

Tarea 3: Algoritmos Genéticos

Juan Pablo Arango Atehortúa, Simón Zapata Caro

Medellín, Colombia

Universidad de Antioquia.

juan.arango17@udea.edu.co, simon.zapata@udea.edu.co.

Resumen—En este informe, se realiza una implementación de Algoritmos Genéticos (GA), los cuales son técnicas de inteligencia computacional, encargados de resolver problemas de optimización complejos, emulando los conceptos de Herencia Genética y Principio de Supervivencia. Utilizando esta técnica, se resuelven dos ejercicios clásicos de optimización, el llamado problema de la mochila (Knapsack Problem) y el problema de la mínima distancia dentro de un cluster disjunto. Para cada ejercicio, se plantea matemáticamente cada problema de optimización y se definen cada uno de los conceptos asociados a los algoritmos genéticos, tales como Población, cromosoma, alelos, función de salud (Fitness), método de cruce/mutación, entre otros.

Index Terms—Genetic Algorithms, Knapsack Problem, Disjoint clustering, Optimization.

I. INTRODUCCIÓN

Los algoritmos genéticos son una estrategia de inteligencia que tratan de emular los conceptos evolutivos de la naturaleza; son utilizados principalmente en *problemas de optimización* complejos, para los cuales las técnicas tradicionales tardarían demasiado tiempo en encontrar un resultado óptimo, o no es posible hacerlo.

Al desarrollar un algoritmo genético, es indispensable definir los siguientes componentes:

- Representación de los individuos
- Función de evaluación
- Población
- Mecanismo de selección de padres
- Operadores de variación (cruce o mutación)
- Mecanismo de supervivencia.

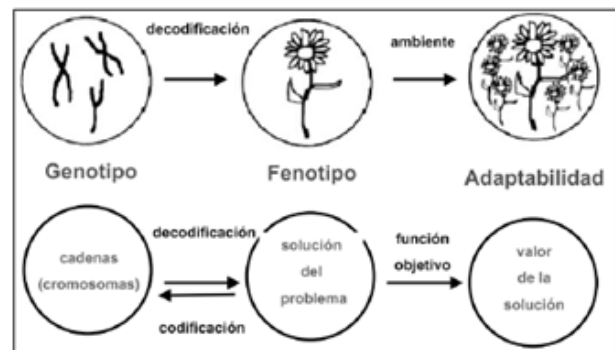


Fig 1. Representación de los individuos, paralelismo con la genética

Como se puede ver en la **Fig 1**, en la **Representación de los individuos** se establece un puente entre el problema real con sus variables características (*Fenotipo*) a una abstracción de dicho espacio real, que conforma un espacio de soluciones donde el proceso evolutivo toma lugar (*Genotipo*). Para dar solución a un problema utilizando algoritmos evolutivos, primero se deben definir un subconjunto de posibles soluciones en el Genotipo (*Población*), cada solución de este subconjunto se conoce como **Cromosoma**, conformado por un número de posiciones que representan la información codificada del problema real, conocidos como los **genes**, y los valores que pueden tomar (ya sea un problema, binario, entero/discreto o continuo) se conoce como los **alelos**. Hay distintas técnicas para definirse una población inicial, entre ellas, está tomar cromosomas aleatorios, o definirlos uniformemente distribuidos en el espacio solución del problema de optimización a resolver (también conocido como región factible o región de restricciones), o una mezcla de ambos. Ahora con la población establecida, se define una **función de evaluación** (también llamada función de Salud, función de Fitness), esta función es característica de cada problema, y es la que nos permitirá conocer que tan buena es una solución en particular (cromosoma), todo con el fin de llegar a un valor óptimo.

Entre los mecanismos de **Selección de padres**, se encuentran el de la *ruleta*, que, como su nombre lo dice, es asignar un área de una ruleta a cada cromosoma de la población dependiendo de su valor de fitness y poner a girar la misma; también existen los mecanismos de *torneo* (escoger un número de cromosomas y ponerlos a competir en función su valor de fitness) y de *ranking* (para cuando los cromosomas tienen valores de fitness similares), y también existen mecanismos híbridos definidos a partir del problema en cuestión en los que importa fijar una regla de decisión dependiente a las restricciones que tenga el problema. Al haber seleccionado los padres, se realizan las operaciones de cruce y/o de mutación; el **cruce** (*crossover* en inglés) consiste en combinar los genes de los padres por parejas y obtener su descendencia, imitando la **Herencia Genética**, es decir, el “hijo” tendrá características de los padres.

La **mutación** tiene como objetivo añadir combinaciones genéticas para favorecer la expansión por el espacio de búsqueda y evitar óptimos locales, esto a partir de variaciones sutiles a los genes de un cromosoma escogido. En el paso siguiente, se añaden los nuevos hijos a la población y, según un **mecanismo de supervivencia**, seleccionar la mejor población, es decir, los que serán los mejores padres para generar hijos óptimos (ya sea por edad o por fitness). Al finalizar, se escoge un **criterio de parada** para escoger el punto óptimo del problema de optimización. A continuación, se hace un resumen de todo lo que conlleva diseñar un algoritmo genético:

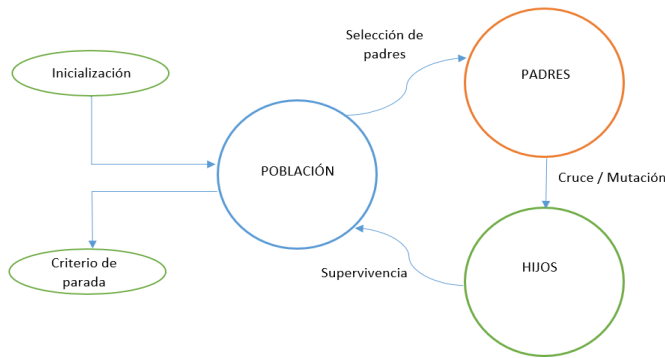


Fig 2. Resumen del proceso de un algoritmo genético.

Para asimilar los conceptos previamente revisados, se realiza la implementación de algoritmos genéticos que dan solución a dos problemas de optimización clásicos, el llamado “problema de la mochila” y un problema de agrupar N datos en K grupos (o clusters)

II. CONTENIDO

A. Solución del problema de la mochila (Knapsack Problem)

El problema de la mochila (KP) puede ser brevemente descrito de la siguiente manera: Tenemos un conjunto de N elementos que, inicialmente, queremos empaçar en una mochila, donde cada uno tiene un valor de beneficio b_j y un peso asociado ω_j ; la tarea es seleccionar un subconjunto de aquellos ítems que maximicen el beneficio total, mientras se mantiene el peso total a lo sumo en un valor W_{max} .

El modelamiento de este problema es ampliamente utilizado tanto en ámbitos de la ingeniería, como en ámbitos económicos.

Ahora, cuando el número de artículos tiende a tomar un valor alto, saber cuáles artículos empaçar en la mochila se vuelve una labor engorrosa utilizando métodos tradicionales, por lo cual se ve la necesidad de emplear una heurística para resolver dicho problema, y en el caso que nos compete, diseñar un algoritmo genético para hallar el punto óptimo.

De una manera formal, el problema de optimización es el siguiente:

Parámetros:

- b_j : beneficio del j -ésimo artículo
- ω_j : peso del j -ésimo artículo
- W_{max} : capacidad total de la maleta

Variables:

- x_j : decisión de llevar el j -ésimo artículo

Función de fitness:

$$\bullet \max: F(x_j) = \sum_{j=0}^{N-1} b_j x_j$$

Sujeta a:

- 1. $\sum_{j=0}^{N-1} \omega_j x_j \leq W_{max}$
- 1. $x_j \in [0, 1] \forall j = 0, 1, 2, \dots, N-1$

Fig 3. Definición formal del problema

A partir de las variables definidas, se define cada Cromosoma del Genotipo como un vector de tamaño N que contiene cada variable x_j ; los genes representarán a los N artículos y los valores que toman serán binarios (si el j -ésimo artículo se empaça o no en la mochila). En la función de fitness se toman en cuenta sólo los beneficios cuyos artículos se empaçan en la maleta (ver **Fig 3**)

- **Selección de supervivencia de la Población:** la población en cada generación será fija, con un valor de 20 Cromosomas; se inicializa de manera aleatoria, y para cada siguiente generación sobreviven los 20 cromosomas que, además de tener el mejor Fitness, cumplan con la restricción de capacidad máxima. Para el problema tratado, como se sugiere en [3], se podría penalizar los individuos no factibles (que no cumplan la condición) para darles posibilidad de dejar descendencia, especialmente porque estos individuos podrían generar hijos que sí sean factibles y optimicen el fitness (por ejemplo, si los máximos globales se ubican cerca de la frontera de la región factible), pero en nuestro caso, es muy poco probable que ocurra, debido a la naturaleza del problema (Restricciones y función de Fitness Lineal [4]); por tanto, para cada generación y a la hora de seleccionar los mejores padres para cruzar/mutar, se toma una posición elitista y se elimina a los individuos no factibles para reducir su efecto en próximas generaciones.

- **Selección de cruce/mutación de los padres:**

Se escoge el *método de la ruleta*, el cual se basa en calcular el fitness de cada individuo de la población, con base en éstos valores hallar una frecuencia relativa donde los mejores individuos (con mayor función de fitness) sean más propensos a ser escogidos como padres y escoger un número aleatorio (simulando el giro de una ruleta) para seleccionar cada Padre.

- **Estrategia de cruce:**

Se escoge el *cruce multipunto* (a dos puntos), que es seccionar cada pareja de padres en tres posiciones equidistantes de las N de cada cromosoma, e intercambiar la sección de la mitad, como se puede apreciar en la Fig 4.

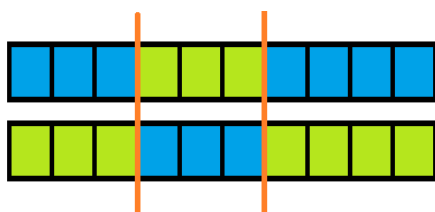


Fig 4. Cruce Multipunto

- **Estrategia de mutación:**

Para mutar cada padre seleccionado, se realizan un *bit-flip* a un gen, escogido aleatoriamente, como se muestra en la Fig 5, para que la población no se atasque en un óptimo local

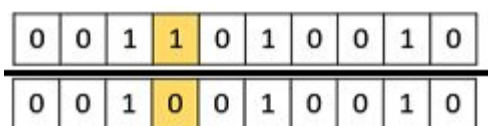


Fig 5. Bit Flip

- **Criterio de Parada:**

Iterar hasta 100 generaciones o hasta que se llegue a un óptimo.

- **Resultados:**

Se prueba para $N = 10$ artículos, con cada b_j y w_j

$$W_{max} = 0.6 * \sum_{\forall j} w_j$$
seleccionados aleatoriamente, y

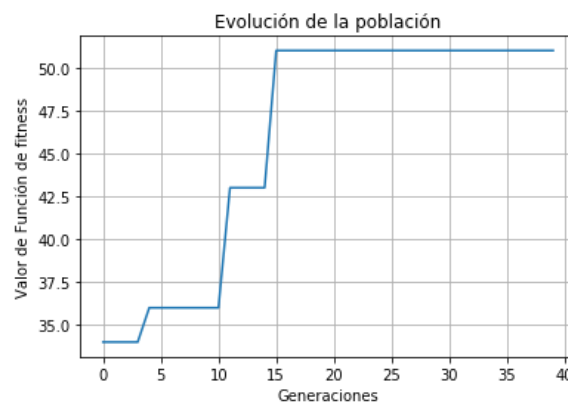


Fig 6. Evolución de la población, problema de la mochila

Capacidad: 37.8
 Valor óptimo: 51
 con un peso de: 34

	beneficios	pesos	variables
0	[9]	[9]	1
1	[7]	[3]	1
2	[9]	[7]	1
3	[1]	[7]	0
4	[9]	[9]	1
5	[0]	[9]	0
6	[1]	[9]	0
7	[8]	[5]	1
8	[9]	[1]	1
9	[0]	[4]	0

Fig 7. Resultados finales de la ejecución del proceso de optimización (para $N = 10$)

- **Análisis de los resultados:** Se observa que para $N = 10$ artículos, se obtiene una respuesta óptima en un tiempo corto, lo cual, era lo esperado, ahora, al ejecutarse el algoritmo aumentando el número de artículos, el tiempo no aumenta considerablemente y llega a una respuesta que, aunque no es óptima global, se acerca a esta.

B. Solución del problema de agrupamiento de N puntos en K clusters

N puntos que en general pueden ser de dimensión M , se ubican en el espacio, el objetivo de este problema es entonces dividirlos en K grupos utilizando el criterio de la mínima distancia entre los grupos, es decir aquellos puntos que sean más similares deben ser agrupados en el mismo grupo K_j .

Parámetros:

- N : número de elementos del problema
- K : número de grupos
- $\mathbf{X} = [x_n]$: datos en R^2

Variables:

- $[c_0, c_1, c_2, \dots, c_n, \dots, c_{N-1}]$: pertenencia del dato x_n al cluster C_k

Tipo de problema

- Minimización

Función de fitness:

$$F(\mathbf{X}) = \frac{1}{\sum_{k=0}^{K-1} \sum_{x_i \in C_k} \sum_{x_j \in C_k, j > i} \|x_i - x_j\|}$$

Fig 8. Parámetros del problema.

Para el desarrollo de este problema se consideró la dimensión de los puntos $M = 2$, el número de puntos $N=150$ y el número de grupos a ser clasificados $K = 3$.

Cada cromosoma (solución particular al problema dado) está conformado por un arreglo de N posiciones que indica a qué grupo K_j pertenece el punto i .

La función de fitness se toma como el inverso de la sumatoria de todas las distancias entre puntos de un mismo grupo para que convenientemente la función de fitness pudiera ser maximizada, debido a que inicialmente el problema es de minimización.

Los puntos se generaron aleatoriamente mediante tres funciones de densidad de probabilidad Gaussiana, cada una con diferente media.

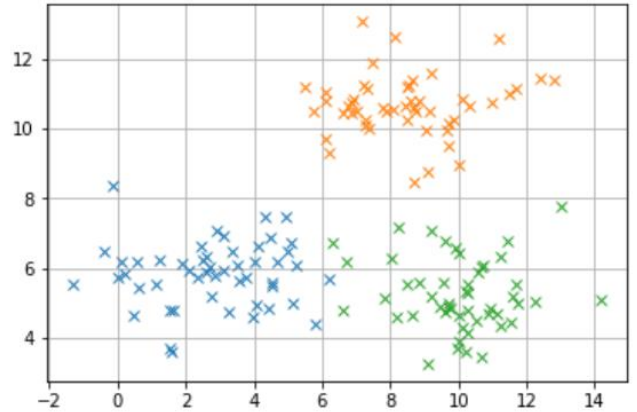


Fig 9. Puntos generados aleatoriamente

- **Política de supervivencia de la Población:**
la población en cada generación será fija, con un valor de 20 Cromosomas; se inicializa de manera aleatoria, y para cada siguiente generación sobreviven los 20 cromosomas que tengan el mejor Fitness (elitismo).
- **Selección de cruce/mutación de los padres:**
Se escoge el *método de la ruleta*, el cual se basa en calcular el fitness de cada individuo de la población, con base en éstos valores hallar una frecuencia relativa donde los mejores individuos (con mayor función de fitness) sean más propensos a ser escogidos como padres y escoger un número aleatorio (simulando el giro de una ruleta) para seleccionar cada Padre.
- **Estrategia de cruce:**
Se implementó el cruce uniforme este consiste en asignar cada gen del cromosoma hijo el gen del padre 1 con una probabilidad P o el padre 2 con una probabilidad de $(1-P)$ en este caso se tomó $P = 0.5$ por lo que ambos padres tienen la misma probabilidad de heredar cada uno de sus genes al cromosoma hijo.



Fig 10. cruce uniforme

- **Estrategia de mutación:**
Se implementó la mutación mediante *random resetting* la cual consiste en asignar un valor aleatorio del conjunto de valores permisibles a un gen escogido al azar.

Es importante anotar que se excluyó el cromosoma con mejor valor de la función de fitness de ser mutado, protegiéndolo de ser modificado.

- **Criterio de Parada:**

Iterar hasta 500 generaciones.

- **Resultados:**

Utilizando los parámetros previamente definidos y una tasa de mutación y de cruce igual a 40% se obtuvieron los siguientes resultados.

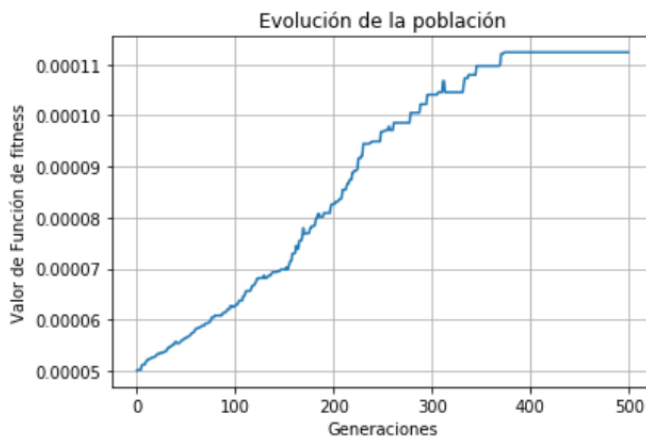


Fig 11. Evolución de la población, problema de agrupamiento

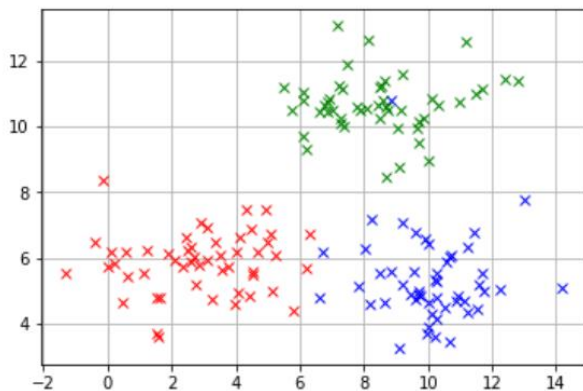


Fig 12. Solución óptima encontrada.

Se observa que la función de fitness aumentó constantemente hasta la generación 375 aproximadamente y a partir de esta se estabiliza indicando que se llegó a una solución óptima, y a simple vista se observa que el resultado obtenido es bastante bueno puesto que los grupos están bastante cercanos entre sí, e incluso desde la misma generación de los mismos usando una variable aleatoria Gaussiana, algunos puntos se mezclaban entre ellas, sin embargo se puede apreciar que el número de muestras mal clasificadas es menor a 5 por lo que el error es

aproximadamente del 0.033% es decir que se obtuvo un *Accuracy* de 99.67% aproximadamente.

III. CONCLUSIONES

Los algoritmos genéticos permiten encontrar soluciones óptimas o cercanas a ellas en un tiempo mucho inferior comparado con métodos tradicionales como la fuerza bruta, en cuyo caso hubiera sido necesario evaluar la función de fitness para cada una de las posibles soluciones particulares, lo que resultaría en un costo computacional mucho mayor puesto que a pesar de las 500 generaciones, el número de veces que se debió calcular la función de fitness es mucho menor en comparación con caso anteriormente expuesto.

Es importante resaltar la importancia de conservar las mejores propiedades de los padres de la población, para que sean heredadas por los hijos y se generen individuos con funciones de fitness mayores a las ya existentes en la población (en búsqueda del óptimo global). Además de saber que padres son sanos (con respecto a la función de fitness), como factibles (que cumplan las restricciones)

La mutación juega un papel esencial en el algoritmo genético debido a que permite mantener una variedad en la población lo que permite la exploración del espacio de soluciones y así evitar quedar estancados en un máximo local.

IV. REFERENCIAS

- [1] Isaza, Claudia; Gómez, William E.; *Notas de clase de Fundamentos de Inteligencia Computacional*, módulo de Algoritmos Genéticos, Universidad de Antioquia.
- [2] E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. SpringerVerlag, 2003.
- [3] J.H. Sánchez, J.I. Turrado. *Algoritmos genéticos: Aplicación al problema de la mochila*.: Universidad Carlos III de Madrid
- [4] LUENBERGER, David G., et al. *Linear and nonlinear programming*. Reading, MA: Addison-wesley, 1984.