Dual Fitting

Notas de aula de MAC-5727

(Material Extra do Capítulo 4)

Prof.ª Yoshiko Wakabayashi

– Versão pós-aula feita por César Gamboa em 12/04/09 –

9 de junho de 2009

Veremos aqui o método *dual fitting*, que pode ser usado para analisar desempenhos de algoritmos combinatórios usando teoria da dualidade de programação linear. Analisaremos aqui o desempenho do algoritmo guloso para o problema MinCC visto no Capítulo 2 (do livro-texto), devido a Chvátal. A apresentação é baseada na Seção 13.1 do livro de Vazirani [1].

Problema MinCC $(E, \mathcal{S}, c)$: *Dados E um conjunto finito, $\mathcal{S}$ uma coleção finita de subconjuntos de E e um custo $c_S \in \mathbb{Q}_{\geq}$ para cada $S \in \mathcal{S}$, encontrar uma cobertura τ de E que minimize $c(\tau)$.*

Podemos supor, spg, que os subconjuntos em $\mathcal{S}$ coletivamente cobrem E.

Considere o seguinte algoritmo guloso (visto em aula) para este problema:

Algoritmo MinCC-Chvatal$(E, \mathcal{S}, c)$

1. se $E = \emptyset$
2. então devolva $\emptyset$
3. senão seja Z em $\mathcal{S}$ tal que $c_Z/|Z \cap E|$ é mínimo
4. $E' \leftarrow E \setminus Z$; // *price*$(e) = c_Z/|Z \cap E|$ para todo $e \in Z \cap E$
5. $\mathcal{S}' \leftarrow \{S \in \mathcal{S} : S \cap E' \neq \emptyset\}$
6. seja c' a restrição de c a $\mathcal{S}'$
7. $\tau' \leftarrow$ MinCC-Chvatal$(E', \mathcal{S}', c')$
8. devolva $\{Z\} \cup \tau'$

OBS: Na linha 4, a informação *price*(e) foi introduzida para ajudar na análise a seguir. Corresponde aos "preços" dos elementos de E. Note que estes são atribuídos no momento em que os elementos são cobertos pelo algoritmo.

Teorema 1: *O algoritmo* MinCC-Chvatal *é uma H_n-aproximação para o problema* MinCC

Demonstração: (V_D) é Essas desigualdades, válidas para Considere a seguinte relaxação linear de MinCC$(E, \mathcal{S}, c)$.

[Primal P] Encontrar um vetor x indexado por $\mathcal{S}$ que:

$$\begin{array}{rl} \text{minimize} & cx \\ \text{sob as restrições} & \sum_{S \ni e} x_S \geq 1 \text{ para cada } e \in E \\ & x_S \geq 0 \text{ para cada } S \in \mathcal{S}. \end{array}$$

O correspondente problema dual é o seguinte.

[Dual D] Encontrar um vetor y indexado por E que:

$$\begin{array}{rl} \text{maximize} & y(E) \\ \text{sob as restrições} & \sum_{e \in S} y_e \leq c_S \text{ para cada } S \in \mathcal{S} \\ & y_e \geq 0 \text{ para cada } e \in E. \end{array}$$

Dada uma instância $(E, \mathcal{S}, c)$, o algoritmo guloso MinCC-Chvatal encontra uma cobertura, digamos Λ, que define uma solução primal inteira: $x_S = 1$ se $S \in \Lambda$; e $x_S = 0$, caso contrário.

O valor *price*(e) definido na linha 4 do algoritmo pode ser escrito como

$$\mathit{price}(e) = \frac{c(Z)}{|Z - \Lambda_Z|},$$

onde Λ_Z é o conjunto de elementos de E que já estão cobertos no momento em que Z é escolhido. O vetor *price* (indexado por E) é uma solução dual que pode ser inviável: não necessariamente satisfaz a restrição $\sum_{e \in S} \mathit{price}(e) \leq c_S$ para cada $S \in \mathcal{S}$. Porém, podemos notar que o valor desta solução dual (V_D) é pelo menos o valor da solução primal (V_P) encontrada pelo algoritmo. De fato, note que se $S \in \Lambda$, então

$$c(S) = \sum_{e \in S} \frac{c(S)}{|S|} \leq \sum_{e \in S} \frac{c(S)}{|S - \Lambda_S|} = \sum_{e \in S} \mathit{price}(e).$$

Portanto,

$$V_P = \sum_{S \in \Lambda} c(S) \leq \sum_{e \in E} \mathit{price}(e) = V_D.$$

Vamos então "encolher essa solução dual", de modo a obter uma solução dual viável. Considere agora

$$y_e := \frac{\mathit{price}(e)}{H_n} \quad \text{para todo } e \in E.$$

Vamos mostrar que y define uma solução viável para o problema dual D.

Considere $S \in \mathcal{S}$ e chame de $e_1, e_2, \ldots, e_k$ os elementos de S na ordem em que são cobertos pelo algoritmo guloso.

Considere a iteração na qual o algoritmo cobre o elemento e_i. Neste momento, S contém pelo menos $k - i + 1$ elementos não-cobertos. Assim, nessa iteração o próprio conjunto S pode cobrir e_i a um custo médio — por elemento novo — de no máximo $c(S)/(k - i + 1)$. Como o algoritmo escolhe sempre o conjunto de melhor custo-benefício nessa iteração, segue que

$$\mathit{price}(e_i) \leq \frac{c(S)}{k - i + 1}.$$

Portanto,

$$y_{e_i} \leq \frac{c(S)}{(k - i + 1)\, H_n}.$$

Fazendo a soma para todos os elementos em S, obtemos

$$\sum_{e \in S} y_e \leq \sum_{i=1}^{k} \frac{c(S)}{(k - i + 1)\, H_n} = \frac{c(S)}{H_n} \sum_{i=1}^{k} \frac{1}{k - i + 1} = c(S) \frac{H_k}{H_n} \leq c(S).$$

Portanto, y é uma solução viável para o problema dual D, e consequentemente,

$$y(E) \leq \mathrm{opt}_D = \mathrm{opt}_P \leq \mathrm{opt}(E, \mathcal{S}, c),$$

onde opt_D (resp. opt_P) denota o valor ótimo do problema dual D (resp. primal P).

Finalmente, podemos concluir que

$$c(\Lambda) = \sum_{S \in \Lambda} c(S) = V_P \leq V_D = H_n\, y(E) \leq H_n\, \mathrm{opt}(E, \mathcal{S}, c).$$

Com isso, completamos a prova de que o algoritmo MinCC-Chvatal produz uma H_n-aproximação para o problema MinCC. $\square$

Referências

[1] V.V. Vazirani. *Approximation Algorithms*. Springer, 2001.