Modelación y Simulación 2024

Lab 08

17.octubre.2024

1. **(Problema Knapsack)**
   Considere el problema Knapsack general: se tiene un conjunto de n objetos, con valores o recompensas $\mathbf{v} = (v_1, v_2, \ldots, v_n)$, y pesos $\mathbf{w} = (w_1, w_2, \ldots, w_n)$, respectivamente. Todos los valores y pesos son no-negativos, $v_i \geq 0$ y $w_i \geq 0$, para todo $1 \leq i \leq n$. Suponga que se desea elegir objetos para llevar, de forma que el peso total de los objetos elegidos no sobrepasa la capacidad $K \geq 0$ de la mochila. El objetivo es determinar la selección óptima de objetos que maximiza la recompensa total, sujeto a la restricción de capacidad.

   Diseñar un algoritmo genético para resolver un problema Knapsack cualquiera. Su algoritmo debe recibir como argumentos la lista $\mathbf{v}$ de valores de cada objeto, la lista $\mathbf{w}$ de pesos de cada objeto, la capacidad K de la mochila. Debe recibir también los parámetros específicos de su algoritmos genético. Como resultado, el algoritmo debe devolver la lista de objetos seleccionados para llevar, el valor de la recompensa total, y el peso total de la selección óptima.

   Usted es libre de elegir la estructura interna de su algoritmo genético. Puede elegir el diseño de su operador de selección, de sus operadores de cruce y de mutación. Considere también libre elegir los parámetros internos de su algorito, por ejemplo:

   - $N =$ tamaño de la población,
   - $s =$ % de selección o sobrevivencia,
   - $c =$ % de nuevos individuos originados por cruce,
   - $m =$ % de nuevos individuos originados por mutación,
   - $F =$ % función de *fitness*,
   - operadores de cruce y de mutación,
   - $maxI =$ numero máximo de iteraciones, u otros criterios de paro,
   - ...

2. **(Knapsack)**
   Para los siguientes datos: $K = 50$ y

Item	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Valor	10	12	8	5	8	5	6	7	6	12	8	8	10	9	8	3	7	8	5	6
Peso	6	7	7	3	5	2	4	5	3	9	8	7	8	6	5	2	3	5	4	6

   (a) Formular el problema como un problema de programación lineal, y resolverlo mediante las librerías JuMP (Julia) o Pulp (Python).
   Hallar la selección óptima y encontrar el valor total de recompensa y el peso total de la selección.

   (b) Resolver el problema mediante su algoritmo genético del Ejercicio 1. Hallar la selección óptima y encontrar el valor total de recompensa y el peso total de la selección. Comparar esta solución con la obtenida mediante programación lineal.
   ¿Es la misma solución? ¿Es mejor o peor?

3. **(TSP)**
Dado un grafo G de N vértices, un **tour** en G es un recorrido que comienza en algún vértice V, visita todos los nodos exactamente una vez (sin repetir), y regresa de nuevo al nodo V (es un trayecto cerrado que visita todos los nodos exactamente una vez).

El problema del viajero (*Traveling Salesman Problem*) o TSP, consiste en dado un grafo G, encontrar el tour que minimiza la distancia total de recorrido, y que visita todas las ciudades exactamente una vez.

Resolver el problema TSP con los datos del problema **ch150.tsp.gz** disponibles en el sitio `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/`.

Para ello, diseñar específicamente un algoritmo genético para el TSP. El algoritmo debe recibir como argumentos la lista de coordenadas de los nodos de G, y los parámetros específicos de su algoritmos genético. Como resultado, el algoritmo debe devolver el tour óptimo encontrado, esto es, la lista ordenada en la que se visitan las ciudades y la distancia total recorrida en dicho tour.

Mostrar en un video o archivo GIF, una animación 2D de los tours encontrados en cada iteración de su algoritmo genético (elija una muestra de iteraciones, no todas), pero asegúrese de mostrar al final la solución *best* obtenida por el algoritmo genético. Dejar constancia de la figura 2D de la solución óptima encontrada por su algoritmo.

Puede comparar su resultado con la solución óptima indicada en el archivo **ch150.opt.tour.gz**.

Dejar constancia de la salida del algoritmo para el experimento en donde se halló la mejor solución óptima.

Nota: El equipo que encuentre la mejor solución subóptima será exonerado del Examen Corto 2.