

OPTIMIZACIÓN DE FUNCIONES A TRAVÉS DE UN ALGORITMO GENÉTICO

FUNCTION OPTIMIZATION WITH BASIC GENETIC ALGORITHM

Yohoni Cuenca Sarzuri*

RESUMEN	ABSTRACT	RESUMO
<p>El presente trabajo describe la optimización de una función simple por máximos y mínimos, comparando este resultado con el conseguido a través de un módulo macro en hoja Excel del algoritmo genético, que inicia la búsqueda de un óptimo a partir de una población aleatoria de soluciones posibles seguido de operaciones genéticas: (asignación por ruleta, cruce, mutación y elitismo) hasta que casi todas las poblaciones y generaciones convergen en un valor similar al obtenido por máximos y mínimos.</p> <p>Si bien la función ejemplo examinada es muy simple (sólo presenta un máximo para el intervalo 0 a 31), ha permitido verificar la capacidad que tiene el algoritmo genético y sus operadores para llegar a encontrar un valor óptimo global a partir de datos iniciales no estructurados rigurosamente (heurística).</p> <p>PALABRAS CLAVES: Algoritmo genético, operaciones genéticas aleatorias, heurística, computación evolutiva.</p>	<p>This paper describes the optimization of a simple function by maximum and minimum, comparing this result with the one obtained through a macro module in Excel sheet of the genetic algorithm, which starts the search for an optimum from a random population of possible solutions followed by genetic operations: (assignment by roulette, crossing, mutation and elitism) until almost all populations and generations converge in a value similar to that obtained by maximums and minimums.</p> <p>Although the example function examined is very simple (it only presents a maximum for the interval 0 to 31), it has allowed to verify the ability of the genetic algorithm and its operators to reach an optimal global value from unstructured initial data. rigorously (heuristic).</p> <p>KEYWORDS: Genetic algorithm, random genetic operations, heuristics, evolutionary computation</p>	<p>Este artigo descreve a otimização de uma função simples por máximo e mínimo, comparando este resultado com o obtido através de um módulo macro na planilha Excel do algoritmo genético, que inicia a busca por um ótimo de uma população aleatória de possíveis soluções seguido de operações genéticas: (atribuição por roleta, cruzamento, mutação e elitismo) até que quase todas as populações e gerações convergem em um valor similar ao obtido por máximos e mínimos.</p> <p>Embora a função de exemplo examinada seja muito simples (apresenta apenas um máximo para o intervalo de 0 a 31), permitiu verificar a capacidade do algoritmo genético e seus operadores para alcançar um valor global ótimo a partir de dados iniciais não estruturados. rigorosamente (heurística).</p> <p>PALAVRAS-CHAVE: Algoritmo genético, operações genéticas aleatórias, heurísticas, computação evolutiva.</p>
<p>History of the article: Received 12/11/2018. Style review 15/11/2018. Accepted 25/01/2019.</p>		

INTRODUCCIÓN

Por lo general el término algoritmo¹, está relacionado con un procedimiento lógico para obtener el resultado de un problema complicado, haciendo uso repetido de otro método matemático de cálculo más sencillo.

En la actualidad, los algoritmos son parte de métodos de resolución de problemas que siguen una sucesión de pasos, simbolizados, a través de diagramas de flujo como es el caso de los programas de computación, o refiriéndose a algún modelo de la naturaleza.

Al respecto, John Henry Holland propuso en 1970 la aplicación de un tipo de algoritmo similar al que controla el comportamiento de la herencia biológica² y genética molecular. En consecuencia, un algoritmo genético básico, promueve la evolución de un grupo de formaciones binarias a través de acciones aleatorias, parecidas a las que gobiernan la herencia biológica (recombinaciones genéticas y mutaciones), así como también la selección de acuerdo con algún criterio, para decidir cuáles son los Individuos mejor adaptados que sobreviven al medio (en equivalencia a un óptimo), y cuáles los menos aptos, a ser descartados. Esta primera característica, permite utilizar este algoritmo genético³ para resolver problemas de optimización.

De acuerdo con este enfoque, en el presente artículo se considera el procedimiento que sigue una aplicación

algorítmica genética básica, para optimizar una función, encontrando el valor x que permita a la función $f(x)$ alcanza su máximo valor.

DESARROLLO

Si por ejemplo la función ha optimizar corresponde a la siguiente expresión algebraica:

$$f(x) = x^3 - 60x^2 + 900x + 100 \quad (1)$$

Tradicionalmente se pueden encontrar puntos críticos (máximos y mínimos), a partir de la primera y segunda derivadas. Por el criterio de la primera derivada, si la función $f'(x_0)$ es igual a cero o no existe, entonces hay un punto crítico en x_0 . Así se tiene que los puntos críticos de la función son: $x_1 = 10$, $x_2 = 30$.

$$\begin{aligned} (x^3 - 60x^2 + 900x + 100)' &= 0 \\ 3x^2 - 120x + 900 &= 0 \\ 3(x-10)(x-30) &= 0 \\ x_1 &= 10, x_2 = 30 \end{aligned}$$

Por la segunda derivada, si $f''(x_0) < 0$ entonces x_0 es un máximo.

$$\begin{aligned} f'' &= ((x^3 - 60x^2 + 900x + 100))' \\ &= (3x^2 - 120x + 900)' \\ &= 6x - 120 \end{aligned}$$

Evaluando las dos raíces se tiene:

$$\begin{aligned} 6(10) - 120 &= -60, \text{ como } -60 < 0 \text{ es } \textit{máximo}. \\ 6(30) - 120 &= 60, \text{ como } 60 > 0 \text{ es } \textit{mínimo} \end{aligned}$$

Por lo tanto $f(x) = x^3 - 60x^2 + 900x + 100$

$$\begin{aligned} \text{Cuando } x = 10, f(x) &= 4100 \text{ (máximo valor)} \\ \text{Cuando } x = 30, f(x) &= 100 \text{ (mínimo valor)} \end{aligned}$$

¹ Recordando por ejemplo, el uso de algoritmos aritméticos para efectuar operaciones tales como la multiplicación, división, obtención de raíces cuadradas, máximo común divisor y mínimo común múltiplo de un número mediante su descomposición en factores primos, división de un polinomio por $x - a$ mediante la regla de Ruffini, entre otros.

² Características, fisiológicas, morfológicas y bioquímicas de los seres vivos según el medio donde habitan, y que se transmiten de padres a hijos (descendencia).

³ Diferenciándose de otros algoritmos evolutivos por utilizar números binarios para representar a sus componentes.

Esta solución analítica puede comprobarse gráficamente, ver figura 1.

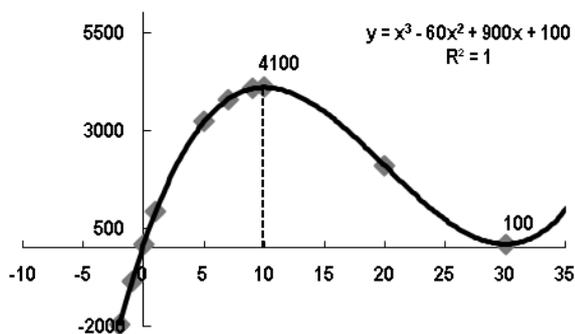
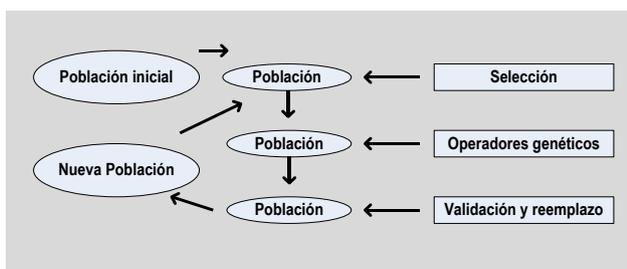


Figura 1: Representación gráfica máximo y mínimo $f(x) = x^3 - 60x^2 + 900x + 100$

Proceso iterativo del algoritmo genético básico

En la figura 2, se muestra de manera general el proceso cíclico del algoritmo genético. El proceso inicial con la definición de la población inicial sobre el cual se aplica procesos de selección, operadores genéticos y validación o reemplazo, hasta obtener una nueva población y el ciclo vuelve a repetirse.



Fuente: Elaboración propia

Figura 2: Proceso cíclico del algoritmo genético

Definición de la población inicial

El proceso de búsqueda parte de una población inicial, generalmente el procedimiento es aleatorio, cada uno de los individuos o soluciones que conforman la población inicial se define aleatoriamente del universo de soluciones.

Codificación de la población

- Desde una visión biológica la información genética de un organismo está en su cromosoma⁴ el cual contiene la información genética del organismo representado mediante sus alelos, la configuración del cromosoma expresado como genotipo⁵ tiene su efecto indicado como fenotipo⁶, el cual hace referencia a lo que se desarrollará a partir del genotipo.
- Refiriéndose a esta similitud, en los algoritmos genéticos los cromosomas pueden ser representados como cadenas binarias y a partir de este genotipo llegar al fenotipo que puede representar a un número o símbolo. La regla de correspondencia de genotipo a fenotipo describe un proceso de codificación y decodificación.

⁴ Estructura compleja pero muy bien organizada de información.

⁵ Conjunto de genes de un organismo.

⁶ Conjunto de rasgos característicos de un organismo.

Operadores genéticos

Para obtener poblaciones con mejores individuos es necesario hacer cambios en los cromosomas de la nueva población mediante los operadores genéticos, los operadores genéticos más usuales son la selección, cruce y mutación, aunque también se pueden utilizar otros operadores como: reagrupamiento, colonización-extinción o migración. A continuación se describen los operadores principales.

- Selección**, este operador permite conformar la nueva población, existen varios métodos como por ejemplo: selección por ruleta, selección por torneo, orden lineal y selección aleatoria.
- Cruce**, los métodos más usuales para el cruce son: cruce de un punto, dos puntos y cruce uniforme. También existen cruces donde la codificación no necesariamente es binaria, por ejemplo si los genotipos empleados son enteros o reales puede usarse operadores de cruce como: media, media geométrica, y extensión (Gestal, Rivero, Rabuñal, y Pazos, 2010).
- Mutación**, realiza un cambio aleatorio en el genotipo, generalmente en un alelo.
- Copia**, es una opción asexual que permite realizar la copia de un individuo en la nueva población.
- Elitismo**, es caso particular de la copia, ya que se realiza la copia de los mejores individuos en la nueva población.

Estrategia de reemplazo

- La transición de una población a una nueva población luego de aplicarse los operadores genéticos pasa por una estrategia de reemplazamiento o validación, que tiene la finalidad de reunir en la nueva población las mejores soluciones, las estrategias de reemplazamiento más usuales son: elitismo, reemplazar al peor de la población, torneo restringido, peor entre semejantes y algoritmo de crowding (remate determinístico).

Solución por algoritmo genético básico

Para la optimización de la función:

$$f(x) = x^3 - 60x^2 + 900x + 100$$

Aplicando un algoritmo genético básico, se considera una población inicial de soluciones y a partir de ésta, se construyen nuevas generaciones de soluciones que convengan a la solución global, bajo este enfoque solo las soluciones más aptas conforme a una evaluación, permanecerán en el tiempo y tendrán descendencia, mientras que las peores soluciones serán eliminadas.

Estructura del algoritmo genético

La estructura del algoritmo genético que optimiza la función se muestra en el pseudocódigo de la figura 3.

```

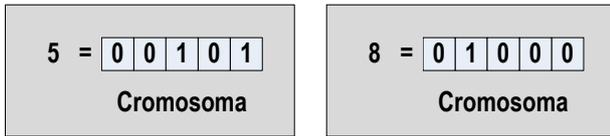
Función Algoritmo_genético
  Población_inicial = genera_aleatoriamente()
  Mientras la población no converge
    Cruzar padres
    Mutar aleatoriamente
    Nueva población = elitismo(población actual)
  Fin mientras
End función
    
```

Fuente: Elaboración propia

Figura 3: Pseudocódigo del algoritmo genético

Codificación y tamaño de ensayo de la población

Para la codificación de la población se considera la transformación de números del sistema decimal al binario, así cada número decimal estará representado por un cromosoma (binario con un número de bits⁷ = alelos), figura 4, con estas características el rango de forma corresponde a números enteros en el intervalo [0, 31], seleccionados aleatoriamente para una población de tamaño diez. En la tabla 1 se muestran las primeras operaciones de selección realizadas sobre la población inicial de ensayo.



Fuente: Elaboración propia

Figura 4: Codificación en binario

Tabla 1
Operaciones sobre el tamaño de población elegida aleatoriamente

C	2	3	4	5	6	7
1	13	3857	0,16587	0,16587	0,224	15
2	15	3475	0,14944	0,31532	0,045	13
3	8	3972	0,1708	0,48615	0,583	13
4	0	100	0,00430	0,49045	0,408	8
5	24	964	0,04145	0,53191	0,303	15
6	25	725	0,03118	0,56309	0,444	8
7	13	3857	0,16587	0,72896	0,728	13
8	9	4069	0,17499	0,90396	0,197	15
9	22	1508	0,06485	0,96881	0,515	24
10	25	725	0,03118	1,00000	0,325	8

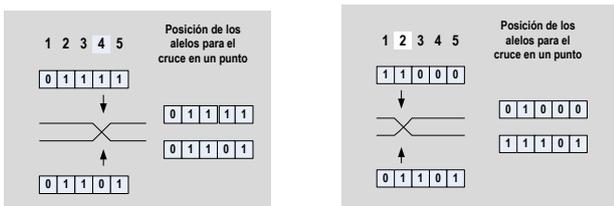
Referencias:

- C) Representa al número de cromosomas para el tamaño de la población de ensayo,
- 2) Población designada por un número x (sistema decimal),
- 3) Valores de la función para cada x elegido,
- 4) Proporción de cada $f_{(x)}$, el peso de estas funciones sirve para utilizar el método Aleatorio de la (ruleta) para obtener la nueva población,
- 5) Proporción acumulada de $f_{(x)}$,
- 6) Despliegue de números Random (aleatorios),
- 7) Nueva población elegida por un número x aleatorio (sistema decimal).

Fuente: Elaboración propia

Cruzamiento

Este operador considera el cruce de cromosomas en un punto, la posición de cruce se cuenta de izquierda a derecha entre los padres, ver figura 5.



Fuente: Elaboración propia

Figura 5: Dos cromosomas padres se cruzan en las posiciones indicadas generando nuevas cromosomas

⁷ La cantidad de alelos del cromosoma, está en función del número de individuos establecido para la población, y puede aproximarse por la relación: $n^{\circ}bits = \log población / \log 2$.

En la tabla 2 se muestran los resultados del cruzamiento de cromosomas. La columna binario indica la conversión del número decimal x' correspondiente a binario, la columna punto determina el lugar de cruce generado de manera aleatoria, sin embargo, existe la posibilidad de no realizar el cruzamiento si el número aleatorio es cinco, finalmente en la última columna se tiene una nueva población resultado del cruce.

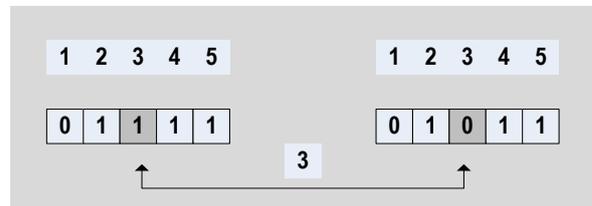
Tabla 2
Cruzamiento de cromosomas

x'	Binario	Punto de cruce	Cromosomas hijos
15	01111	4	01111
13	01101		01101
13	01101	5	01101
8	01000		01000
15	01111	2	01000
8	01000		01111
13	01101	4	01101
15	01111		01111
24	11000	3	11000
8	01000		01000

Fuente: Elaboración propia

Mutación

En esta operación se considera mutar (cambiar las características de un determinado porcentaje de los cromosomas de la población, para el ejemplo en cuestión, el porcentaje de mutación fue 10 por ciento, modificando uno de sus alelos a un valor opuesto. La figura 6 muestra esta operación.



Fuente: Elaboración propia

Figura 6: Mutación del cromosoma padre alelo 3

Tabla 3
Proceso de mutación

Cruce	Posición	Mutación	x''
01111	3	01011	11
01101	0	01101	13
01101	0	01101	13
01000	0	01000	8
01000	0	01000	8
01111	0	01111	15
01101	0	01101	13
01111	0	01111	15
11000	0	11000	24
01000	0	01000	8

Fuente: Elaboración propia

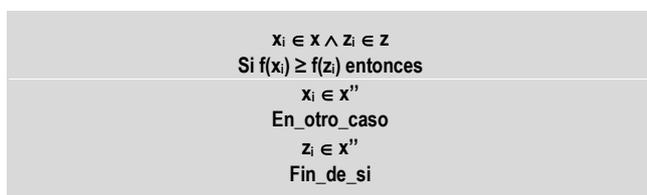
Los cambios realizados luego de realizar la mutación al 10 por ciento de la población se muestran en la tabla 3. En la columna posición se indica el alelo a mutar de manera aleatoria, el valor 0 en la posición indica que no se realizará la mutación, particularmente para esta tabla, el

factor aleatorio permite una mutación al 10 por ciento de la población. Las columnas cruce y mutación describen el cambio realizado después de la mutación, finalmente la columna x'' muestra la nueva población obtenida después del cruce y mutación.

Elitismo

Se ha demostrado que un algoritmo genético debe aplicar el operador genético de elitismo para converger hacia un óptimo. Este elemento puede verse como un mecanismo que valida la transición de una población inicial a otra mejor (Rudolph, 1994).

Bajo esta perspectiva, a partir de una población x se ha realizado operaciones de selección terminando en una población x' , después de las operaciones de cruce y mutación se genera otra población z . El proceso de validación trabaja sobre cada uno de los individuos o soluciones de la población z , si un individuo ha mejorado después de todo estos proceso pasa a formar parte de una nueva población x'' , representando a una población formada por las mejores soluciones. La figura 7 representa el mecanismo de validación utilizado.



Fuente: Elaboración propia

Figura 7: Mecanismo de validación de las soluciones

Así el mecanismo de validación permite que una solución x_i que pertenece a la población x , pueda permanecer en la población de los mejores x'' , siempre y cuando la solución z_i obtenida a partir de x_i resulte ser una solución deficiente, en otro caso la solución z_i ingresa a la población de los mejores x'' en lugar de x_i . Ver tabla 3.

Convergencia a la solución

Luego de ejecutarse el módulo macro con los parámetros:

- Número de hijos por generación = 10
- Numero de generaciones = 8
- Porcentaje de mutación = 20%

Se obtienen las siguientes generaciones de poblaciones descritas en la tabla 4.

Tabla 4
Generación de poblaciones

Gen	Población									
	1	2	3	4	5	6	7	8	9	10
1	6	10	2	18	5	28	3	13	8	27
2	14	10	10	18	6	6	3	13	10	2
3	10	10	10	6	10	10	6	10	10	10
4	10	10	10	14	10	10	10	10	10	10
5	10	10	10	10	10	10	10	10	10	10

La columna gen indica el número de generaciones, la convergencia a la solución sucede en la generación 5, las columnas del 1 al 10 representan a las diez soluciones de cada generación, así la población converge al máximo global $x=10$.

Fuente: Elaboración propia

Discusión de resultados

Encontrado el óptimo por derivación (máximos y mínimos) de la función ejemplo, y comparando con el obtenido, a través del algoritmo genético básico y sus operadores aleatorios de cruce, mutación, elitismo y convergencia partiendo de una población inicial, ambos coinciden en el mismo número de x , que permite a la función $f(x)$ alcanzar su máximo valor.

CONCLUSIONES

Si bien la función ejemplo examinada es muy simple (sólo presenta un máximo para el intervalo 0 a 31), ha permitido verificar la capacidad que tiene el algoritmo genético y sus operadores para llegar a encontrar un valor óptimo global a partir de datos iniciales no estructurados rigurosamente (heurística).

COMENTARIO FINAL

Importancia actual del algoritmo genético

El desarrollo específico de software y hardware en la programación por computadora, ha demostrado que un algoritmo genético es muy útil para solucionar problemas de optimización con o sin restricciones, siguiendo un proceso de selección que a través de varias generaciones sucesivas evoluciona hacia la solución óptima.

Entre las muchas aplicaciones del algoritmo genético por computadora, se tienen por ejemplo la optimización de funciones, el dilema del prisionero y el problema del vendedor viajero. (Michalewicz, 1996). La solución de sistemas de ecuaciones, el problema de la N reinas de ajedrez (Gestal, et-al, 2010). En consecuencia, el algoritmo genético se puede aplicar para resolver problemas donde los algoritmos de optimización estándar tienen indeterminaciones, así como también en problemas donde la función objetivo es discontinua, no diferenciable, estocástica o no lineal, ingresando en la clasificación de los métodos heurísticos de optimización, conjuntamente con los algoritmos de evolución diferencial, búsqueda diferencial, relajación dinámica, hill climbing (ascenso acelerado), Nelder-Mead, optimización por enjambre de partículas y optimización artificial de la colonia de abejas, por citar a los más actuales.

BIBLIOGRAFÍA

Optimización matemática, <https://es.m.wikipedia.org>, Consulta: 10/10/2018,
 Goldberg, D. E., Holland, J.H., 1988, Machine learning 3:95.<https://doi.org/10.1023/A:1022602019183>, Link Springer .com,
 Michalewicz, Z., 1996, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, ISBN: 978-3540606765, Alemania,
 Gestal, M., et-al., 2010, Introducción a los algoritmos genéticos y programación genética, monografía n° 140, Universidad de La Coruña, ISBN: 978-84-9749-422-9, Galicia – España,
 Rudolph, G., 1994, Convergence properties of canonical genetics algorithms, IEEE Transactions on neural networks, Volume: 5, Issue: 1, Jan, pp: 96–101, USA.

(*), Doctor en Ciencias y Humanidades – Mención Informática, Docente carrera de Mecánica Automotriz, Facultad de Tecnología – UMSA.