

Alejandro Pardo
Alejandro Lancheros

A partir de un análisis del problema planteado en el proyecto, se consideró que se podría llegar a una solución bastante acertada con algoritmos genéticos, es por esto que se implementó la solución con base en varios aspectos; inicialmente planteamos la representación de cada individuo con un arreglo cuya longitud es el número de células presentes en el grafo, cada posición (índice) indica una célula y el valor representa el grupo del que hace parte. Posteriormente se planteó la función *fitness* la cual establece la cantidad de grupos que se pueden formar, esta función se quiere minimizar pues se requiere conocer la mínima cantidad de grupos donde cada célula dentro de dicho grupo esté conectada con cada una de las otras. Esta función discrimina a partir de la lista de adyacencias que se creó al inicio, donde si los nodos pertenecientes a un grupo común no están conectados entre sí, se penaliza sumando al *fitness* de este individuo. La función de mutación únicamente selecciona un índice (célula) al azar y lo cambia a un grupo aleatorio, esto puede ocurrir con una probabilidad del 20%, frente a 80% de la función crossover que simplemente genera más individuos a partir de elementos antiguos. También, se eligió utilizar la función de selección por torneo con elitismo, pues permite que los mejores individuos se seleccionen con mayor probabilidad y hace que los mejores individuos permanezcan de generación en generación. La función que implementa el algoritmo genético utilizando la librería *pyeasyga*, se le adaptaron componentes para resolver el problema de optimización específico, dentro de las configuraciones se pueden encontrar dos hitos importantes, *generations* (número de iteraciones (generaciones) que el algoritmo realizará) y *population size*, se decidió elegir valores acordes y adecuados a los casos, con el fin de obtener la mejor posible solución en un tiempo considerablemente aceptable.

A 3x3 grid graph with nodes labeled 1 through 9. The nodes are arranged in a 3x3 grid. The horizontal axis is labeled 0, 1, 2 and the vertical axis is labeled 0, 1, 2. The nodes are connected by edges. The nodes 1, 3, 5, 7, and 9 are highlighted in red. The edges connecting these nodes are also highlighted in red.

Individuo antes del algoritmo genético:

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Mejor individuo posible después del algoritmo genético:

1	2	1	3	2	3	4
---	---	---	---	---	---	---

2. Análisis de complejidades espaciales y temporales

En el desarrollo del algoritmo se consideraron dos puntos importantes que influyen en la complejidad espacial y temporal, uno de ellos es la creación del grafo la cual después de analizar los ciclos de alta complejidad dentro de esta función se entendió que dicha complejidad temporalmente en notación BIG O es $O(n^2)$, donde n es el número de vértices o células en el grafo. Adicionalmente, la función de fitness del algoritmo genético también tiene una complejidad temporal $O(n^2)$; también se debe considerar la cantidad de generaciones y el tamaño de la población. Por tanto la complejidad temporal de este algoritmo es:

$$O(g * p * n^2)$$

donde:

g: es el número de generaciones

p: es el tamaño de la población

n: es la cantidad de células (nodos) en el grafo.

Para la complejidad espacial se entendió que esta se maneja en función de la lista de adyacencias del grafo generado, siendo así la memoria necesaria para almacenar esta lista, en notación BIG O es $O(V + E)$.

Se debe tener en cuenta que ambas complejidades aumentan dependiendo de la cantidad de casos de prueba, es decir, si se hacen N pruebas las complejidades totales sería la multiplicación de N por la complejidad respectiva del algoritmo.

3. Respuestas a los escenarios de comprensión de problemas algorítmicos

ESCENARIO 1

i) Un nuevo desafío a considerar es permitir que dentro de cada grupo evaluado como parte de una posible solución, exista un máximo de dos células que no estén necesariamente conectadas entre sí. Esto implicaría una modificación en la función de fitness, ya que el algoritmo debería aceptar como válidas aquellas configuraciones donde hasta un par de

células no conectadas pertenezcan al mismo grupo. Esta nueva regla ampliará el espacio de búsqueda de soluciones válidas, lo que podría ralentizar la convergencia del algoritmo hacia una solución óptima.

ii) En cuanto a modificación, se realizaría cambio en la penalización de la función fitness, donde ahora se penalizará si un grupo tiene más de un par de células no conectadas, las funciones crossover y mutate seguirán funcionando de la misma manera, sin embargo, los individuos iniciales pueden generarse de diferente forma, permitiendo algunos nodos no conectados dentro de un mismo grupo.

ESCENARIO 2:

i) Como nuevo reto, se deberá definir una restricción del tamaño de los grupos asegurándose que ninguno exceda el tamaño T de células, el espacio de búsqueda aumentará, ya que el algoritmo debería considerar tanto las conexiones entre las células como el límite de tamaño de los grupos, además, minimizar el número de grupos puede generar conflicto con la limitación de tamaño, requiriendo así ajustes en la evaluación del fitness.

ii) En cuanto a los cambios en la solución, se debería modificar la función fitness para que penalice las soluciones donde un grupo tenga más de T células. También, la mutación debe evitar que un grupo crezca más allá de T células a la hora de la modificación de grupos, del mismo modo, modificar la función crossover para evitar que grupos combinados excedan el tamaño máximo T .