

Algorítmica (11593) VA, Voraces y RP-13/enero/2020 Superior de Ingeniería



Escuela Técnica

NOMBRE:	NÚM.:
1	3.5 puntos

Hay N tareas a ejecutar (numeradas de 1 a N) y M procesadores idénticos (numerados de 1 a M), con $M \leq N$, que trabajan en paralelo. El tiempo necesario para procesar la tarea i-ésima en cualquiera de los procesadores es t_i . Teniendo en cuenta que el orden en el que se ejecuten las tareas dentro de un mismo procesador no es significativo, diseña un algoritmo de Ramificación y Poda que determine qué tareas se deben ejecutar en cada procesador de forma que se minimice el tiempo de finalización de la tarea que termine más tarde.

- a) Expresa el problema en términos de optimización: expresa formalmente el conjunto de soluciones factibles y la función objetivo a minimizar. Explica brevemente cómo expresas una solución x del conjunto X. Pon un ejemplo de solución factible para una instancia con 5 tareas y 3 procesadores.
- b) Describe los siguientes conceptos sobre los estados que serán necesarios para el algoritmo:
 - 1) Representación de un estado (no terminal). Pon un ejemplo para la instancia con 5 tareas y 3 procesadores.
 - 2) Condición para que un estado sea solución. Pon un ejemplo para la instancia con 5 tareas y 3 procesadores.
 - 3) Identifica el estado inicial que representa todo el conjunto de soluciones factibles.
- c) Define una función de ramificación. Contesta a las siguientes cuestiones:
 - 1) Explica la función.
 - 2) Define la función (en python o en lenguaje matemático).
 - 3) Pon un ejemplo para una instancia con 10 tareas y 5 procesadores.
- d) Diseña una cota optimista no trivial. Contesta a las siguientes cuestiones:
 - 1) Explica la cota (caso general, de un estado intermedio).
 - 2) Explica el cálculo de la cota del estado inicial.
 - 3) Define la cota (en python o en lenguaje matemático).

2 1.5 punto

Podemos mejorar el algoritmo anterior incializando la variable "mejor solución vista hasta el momento" a una solución factible arbitraria. Pero sería mejor usar una estrategia voraz en lugar de usar una cualquiera. Para ello, vuelve al planteamiento original:

"Hay N tareas a ejecutar (numeradas de 1 a N) y M procesadores idénticos (numerados de 1 a M), con $M \leq N$, que trabajan en paralelo. El tiempo necesario para procesar la tarea i-ésima en cualquiera de los procesadores es t_i . Teniendo en cuenta que el orden en el que se ejecuten las tareas dentro de un mismo procesador no es significativo, diseña un algoritmo Voraz que determine qué tareas se deben ejecutar en cada procesador de forma que se intente minimizar en lo posible el tiempo de finalización de la tarea que termine más tarde."

- 1. Explica en lenguaje natural la estrategia voraz a seguir.
- 2. Indica el coste del algoritmo a desarrollar.
- 3. Escribe una función Python que reciba la lista de los tiempos de las N tareas y el número de procesadores M y que devuelva una lista con la asignación de tareas a procesadores.

3 2.5 puntos

Dado un tablero de ajedrez de tamaño $N \times N$ donde cada casilla tiene asignado un valor, haz una traza de un algoritmo Ramificación y Poda que sitúe N torres en dicho tablero de modo que las torres no se amenacen entre sí y que la suma de las casillas ocupadas por las torres sea máxima. (Nota: las torres se mueven libremente en horizontal y vertical.) Haz la traza para este ejemplo con N=4.

4	5	2	1	2
4 3	4	6	3	2
2	7	3	1	7
1	3	2	4	6
	1	2	3	4

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) Explica brevemente cómo vas a representar: el estado inicial, un estado incompleto y un estado solución. Pon un ejemplo sobre la instancia de cada tipo de estado.
- c) Para el ejemplo de solución factible que has dado, calcula su valor de función objetivo.
- d) Explica brevemente la cota optimista que vas a utilizar.
- e) Para la traza sobre la instancia presentada, ten en cuenta lo siguiente: La traza se debe mostrar el conjunto de estados activos de cada iteración del algoritmo indicando la cota optimista de cada estado (ej: como superíndice) y subrayando el estado que se selecciona para la siguiente iteración. Hay que indicar también si se actualiza la variable mejor solución y la poda implícita u otras podas.
- f) 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.

4 2.5 puntos

Disponemos de N objetos con pesos diferentes $w_1, w_2, ..., w_N$ que deseamos dividir en dos conjuntos de modo que ambos pesen exactamente lo mismo. Por ejemplo, si los pesos de los objetos son $\{7, 12, 1, 2, 8\}$, habría que repartirlos en dos conjuntos de forma que cada uno sume (7 + 12 + 1 + 2 + 8)/2 = 15. En este caso sí es posible: $\{7,8\}$ y $\{12, 1, 2\}$.

Se pide que implementes un algoritmo que, utilizando la **técnica de Búsqueda con Retroceso**, devuelva la solución encontrada, si es que existe.

- 1. Explica brevemente la estrategia a seguir.
- 2. Escribe un algoritmo (en seudo-código o python) que implemente la estrategia anterior.