Problema da 2-Partição Mínima

Notas de aula de MAC-5727

(Material Extra do Capítulo 2 – PTAS)

Prof.ª Yoshiko Wakabayashi

— Transcrito por Rafael da P. Barbosa em março de 2009 —

agosto de 2016

Problema $\textsc{MinPart}(X, w)$: *Dados um conjunto de n itens, $X = \{x_1, x_2, \ldots, x_k\}$, e um peso $w_i \in \mathbb{Z}_{\geq}$ para cada item x_i, $1 \leq i \leq n$, encontrar uma 2-partição $\{X_1, X_2\}$ de X que minimiza $max\{w(X_1), w(X_2)\}$.*

Sabe-se que o problema MinPart é NP-difícil (redução via problema da 2-Partição Exata [1]). Apresentamos a seguir um PTAS para esse problema. A ideia do algoritmo $\textsc{MinPart}_\varepsilon$, descrito abaixo, é resolver otimamente uma subinstância que consiste dos $k(\varepsilon)$ itens mais pesados, e depois estender a solução obtida. A extensão é feita acrescentando-se um a um os itens restantes, de forma a manter cada 2-partição parcial a mais equilibrada possível.

Algoritmo $\textsc{MinPart}_\varepsilon(X, w)$

```
Se (1 + ε) ≥ 2 então devolva a solução {X, ∅}
Senão
    Ordene os itens em ordem decrescente de peso:
    considere x_1, x_2, ..., x_n a ordem obtida
    k(ϵ) ← ⌈(1 + ε)/ε⌉
    /* Fase 1 */
    Encontre uma 2-partição ótima {X_1, X_2} de {x_1, x_2, ..., x_k(ε)}
    /* Fase 2 */
    Para j = k(ε) + 1, ..., n faça
        Se w(X_1) ≤ w(X_2) então X_1 ← X_1 ∪ {x_j}
        Senão X_2 ← X_2 ∪ {x_j}
    Devolva {X_1, X_2}
```

Teorema 1: *O algoritmo* $\textsc{MinPart}_\varepsilon$ *é um PTAS para o problema* $\textsc{MinPart}$.

Demonstração: Considere (X, w) uma instância do problema $\textsc{MinPart}$ ao qual aplicamos o algoritmo $\textsc{MinPart}_\varepsilon$. Seja $M := w(X)/2$. Claramente,

$$\operatorname{opt}(X, w) \geq M. \tag{1}$$

Vejamos os dois casos a seguir:

(A) Se $(1 + \epsilon) \geq 2$, então claramente a solução $\{X, \emptyset\}$ é uma 2-aproximação e, portanto, uma $(1 + \epsilon)$-aproximação.

(B) Suponha $(1 + \epsilon) < 2$. Suponha também, sem perda de generalidade, que $w(X_1) \geq w(X_2)$ e que x_u é o item que foi adicionado por último em X_1.
Então,

$$\begin{aligned} w(X_1) - w_u &\leq w(X_2) \\ w(X_1) &\leq w(X_2) + w_u \\ w(X_1) &\leq w(X) - w(X_1) + w_u \\ w(X_1) &\leq w(X)/2 + w_u/2, \end{aligned}$$

e portanto,

$$w(X_1) \leq M + w_u/2. \tag{2}$$

Se x_u foi adicionado a X_1 na Fase 1 do algoritmo, então a solução devolvida pelo algoritmo é ótima. Suponha então que x_u não foi adicionado na Fase 1. Neste caso, temos que $w_j \geq w_u$, para todo j tal que $1 \leq j \leq k(\epsilon)$.
Logo,

$$2M = w(X) = \sum_{i=1}^{n} w_i \geq \sum_{i=1}^{k(\varepsilon)+1} w_i \geq (k(\varepsilon) + 1) w_u,$$

de onde segue que

$$w_u/2 \leq M/(k(\varepsilon) + 1). \tag{3}$$

Da definição de $k(\varepsilon)$ no passo 4 do algoritmo, temos que

$$1/(k(\varepsilon) + 1) \leq \varepsilon. \tag{4}$$

Combinando as desigualdades (2), (3), (4) e (1), obtemos

$$\begin{aligned} w(X_1) &\leq M + w_u/2 \\ &\leq M + M/(k(\varepsilon) + 1) \\ &= M(1 + 1/(k(\varepsilon) + 1)) \\ &\leq M(1 + \varepsilon) \\ &\leq (1 + \varepsilon) \operatorname{opt}(X, w). \end{aligned}$$

Logo, o algoritmo MinPart$_\varepsilon$ tem razão de aproximação $(1+\varepsilon)$. Para um ε fixo, o consumo de tempo desse algoritmo é polinomial em n (ordenação de n elementos no passo 3, e teste linear em n nos passos 6–8 do algoritmo). (Notamos que o algoritmo consome tempo exponencial em $1/\varepsilon$ no passo 5 — para ε fixo tal tempo é considerado constante). Logo, o algoritmo MinPart$_\varepsilon$ é um PTAS para o problema MinPart.

□

EXERCÍCIO.

Considere o seguinte algoritmo MinPartGuloso para resolver o problema MinPart (X, w).

Ordene os itens de X em ordem decrescente de seus pesos. Comece com $X_1 = X_2 = \emptyset$, e obedecendo a ordem obtida, insira os itens em X_1 ou em X_2 de acordo com a seguinte regra: sempre insira o próximo item no conjunto com o menor peso total (se houver empate, desempate arbitrariamente).

Ou seja, relativamente ao algoritmo MinPart$_\varepsilon$ a diferença é que começa-se pela ordenação (passo 3), pula-se a Fase 1 e executa-se a Fase 2 (passos 6–8) desse algoritmo com $k(\varepsilon) = 0$.

Prove que esse algoritmo guloso tem razão de aproximação 7/6 e que essa razão é justa. Se não conseguir obter 7/6, prove a melhor razão menor que 2 que você conseguir.

Referências

[1] M.R. Garey and D.S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness.* Freeman, 1979.