UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Algorítmica (11593)-Seguimiento-19 de diciembre de 2017

NOMBRE:

1 — 5 puntos

Una empresa de exportaciones-importaciones tiene acumulados en su almacén L lotes de productos. Cada lote i, para $1 \leq i \leq L$, está compuesto de n_i productos idénticos. Cada uno de los j productos de cada lote i (identificado por $1 \leq j \leq n_i$) tiene asociado su valor de compra c_i y su peso p_i en kilos. Necesitamos hacer sitio en nuestro almacén pues va a llegar más mercancia próximanente. Para ello, la empresa ha alquilado un contenedor marítimo con una capacidad de K kilos para transportar productos a otra sede. La empresa necesita seleccionar cuántos productos de cada lote debe transportar, sabiendo que puede vender en la otra sede cada producto del lote i a un precio v_i. Evidentemente, la empresa desea maximizar el beneficio.

Una posible instancia del problema sería: un contenedor marítimo con una capacidad de 100 toneladas (100×10^3 kilogramos), 5 lotes de productos, con las siguientes características:

	lote 1	lote 2	lote 3	lote 4	lote 5
n_i	1000	200	100	600	400
p_i	100	500	80	10	200
c_i	20	100	120	10	100
v_i	30	200	200	15	220

Diseña un algoritmo de Ramificación y Poda que obtenga la mejor selección de productos a transportar. Para ello:

a) Formaliza el problema en términos de optimización:

1) Identifica y expresa el conjunto de soluciones factibles X.
2) Identifica y expresa la función objetivo a optimizar f.
3) Identifica y expresa la solución óptima buscada $\hat{x}$.
4) ¿Hay alguna condición para que, dada una instancia del problema, exista al menos una solución factible? Si es así, indica cuál. Pon tres ejemplos de soluciones factibles (si es que existen) para la instancia presentada.

b) Describe los siguientes conceptos sobre los estados que serán necesarios para el algoritmo.

1) Representación de un estado (no terminal) y su coste espacial. Pon un ejemplo para la instancia presentada.
2) Condición para que un estado sea solución. Pon un ejemplo para la instancia presentada.
3) Identifica el estado inicial que representa todo el conjunto de soluciones factibles.

c) Define una función de ramificación. Analiza su coste temporal. Pon un ejemplo para la instancia presentada.

d) Diseña una cota superior no trivial. Estudia cómo se puede realizar el cálculo de la cota superior de forma eficiente. ¿Cuál sería su coste temporal? Analízalo.

(MOCHILA DISCRETA CON REPETICIÓN)

	lote 1	lote 2	lote 3	lote 4	lote 5
n_i	1000	200	100	600	400
p_i	100	500	80	10	200
c_i	20	100	120	10	100
v_i	30	200	200	15	220
c_i	20	100	120	10	100
g_i	10	100	80	5	120

$$X = \{(x_1, x_2, ..., x_L) \mid 0 \le x_i \le n_i, 1 \le i \le L, \sum_{1\le i\le L} x_i p_i \le K\}$$

$$f((x_1, x_2, ..., x_L)) = \sum_{1\le i\le L} x_i(v_i - c_i)$$

$$\hat{x} = \operatorname*{argmax}_{x\in X} \sum_{1\le i\le L} x_i(v_i - c_i)$$

Ejemplos de soluciones para la instancia presentada: (1000,0,0,0,0), (100,100,100,500,200), (0,100,0,0,400).

Un estado se representará con una secuencia incompleta $(x_1, x_2, ..., x_k)$, con $k < L$. Tendrá un coste espacial de $O(L)$. Un ejemplo: (100, 100, ?).

Para que un estado sea solución la tupla deberá tener una talla igual a L. Un ejemplo: (100,100,100,500,200).

El estado inicial se puede representar como una tupla vacía (?).

$$branch(x_1, x_2, ..., x_k, ?) = \{(x_1, x_2, ..., x_k, x_{k+1}, ?) \mid 0 \le x_{k+1} \le n_{k+1} \mid \sum_{1\le i\le k+1} x_i p_i \le K\}$$

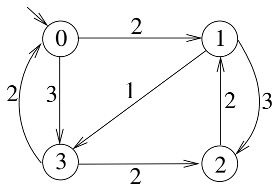
$$branch(x_1, x_2, ..., x_k, ?) = \{(x_1, x_2, ..., x_k, x_{k+1}, ?) \mid 0 \le x_{k+1} \le 0 \le x_{k+1} \le min(n_{k+1}, \lfloor (K - \sum_{1\le i\le k} x_i p_i)/p_{k+1} \rfloor)\}$$

En este caso una cota puede ser asumir que el espacio que queda libre en el contenedor se puede rellenar (completamente) con todas las unidades que se deseen del mejor lote que quede por considerar. Se podrían ordenar los lotes por beneficio unitario máximo al principio y de esta forma se rellenaría con productos del siguiente lote (que sería el mejor de los que quedan por considerar). Se puede hacer forma incremental con coste constante.

Dado un grafo $G = (V, E)$ dirigido y ponderado positivo por la función $d : E \rightarrow \mathbf{R}^{>0}$, haz una traza de un algoritmo de Ramificación y Poda que encuentre el camino de mínimo coste que:

a) comience en el vértice inicial y que
b) pase por todas las aristas del grafo al menos una vez.

En el grafo del ejemplo, siendo el vértice 0 el inicial, el camino (0, 3, 2, 1, 2, 1, 3, 0, 1), de coste 17 es una solución (no necesariamente la óptima).



Ten en cuenta:

a) Sigue una estrategia por primero el mejor (en caso de empate, el estado más cercano a una solución) y con poda implícita.

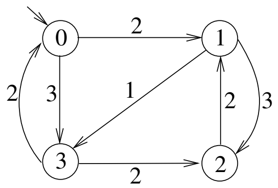
b) Describe la cota optimista que vas a utilizar y analiza su coste temporal.

c) Lista 3 ejemplos de soluciones factibles (si es que existen), y su valor de función objetivo.

d) Si para la traza utilizas el esquema que inicializa la variable mejor solución $\hat{x}$ a una solución factible o a una cota pesimista, describe qué algoritmo o método utilizas para calcularla y cuál es su coste temporal.

e) En cada iteración de la traza, deberás indicar:

- El conjunto de estados activos (aunque en realidad esté representado con un min-heap, represéntalo como un conjunto desordenado, teniendo en cuenta que siempre estará accesible el estado con menor puntuación).
- Marca el estado seleccionado para explorar.



Solución: El conjunto de soluciones factibles o espacio de búsqueda serán todos los posibles caminos que cumplan los requisitos, esto es, que pasen por todas las aristas y que comiencen en el vértice inicial (suponemos que es el 0):

$$X = \left\{ (x_1, x_2, \ldots, x_n) \mid x_1 = 0, (x_i, x_{i+1}) \in E, 1 \le i < n, \bigcup_{1 \le i < n} \{(x_i, x_{i+1})\} = E \right\}$$

Nuestra función objetivo es minimizar

$$f((x_1, x_2, \ldots, x_n)) = \sum_{1 \le i < n} d(x_i, x_{i+1}).$$

Y deseamos calcular

$$(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) = \arg \min_{(x_1, x_2, \ldots, x_n) \in X} \sum_{1 \le i < n} d(x_i, x_{i+1}).$$

Un estado se puede representar como una secuencia (camino) incompleta: $(x_1, x_2, \ldots, x_k, ?)$. El estado inicial será la secuencia incompleta $(0, ?)$, pues los caminos deben comenzar por el vértice 0. Cuando se haya pasado por todas las aristas, es decir, cuando $\bigcup_{1 \le i < n} \{(x_i, x_{i+1})\} = E$, la secuencia será un estado solución. Definido así el estado, la función *branch* para un estado ramificable será:

$$\mathit{branch}((x_1, x_2, \ldots, x_k, ?)) = \{(x_1, x_2, \ldots, x_k, x_{k+1}, ?) \mid (x_k, x_{k+1}) \in E\}.$$

Una posible cota optimista para este problema se puede obtener sumando a la distancia del camino ya recorrido, la suma del peso de las aristas no visitadas:

$$F((x_1, x_2, \ldots, x_k, ?)) = \sum_{1 \le i < k} d(x_i, x_{i+1}) + \sum_{(m,n) \in E, (m,n) \notin \bigcup_{1 \le i < k} \{(x_i, x_{i+1})\}} d(m, n),$$

que se puede obtener en tiempo constante si se calcula de forma incremental:

$$F((x_1, x_2, \ldots, x_k, x_{k+1}, ?)) = \begin{cases} F((x_1, x_2, \ldots, x_k, ?)) + d(x_k, x_{k+1}), & \text{si } (x_k, x_{k+1}) \in \bigcup_{1 \le i < k} \{(x_i, x_{i+1})\}; \\ F((x_1, x_2, \ldots, x_k, ?)), & \text{en otro caso.} \end{cases}$$

Para comprobar eficientemente la condición de si una nueva arista (x_k, x_{k+1}) ya estaba visitada en un estado $(x_1, x_2, \ldots, x_k, ?)$, esto es, si $(x_k, x_{k+1}) \in \bigcup_{1 \le i < k} \{(x_i, x_{i+1})\}$ será necesario un vector de booleanos indexado por las aristas que marque, para cada estado, las aristas ya visitadas. También será necesario este vector de booleanos para comprobar eficientemente si el estado es o no solución.

$A_0 = \{(0?, 15)*\}$
$A_1 = \{(01?, 15)*, (03?, 15)\}$
$A_2 = \{(013?, 15)*, (012?, 15), (03?, 15)\}$
$A_3 = \{(0130?, 15)*, (0132?, 15), (012?, 15), (03?, 15)\}$
$A_4 = \{(01301?, 17), (01303?, 15)*, (0132?, 15), (012?, 15), (03?, 15)\}$
$A_5 = \{(013030?, 17), (013032?, 15)*, (01301?, 17), (0132?, 15), (012?, 15), (03?, 15)\}$
$A_6 = \{(0130321, 15)*, (013030?, 17), (01301?, 17), (0132?, 15), (012?, 15), (03?, 15)\}$

Se obtiene la solución (01303212) con $fx = 15$. Como el mejor de todos los estados tiene una cota de 15, el algoritmo termina.