Práctica 3: El problema del Viajante

Profesor Responsable: Jorge San Martín Corujo*

23 de Mayo de 2006

Las condiciones del proyecto son las siguientes:

- El proyecto de programación que se presenta a continuación se realizará en grupos de dos o tres alumnos (es decir: ni 1 ni 4).
- El lenguaje de programación se deja a la elección del alumno.
- Las herramientas para poder compilar, instalar y ejecutar el programa deben estar disponibles de manera gratuita bajo un entorno GNU Linux.
- Se entregará un disco o CD con el código fuente, así como una documentación del programa y su código fuente en soporte papel. Además, deberá contener instrucciones claras y sencillas para poder instalar, compilar y ejecutar el programa.
- Es imprescindible que el código esté bien comentado (se recomiendan herramientas automáticas tipo `doxygen` o `javadoc`)
- Se podrá entrevistar a los integrantes de algunos de los grupos acerca de su solución.
- La fecha de entrega del proyecto será el 31 de Agosto de 2006 (jueves), a lo largo de la mañana, en el Edificio Departamental II, despacho 52.
- Para cualquier duda, se puede contactar con el profesor a cargo de la práctica (tanto por correo como en persona).

*jorge.sanmartin@urjc.es

1. Librería CSP

La primera parte de la práctica se basa en la creación de una librería para resolver Problemas de Satisfacción de Restricciones (CSP). Dicha librería debe contener, al menos, todos los algoritmos que se han visto en clase (vuelta atrás cronológica,vuelta atrás cronológica con comprobación hacia adelante y algoritmo MAC), al igual que las diferentes heurísticas genéricas posibles (al menos mínimo valor restante y grado heurístico).

Esta librería debe ser todo lo genérica posible, de manera que pueda atacar diferentes problemas (por ejemplo resolver las n–reinas, el coloreo de mapas, etc).

2. El problema del viajante

2.1. Definición

El problema del viajante consiste en recorrer todo un grafo de manera que no se repita ningún nodo. Es decir, comenzando por un nodo arbitrario, debo conseguir visitar (sin repetición) todos los nodos y volver al origen. Matemáticamente hablando, dicho recorrido se llama ciclo Hamiltoniano de un grafo.

Más formalmente, dado un grafo G con nodos $\{a_1, \ldots, a_n\}$, se trata de encontrar una permutación de nodos σ tal que:

- $\sigma(a_i)$ es adyacente a $\sigma(a_{i+1})$ para $i = 1, \ldots, n-1$
- $\sigma(a_n)$ es adyacente a $\sigma(a_1)$.

La solución a tal problema es la lista ordenada de nodos $[\sigma(a_1), \ldots, \sigma(a_n)]$

2.2. Resolución

Utilizando la librería genérica para problemas de satisfacción de restricciones, se debe programar el problema del viajante. El programa debe tener un interface por consola o por fichero (un interface gráfico alternativo es opcional y se valorará positivamente) con al menos las siguientes opciones:

1. Se le pregunta al usuario el nombre del fichero que contiene el grafo con las ciudades.
2. El programa debe dar la opción para elegir, al menos, los distintos algoritmos de resolución y la aplicación de alguna heurística.

3. El ordenador debe devolver por pantalla una solución del problema (o bien decir que no existe solución).

4. Se debe incluir una colección de ficheros con distintos grafos para poder experimentar.

Evidentemente, se debe de programar de alguna manera la estructura de datos grafo.

3. Informe

Haga un informe con sus experiencias haciendo la práctica, qué le ha resultado más difícil de implementar, qué parte ha sido más fácil, cuál le ha gustado más... y un breve informe técnico, donde presente qué algoritmos funcionan mejor para el problema propuesto, las mejores heurísticas, e intente justificar de alguna manera esos comportamientos prácticos.