

Tarea semana 16: Algoritmos genéticos

Objetivo:

Implementar algoritmos genéticos para solucionar el problema del vendedor ambulante (PVA).

Descripción

El problema del vendedor ambulante es un problema típico de optimización que tiene muchas aplicaciones en distintas áreas. El problema se refiere a buscar la rutas más cortas entre un conjunto de puntos [1]. Se recomienda establecer una estrategia de resolución de problemas siguiendo los pasos del método Pólya.

1. Escriba un algoritmo genético estándar (AG) que busque el camino más corto entre un conjunto de N ciudades de la siguiente manera
 - a) **Inicialización:** Siga el mismo procedimiento indicado en el algoritmo estándar.
 - b) **Codificación de los cromosomas:** programe los cromosomas de forma que sus genes contengan permutaciones del orden de las ciudades a visitar. Es decir, para un problema de 4 ciudades los cromosomas pueden tener la forma: $[0, 3, 1, 2]$ o $[3, 1, 0, 2]$, que debe interpretarse como el orden de las ciudades a visitar. La trayectorias son cerradas, es decir, al final del camino se debe regresar al punto de partida, pero el nodo inicial no debe repetirse en el cromosoma. Además, su rutina debe generar secuencias de cromosomas donde no se repitan ciudades a visitar.
 - c) **Función de ajuste:** el valor de ajuste debe tomarse como el inverso de la longitud euclídeana de la trayectoria seguida.
 - d) **Operadores evolucionarios:** deben modificarse para esta implementación de un AG. En este caso **no** se va a usar un operador de cruce ni un método de selección.
 - e) **Operador mutación:** debe implementarse como un intercambio entre las posiciones de dos genes del cromosoma. Es decir, si el cromosoma $[1, 3, 2, 0]$ se selecciona para mutación, deben elegirse dos genes aleatorios del cromosoma, por ejemplo el g_0 y el g_2 . El gen mutado resultante en este caso es $[2, 3, 1, 0]$. Todos los individuos de la población deben sufrir mutaciones con probabilidad p_{mut} .

- f) Use su programa para encontrar el camino más corto entre ciudades cuyas coordenadas (x_i, y_i) se generan mediante las siguientes funciones:

$$x_i = 0,1[(9 + 13i^2) \bmod 200] \quad (1)$$

$$y_i = 0,1[(7 + 1327i) \bmod 200] \quad (2)$$

donde $i = 1, \dots, 100$. Es decir, se generan coordenadas para un conjunto de 100 ciudades.

- g) Su rutina debe generar un gráfico que muestre las posiciones de las ciudades y la trayectoria más corta encontrada al final de la simulación. Opcionalmente puede generar una animación que se actualice cada vez que el algoritmo encuentra un camino más corto.
- h) Genere un gráfico que muestre los valores de ajuste promedio de la población y el ajuste del individuo mejor adaptado por cada generación de la simulación. Comente acerca del comportamiento de los valores de ajuste durante la simulación. Además mencione los parámetros utilizados en la simulación y su posible influencia en el comportamiento observado de los valores de ajuste.
2. Una población de individuos completamente aleatorios pone al AG en desventaja desde el inicio en este problema. Genere una versión modificada del AG (AGM), e investigue su rendimiento frente a la implementación estándar:
- a) Modifique la función de inicialización del AG para que genere la población inicial de la siguiente manera: para cada individuo se selecciona un punto de partida aleatorio, a partir del nodo seleccionado se forma trayecto donde el nodo siguiente es siempre el más cercano.
- b) **Operador de mutación:** Aplique el operador mutación a esta población inicial de la siguiente manera: deje el primer individuo sin cambios, pero aplique de 3 a 10 mutaciones al resto de la población.
- c) Una vez que la población inicial ha sido definida, ejecute el AGM usando los operadores descritos en la parte anterior, hasta obtener un resultado satisfactorio.
- d) Use su AGM para encontrar el camino más corto entre el conjunto de ciudades cuyas coordenadas fueron generadas mediante las funciones (1) y (2).
- e) Su rutina debe generar un gráfico que muestre las posiciones de las ciudades y la trayectoria más corta encontrada al final de la simulación. Opcionalmente puede generar una animación que se actualice cada vez que el algoritmo encuentra un camino más corto.
- f) Genere un gráfico que muestre los valores de ajuste promedio de la población y el ajuste del individuo mejor adaptado por cada generación de la simulación. Comente acerca del comportamiento de los valores de ajuste durante la simulación. Además mencione los parámetros utilizados en la simulación y su posible influencia en el comportamiento observado de los valores de ajuste.

- g) Basado en los gráficos de evolución de los valores de ajuste, los tiempos de ejecución y los parametros seleccionados para cada implementación de algoritmo genético estandar; compare el rendimiento del AG contra el rendimiento del AGM.

Entrega:

1. Fecha de entrega: 15 de junio de 2022.
2. Se puede presentar de manera individual o en parejas.
3. La entrega se tecDigital en el rubro TS16.
4. Debe entregarse un reporte de resultados que contenga:
 - enlace al código usado para generar las simulaciones,
 - una sección de resultados con las gráficas obtenidas de forma ordenada,
 - y una sección discusión donde comente los resultados obtenidos y haga sus conclusiones finales.
5. El código debe estar alojado en un repositorio de acceso público en sitios web como `github.com` o `gitlab.com`.
6. Debe tomar en cuenta las buenas prácticas de programación a la hora de escribir su código. Principalmente las buenas prácticas respecto a
 - la asignación de los nombres adecuados a las variables, arreglos y funciones,
 - uso de funciones para realizar tareas repetitivas,
 - uso de comentarios cuando sean necesarios,
 - y al uso de expresiones sencillas y legibles.
7. La calificación de la tarea se hará basada en una rúbrica de evaluación que se hará disponible en tecDigital.

Referencias

- [1] M. Wahde, *Biologically Inspired Optimization Methods: An Introduction*. WIT Press, 1 ed., 2008.