

Ejercicios Tema 3

Algoritmia para problemas difíciles

Departamento de Informática e Ingeniería de Sistemas
Escuela de Ingeniería y Arquitectura – Universidad de Zaragoza

27 de noviembre de 2023

Ejercicios sobre algoritmos aproximados

Hay que elegir (por moodle) uno de los cinco primeros ejercicios y uno de los tres últimos. En cada ejercicio hay que encontrar una cota superior para el ratio de aproximación que consiga el algoritmo.

Ejercicio 1 Problema: *Equilibrado de Carga a dos velocidades.* El problema de Equilibrado de Carga consiste en asignar trabajos a máquinas intentando minimizar el tiempo máximo utilizado por una máquina o “makespan”, teniendo en cuenta que un trabajo no puede ser repartido entre varias máquinas. Supongamos que disponemos de m máquinas lentas y k máquinas rápidas. Las máquinas rápidas pueden realizar el doble de trabajo por unidad de tiempo que las máquinas lentas. Tenemos un conjunto de n trabajos: el trabajo i tarda tiempo t_i en una máquina lenta y tiempo $t_i/2$ en una máquina rápida.

Algoritmo: Voraz sin ordenar.

Ejercicio 2 Problema: *Equilibrado de Carga a velocidad acotada.* El problema de Equilibrado de Carga consiste en asignar trabajos a máquinas intentando minimizar el tiempo máximo utilizado por cada máquina o “makespan”, teniendo en cuenta que un trabajo no puede ser repartido entre varias máquinas. Tenemos n trabajos y m máquinas. Supongamos que no todos los procesadores tienen la misma velocidad. El procesador j tiene velocidad v_j , con $1 \leq v_j \leq 3$. El proceso i tarda tiempo t_i/v_j en ejecutarse en el procesador j .

Algoritmo: Voraz sin ordenar.

Ejercicio 3 Problema: *Equilibrado de Carga con tareas acotadas.* El problema de Equilibrado de Carga consiste en asignar trabajos a máquinas intentando minimizar el tiempo máximo utilizado por cada máquina o “makespan”, teniendo en cuenta que un trabajo no puede ser repartido entre varias máquinas. Tenemos n trabajos y m máquinas. Supongamos que no todos los procesadores tienen la misma velocidad. El procesador j tiene velocidad $v_j \geq 1$. El proceso i tarda tiempo t_i/v_j en ejecutarse en el procesador j .

Sea

$$T = \frac{\sum_{i=1}^n t_i}{\sum_{j=1}^m v_j}.$$

Es sencillo razonar que el makespan óptimo es mayor o igual que T . Supongamos que cada proceso no es demasiado grande, en concreto podemos asumir que para cada i , $t_i \leq T/2$.

Algoritmo: Voraz sin ordenar.

Ejercicio 4 Supón que actúas como consultor para la autoridad portuaria de un pequeño país del Pacífico. En la actualidad su facturación es de varios miles de millones de euros por año, y sus beneficios están limitados casi enteramente por la velocidad a la que pueden descargar los barcos que llegan a puerto. Aquí está el problema básico que afrontan. Llega un barco, con n contenedores de peso w_1, w_2, \dots, w_n . En el puerto hay esperando un conjunto de camiones, cada uno de los cuales puede transportar K unidades de peso. (Puedes suponer que K y cada w_i son enteros.) Puedes cargar varios contenedores en cada camión, sujeto a la restricción de peso K ; el objetivo es minimizar el número de camiones necesarios para transportar todos los contenedores.

Algoritmo: Voraz sin esperar: Empieza con un camión vacío y empieza a apilar contenedores $1, 2, 3, \dots$ en él, en el orden que lleguen, hasta que llegues a un contenedor que haga sobrepasar el límite de peso. Ahora declara este camión “cargado” y envíalo; entonces continúa el proceso con un nuevo camión.

Ejercicio 5 Los responsables de PLAZA quieren aumentar la velocidad de descarga de los aviones que llegan al aeropuerto de Zaragoza, pero también quieren limitar gastos.

El problema concreto es el siguiente. Llega un avión, con n bultos de peso p_1, p_2, \dots, p_n (los p_i son reales positivos). Los bultos no se pueden trocear. En el aeropuerto hay esperando un conjunto de furgonetas, cada una de las cuales puede transportar un peso máximo $M > 0$. Se pueden cargar varios bultos en cada furgoneta, sujeto a la restricción de peso M (asumimos que para todo i , $p_i \leq M$); el objetivo es minimizar el número de furgonetas necesarias para transportar todos los bultos.

Algoritmo: Voraz con espera: Empieza con una furgoneta vacía y empieza a apilar bultos $1, 2, 3, \dots$ en ella, en el orden que lleguen, hasta que llegues a un bulto que haga sobrepasar el límite de peso. En lugar de buscar enseguida una nueva furgoneta, prueba antes si el bulto cabe en alguna de las que ya están medio cargadas, para ello mantén todas las furgonetas que se están cargando en la pista rellenándolas todo lo posible hasta terminar de descargar el avión.

Ejercicio 6 Se debe realizar la mudanza de una biblioteca, para la cual se ha contratado una empresa de mudanzas que cobra por el número total de cajas a transportar. Se dispone de k_1 cajas de un metro de alto y k_2 cajas de 50 centímetros de alto. Tenemos n libros de distintos grosores g_i que se colocarán horizontalmente en las cajas. Queremos minimizar el coste de la mudanza.

Algoritmo: Voraz sin ordenar (ayuda, argumentar no puede haber dos cajas llenas a menos de la mitad).

Ejercicio 7 BINPACKING. Supongamos que se nos da un conjunto de n objetos, donde el tamaño s_i del objeto número i satisface $0 < s_i < 1$. Queremos meter todos los objetos en el número mínimo de cajas tamaño uno. Cada caja puede contener cualquier subconjunto de los objetos cuyo tamaño total no sobrepase 1.

Algoritmo: Voraz sin ordenar (ayuda, argumentar no puede haber dos cajas llenas a menos de la mitad).

Ejercicio 8 Supongamos que te dan un conjunto de enteros positivos $A = \{a_1, \dots, a_n\}$ y un entero positivo B . Un subconjunto $S \subseteq A$ se llama *factible* si la suma de los números de S no sobrepasa B :

$$\sum_{a_i \in S} a_i \leq B.$$

La suma de los números de S se llamará la *suma total* de S .

Te gustaría seleccionar un subconjunto factible S de A cuya suma total sea lo mayor posible.

Algoritmo: Voraz.

En caso de entregar alguno de estos ejercicios, la fecha límite es el lunes 4 de diciembre.

Antes de realizar cualquiera de estos ejercicios el alumno debe seleccionarlo en moodle.
--

Cualquier fuente utilizada en la resolución de estos ejercicios debe ser indicada claramente en la solución.
--