Práctica 1.

Codificación y evaluación de soluciones

Metaheurísticas – Grado de Ingeniería Informática

Universidad de Córdoba

2015 / 2016

El objetivo de esta práctica es el de introducir al alumnado en los primeros pasos para la resolución de problemas NP mediante metaheurísticas. Para ello, se presentarán dos problemas clásicos, el problema de la mochila (KP) y el del viajante de comercio (TSP), que comúnmente se abordan con codificaciones sencillas. Las tareas a realizar consistirán en la programación clases y métodos que puedan leer instancias de estos problemas desde un archivo, generar soluciones aleatorias y evaluarlas.

1 Descripción de los problemas

1.1 El problema de la mochila

Supongamos que tenemos n objetos $\{o_1,...,o_n\}$. Cada objeto tiene un beneficio p_i y un peso w_i, asociados, no negativos. Por otro lado, se tiene una mochila donde se pueden introducir los objetos, que soporta un peso máximo W. El problema consiste en meter en la mochila la combinación de objetos que maximice la suma de sus beneficios y que no supere el peso máximo que puede soportar la misma. Su formulación matemática es:

$$\max \sum_{i=1}^{n} p_i \cdot x_i$$

sujeto a:

$$\sum_{i=1}^{n} w_i \cdot x_i \leq W$$
$$x_i \in \{0,1\}$$

1.2 El problema del viajante de comercio

Dado un grafo no dirigido con pesos $G=(V, E, w)$, $V=\{v_1,...,v_n\}$ son sus nodos, $E=\{e_{ij}=(v_i,v_j) \mid v_i,v_j \in V\}$ sus arcos y $w: E \rightarrow R$ una función que asigna un peso a los arcos del grafo,

el objetivo es encontrar el camino que recorre todos los nodos del grafo y cuya suma de pesos es mínimo.

2 Tareas a realizar

Se debe:

1. Descargar e inspeccionar los ficheros de instancias de los problemas disponibles en la página de la asignatura.
2. Implementar una clase `Instance<nombreProblema>` capaz de leer la información contenida en un fichero instancia del problema.
3. Implementar una clase `Solution<nombreProblema>` que represente una solución candidata al problema.
4. Implementar una clase `SolGenerator<nombreProblema>` que pueda generar soluciones aleatorias al problema correspondiente (objetos de la clase anterior, configurados según la solución generada).
5. Implementar un método en la clase `Instance<nombreProblema>` que recibiendo una solución al problema devuelva su valor de aptitud.
6. Implementar un programa para cada problema que, recibiendo un fichero instancia del problema, genere 1000 soluciones aleatorias e imprima:
   a. El mejor valor de aptitud encontrado hasta el momento, en cada iteración
   b. La configuración de la mejor solución, al final
7. Desarrollar los primeros pasos para la codificación y evaluación de soluciones del problema seleccionado, diferente al KP y TSP.
8. Elaborar un informe con la tarea realizada, resultados obtenidos y análisis. Se sugiere presentar algunas tablas con los resultados obtenidos en las diferentes instancias y gráficas de convergencia de las ejecuciones.
9. Subir el informe a moodle **JUNTO CON la evaluación del mismo por otros dos grupos**, quienes muy brevemente indicarán los puntos fuertes y débiles del trabajo realizado y le asignarán una calificación numérica entre 1 y 10.