31-5-2020

Néstor Monzón González - 735418

Andrés Otero García - 757755

PRÁCTICA 2

Ramificación y poda

Algoritmia Básica

1 – Diseño e implementación

El algoritmo de ramificación y poda podría considerarse una variante del backtracking mejorada, ya que, gracias a la poda, cuando se encuentra la primera solución óptima, se asigna su coste como el límite. En la mayoría de los casos, esto permite descartar muchos caminos sin necesidad de expandirlos. Por ejemplo, si se obtiene que una solución óptima tiene un coste de 20, cualquier nodo cuyo coste inicial sea mayor que 20 se descarta (se desactiva). Así, el coste final en tiempo es mucho menor, ya que no es necesario realizar la búsqueda en buena parte de las ramificaciones de un problema dado.

En esta práctica, se ha implementado este algoritmo para aplicarlo a un problema de recorrido de nodos en un espacio bidimensional. Concretamente, contamos con un cierto número de casillas, una posición inicial y una serie de minas repartidas en estas. El objetivo es encontrar uno de los caminos óptimos (más cortos) que pase por todas las minas volviendo a la posición inicial.

A la hora de implementarlo, se decidió utilizar el lenguaje Python, ya que su biblioteca numpy, siendo de alto nivel, está muy optimizada para los cálculos matriciales que cimientan este método (se deben mantener en todo momento una serie de matrices con los costes entre cada mina a visitar).

Desde el punto de vista software, el código se ha dividido en 4 ficheros principales:

* : programa principal. Se encarga, por una parte, de leer y decodificar el fichero de entrada (con la sintaxis descrita en el enunciado), asignando a nuestras estructuras de datos internas los valores del mismo. En segundo lugar, para cada escenario del fichero, llama a la función , que encuentra la solución a un escenario con el método descrito.
* define la clase Escenario con sus correspondientes atributos y métodos. Tipifica el conjunto de parámetros que definen un problema completo: tamaño del espacio, número de minas, posición inicial y posición de cada mina.
* : contiene la función que realiza el algoritmo de poda y ramificación para un escenario, junto con sus distintas funciones auxiliares.
* : define la clase Nodo, con sus datos y