Memoria Práctica 3: Partición de palabras

Alonso del Rincón de la Villa y Alberto Lardiés Getán

24 de mayo de 2023

Índice

1. Diseño

Nos encontramos con un problema de búsqueda en el que debemos encontrar el conjunto de pedidos que maximicen el beneficio total. Las acciones posibles en cada estado es añadir uno de los pedidos restantes (no añadidos) a la solución y cada nodo del espacio de soluciones (estados del problema) consiste en el conjunto de los pedidos aceptados hasta ese momento. Esto constituye un espacio de soluciones de 2^n siendo n el número de pedidos, es decir, todos los subconjuntos del conjunto de los pedidos.

La función de coste será la siguiente:

- Para un nodo (conjunto de pedidos) que viola la restricción de capacidad del tren en algún tramo del recorrido tomará valor infinito.
- Para un nodo cuyos hijos violen la restricción tomará como valor el precio agragado de los pedidos no aceptados. Si este valor es mínimo, el precio agregado de los pedidos aceptados, la magnitud que se pide maximizar, será máximo.
- Para un nodo con hijos que no violen la restricción tomará como valor el mínimo coste de entre sus hijos. De esta forma los nodos desde la raiz a la solución óptima tendrán coste mínimo.

Nos enconramos con un problema muy similar al problema de planificación de tareas a plazo fijo visto en las transparencias. Su función de estimación es también válida aquí, es decir, para cualquier nodo del espacio de soluciones es menor que el valor de la función de coste para ese mismo nodo. La función de estimación en custión:

- toma valor infinito para un nodo que viola la restricción de capacidad del tren en algún tramo del recorrido y
- para un nodo cuyos hijos violen la restricción toma valor igual a la suma de los precios de los pedidos no añadidos de índice (determinado por el orden en el fichero de entrada) menor al máximo de los pedidos añadidos.

De la misma forma podemos adaptar su función de cota siendo la nuestra igual, para cada nodo, a la suma de los precios de los pedidos no añadidos.

2. Pruebas

2.1. Diseño de las pruebas

Para dar sentido a las mediciones temporales del algoritmo hemos implementado una solución al problema mediante fuerza bruta. Mediante una búsqueda en anchura intentará encontrar la solución. Esperamos un mejor rendimiento por parte del algoritmo de ramificación y poda, significando esto que la función de estimación ha guiado la búsqueda hacia una buena solución y ha podado ramas que de seguro no llevaban a la óptima.

Para una comparación justa con la solución de fuerza bruta generaremos aleatoriamente varios casos de prueba.

Para comprobar la corrección el generador de pruebas añade al fichero pruebas.txt las pruebas del enunciado como los dos primeros bloques.

2.2. Resultados y conclusiones

Los resultados para los 5 casos de prueba (los 2 del enunciado y 3 generados aleatoriamente) son los siguientes:

Última estación	n pedidos	Fuerza Bruta (ingreso)	Poda (ingreso)
3	5	18	18
5	6	38	38
2	12	-	27
4	18	-	36
4	19	-	20

Última estación	n pedidos	Fuerza Bruta (ms)	Poda (ms)
3	5	1.0493	0.3165
5	6	6.6742	0.1943
2	12	-	0.9574
4	18	-	50.0551
4	19	-	4.2303

Los ingresos de los dos primeros casos de prueba confirman la corrección de nuestros algoritmos y la elección de una heurística admisible y una función de cota correcta.

La mejora de coste temporal es notable con el método de ramificación y poda, significando que la función de estimación guia bien al algoritmo hacia la solución y/o que la función de cota está podando las ramas imposibles.

Para los 3 casos de prueba generados el algoritmo de fuerza bruta agota la memoria del proceso. El espacio de soluciones distintas no es tan grande como para que esto se de, pero la cola FIFO implementada admite estados repetidos.

Pese a ser el quinto caso más grande en terminos de pedidos su coste es mucho menor debido a que aunque un problema tenga un espacio de soluciones mayor, un algoritmo de ramificación y poda puede llevar a la solución antes que en el caso de un espacio de soluciones menor.