

Prácticas de Búsqueda Heurística. Curso 2023-2024.

Sesión 2. El TSP con A*.

Ramiro Varela, Francisco J. Gil-Gala. Irene Díaz. Juan Luis Mateo. Sistemas Inteligentes.
Grado en Ingeniería Informática. EII. Universidad de Oviedo. Campus de los Catalanes. Oviedo

Resumen. En esta sesión se trata de implementar una solución para el TSP en `aima-python`. Para ello, hay que definir las clases necesarias, `ProblemaTSP` y quizá `EstadoTSP`, y luego definir e implementar heurísticos apropiados.

Palabras claves: TSP, A*, heurísticos

1 Introducción

En esta práctica vamos a resolver el problema del TSP con el algoritmo A*, y a experimentar con el prototipo desarrollado sobre instancias del repositorio TSPLIB. La componente mas importante será la implementación de heurísticos y su comparación.

2 Ejercicios para esta práctica

Partimos de la implementación que ya se incluye en el notebook `search4e-Busqueda_2023_2024.ipynb`. Esta implementación no incluye heurísticos, salvo alguno muy simple y muy poco informado. El objetivo de la práctica es implementar varios de los heurísticos vistos en teoría, con distintos costes computacionales y grado de información, y después hacer algunos experimentos para comparar los resultados de estos heurísticos. En principio, tenemos incorporados los datos de tres instancias, la de las diapositivas de 6 ciudades y dos instancias del repositorio TSPLIB: `gr17` y `gr21`, los dos con formato `EDGE_WEIGHT_TYPE`.

Ejercicio 1. En primer lugar, se trata de visualizar y entender la implementación de las clases `EstadoTSP`, `Grafo` y `SymmetricTSP`, y de hacer algunos experimentos preliminares con instancias pequeñas (dados los datos de una instancia en formato `EDGE_WEIGHT_TYPE`, si eliminamos una parte de las filas finales, tenemos otra instancia de tamaño menor). Trataremos de identificar el mayor tamaño de las instancias que se pueden resolver con heurísticos muy poco informados, como el `h0`.

Ejercicio 2. Ahora se trata de definir algunos heurísticos: `h1` (que para cada ciudad que queda por abandonar considera el coste del arco mínimo que toca a esa ciudad) y otros (`h2`, `h3` y `h_MSP`) que se basan en distintas relajaciones del problema y que se especifican en el código.

Ejercicio 3. Por último, se trata de comparar el heurístico h_MST con otro que tenga menor coste computacional, como por ejemplo h_2 . Hay que comprobar si la reducción del número de nodos expandidos con h_MST compensa el exceso en tiempo de evaluación de h_MST con respecto a h_2 . Aquí es importante el tiempo de ejecución.

Bibliografía

1. Russell, S. and Norvig P. Artificial Intelligence, A Modern Approach, Prentice Hall, New Jersey, 4th ed. 2021.
2. Nilsson, N., Principles of Artificial Intelligence, Tioga, Palo Alto, CA, 1980.
3. Pearl, J., Heuristics, Morgan Kaufman, San Francisco, CA, 1983.