Algoritmos de Aproximação

Departamento de Ciência da Computação

Segundo semestre de 2012

Ajuda na Lista 3

1. Considere o exercício 3.5 do WS. O problema de escalonamento lá abordado consiste no seguinte: dadas n tarefas, a serem executadas em um número m de máquinas, onde cada tarefa tem um tempo de processamento p_i e um peso w_i, para $i = 1, \ldots, n$, encontrar um escalonamento sem interrupções das n tarefas nas m máquinas que minimize a soma ponderada por w do tempo de conclusão das tarefas. Chamamos essa soma de *valor* do escalonamento. O exercício pedia a solução deste problema quando m é uma constante.

   (a) Mostre que, se $m = 1$, o problema pode ser resolvido em tempo polinomial em n.

   (b) Mostre que, em um escalonamento ótimo, em cada máquina as tarefas estão ordenadas em ordem não decrescente da razão p_j/w_j.

   Um escalonamento em que as tarefas atribuídas a uma mesma máquina estão escalonadas na ordem dada por (b) é chamado de *padrão*. O item (b) garante que podemos nos concentrar em escalonamentos padrão.

   (c) Considere apenas o caso em que $m = 2$. Um escalonamento padrão neste caso é totalmente determinado por um conjunto $U \subseteq [n]$, que representa as tarefas escalonadas na máquina 1; as demais tarefas ficam na máquina 2.

   Reveja o algoritmo de PD visto em aula para o problema da mochila. Nele, para cada $j = 1, \ldots, n$, gerava-se uma lista $A(j)$ de pares (t, w), onde um par (t, w) em $A(j)$ indicava que existia um conjunto $S \subseteq [j]$ de itens cujos tamanhos somavam exatamente t e os valores somavam exatamente w.

   Projete um algoritmo de PD para o problema do escalonamento acima com $m = 2$, que gera, para $j = 1, \ldots, n$, uma lista $A(j)$ de pares (T, t), onde um par (T, t) em $A(j)$ indica que existe um escalonamento padrão das tarefas em $[j]$ (nas duas máquinas) com valor T em que a máquina 1 fica ocupada até exatamente o intante t. Nessa primeira versão do algoritmo, inclua em $A(j)$ todo par (T, t) que corresponde a um escalonamento padrão assim.

   (d) Dada a lista $A(n)$, como se determina o valor ótimo do problema?

   Um par (T, t) em $A(j)$ é *supérfluo* se podemos removê-lo de $A(j)$ antes do cálculo de $A(j+1), \ldots, A(n)$, e ainda assim podemos extrair o valor ótimo do problema da lista $A(n)$ obtida ao final da PD acima.

   (e) Suponha que $A(j)$ contem dois pares (T, t_1) e (T, t_2). Mostre que o par menos desbalanceado é supérfluo. (Ou seja, o par com t_i mais próximo de $\sum_{\ell=1}^{j} p_\ell/2$ é supérfluo.)

(f) Suponha que $A(j)$ contem dois pares (T_1, t_1) e (T_2, t_2), com $T_1 < T_2$ e $t_1 = t_2$ ou $t_1 = p([j]) - t_2$. Mostre que o par (T_2, t_2) é supérfluo.

(g) Adapte o algoritmo de PD do item (c) para que mantenha em $A(j)$ apenas pares que não são supérfluos de acordo com (e) e (f). Analise o consumo de tempo do algoritmo assim obtido para calcular o valor ótimo do problema.

(h) Considere n tarefas, onde cada tarefa tem um tempo de processamento p_i e um peso w_i, para $i = 1, \ldots, n$. Seja F_m o valor do escalonamento ótimo para tal conjunto de tarefas para um certo m. Quanto vale F_1? Observe que $F_1/2 \leq F_2 \leq F_1$.

Para obter um PTAS para o caso $m = 2$, para um $\epsilon > 0$, divida o intervalo $[0, F_1]$ em $\lceil 2n/\epsilon \rceil$ subintervalos, cada um de tamanho $\lceil \epsilon F_1/2n \rceil$. Para cada j, agrupe os pares (T, t) em $A(j)$ com componente T em um mesmo subintervalo, e mantenha em $A(j)$ apenas um par para em cada grupo. Para escolher qual par manter em $A(j)$ para um grupo, considere que todos os pares do grupo tem o mesmo valor de T e aplique a ideia do item (e).

(i) Calcule o consumo de tempo do algoritmo obtido desta maneira, em função de n e ϵ.

(j) Mostre que o escalonamento produzido por este algoritmo tem valor no máximo $\text{opt} + \epsilon F_1/2$, e conclua que a família de algoritmos é um PTAS para o problema. A família resultante é um FPTAS para o problema?