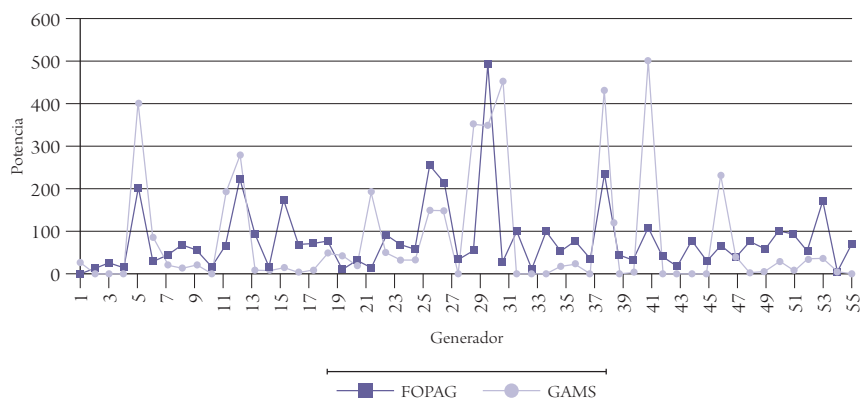


Figura 6. Comparación de potencias en generadores entre Matlab y Gams Power IEEE 118

Fuente: elaboración de los autores.

Conclusiones

Se obtuvieron mejores resultados (menores costos) aplicando los algoritmos genéticos para la solución del flujo óptimo de potencia en comparación con el método de gradiente utilizado por el software Gams. Se obtuvo un valor para el sistema de prueba IEEE 30 buses de 8904.9 \$/h de costo total de generación, que comparado con el software de solución Gams, es 0,33% inferior por hora, y para el caso del sistema IEEE 118 buses el resultado fue de $1,0682 \times 10^5$ \$/h de costo de generación, que comparado con el software de solución Gams es 0,17% inferior por hora.

Los resultados parecen reducciones marginales, pero si se observan los sistemas de prueba están constituidos por 6 y 54 generadores, respectivamente. En sistemas reales, el número de generadores es mucho mayor y el movimiento de dinero es, sin duda, mucho mayor que los valores manejados en los ejemplos presentados. Luego, una reducción similar evaluada en sistemas reales, no en porcentaje, sino en valores reales (dólares, pesos, euros) puede ser representativa.

Por otro lado, se obtuvo una reducción de 10,617 MW de generación para el sistema IEEE 30 buses, y para el sistema IEEE 118 buses se obtuvo una reducción de 16,48 MW de generación lo que demuestra que los algoritmos genéticos son una buena alternativa para la solución de este problema. Tales reducciones son consecuencia de una redistribución de la contribución que cada generador hace al cubrimiento de la demanda total. Tal redistribución no afecta la calidad del sistema, como evaluada en el nivel de tensión de cada barra de los sistemas de prueba, esto se puede apreciar en la figura 5, donde se puede comparar las tensiones de cada barra para los dos métodos.

Finalmente, se puede apreciar que en la figura 6 que desde la producción individual se presentan cambios significativos en algunos de los generadores, lo cual indica que para este problema específico, la no linealidad del problema no afecta la calidad de las soluciones obtenidas por los algoritmos genéticos, lo cual justifica enormemente su aplicación en este campo de ingeniería.

Referencias

- Arguello, G. (1988). *Análisis y Control de Sistemas Eléctricos de Potencia*. Instituto Ecuatoriano de Electrificación, Quito, Ecuador.
- Castillo, E. et ál. (2002). *Formulación y resolución de modelos de programación Matemática en Ingeniería y Ciencia*.
- Coello, C.A. (2004). *La importancia de la representación en los algoritmos Genéticos*.
- Coley, D.A. (1999). An introduction to genetic algorithms for scientist and engineers. In: *World Scientific Publishing*.
- Gopi, E.S. (2007). *Algorithm collections for digital signal processing applications using Matlab*. Springer ISBN 978-1-4020-6409-8 All Rights Reserved©.
- Pandya, K.S. & Joshi, S.K. (2005). A Survey of Optimal Power Flow Methods. In: *Journal of Theoretical and Applied Information Technology*.
- Sivanandam, S.N. & Deepa, S.N. (2008). *Introduction to Genetic Algorithms*. Springer ISBN 978-3-540-73189-4 Springer Berlin Heidelberg New York All Rights Reserved©.
- The MathWorks, Inc. *Genetic Algorithm and Direct Search Toolbox™ User's guide2*. Copyright 2004-2009 by The MathWorks, Inc.
- Wood, A.J. and Wollenberg, B.F. (1996). *Power Generation Operation and Control*. New York, NY: John Wiley & Sons, Inc.

