Algorytmy online: lista 8

Zadanie 1. (3 pkt.) Rozważmy problem routingu na linii o n wierzchołkach, gdzie każda krawędź ma pojemność U. Pokaż dolne ograniczenie na ścisłą konkurencyjność dowolnego algorytmu deterministycznego w wysokości $\Omega(n^{1/U})$.

Zadanie 2. W problemie szeregowania zadań na m *niepowiązanych* maszynach dany jest ciąg zadań, gdzie zadanie t jest zdefiniowane wektorem $p_t = (p_{t,1}, p_{t,2}, \ldots, p_{t,m})$: jeśli zadanie przypiszemy maszynie j, to jej obciążenie ℓ_j zwiększy się o $p_{t,j}$. Celem jest minimalizacja $\max_{j=1}^m \ell_j$. Tak jak w przypadku algorytmu dla powiązanych maszyn chcemy stworzyć procedurę $\textsc{Alg}_\lambda$, która będzie przypisywać zadania *przy założeniu, że* $\textsc{Opt} \leq \lambda$.

Niech $y_{t,j} = 1$ jeśli zadanie t przypisujemy do maszyny j, zaś 0 w przeciwnym przypadku. W kroku k zdefiniujmy następujący program całkowitoliczbowy $\mathcal{P}_k^{\text{int}}$, którego celem jest maksymalizacja liczby przypisanych zadań.

$$
\begin{aligned}
\text{maksymalizuj:} \quad & \sum_{t=1}^{k} \sum_{j=1}^{m} y_{t,j} \quad , \\
\text{przy zachowaniu warunków:} \quad & \sum_{j=1}^{m} y_{t,j} \leq 1 && \text{dla każdego } t \in \{1, \ldots, k\}, \\
& \sum_{t=1}^{k} p_{t,j} \cdot y_{t,j} \leq \lambda && \text{dla każdego } j \in \{1, \ldots, m\}, \\
& y_{t,j} \in \{0, 1\} && \text{dla każdego } t \in \{1, \ldots, k\}, j \in \{1, \ldots, m\}.
\end{aligned}
$$

Niech $\mathcal{P}_k$ będzie liniową relaksacją $\mathcal{P}_k^{\text{int}}$, w którym warunek $y_{t,j} \in \{0, 1\}$ został zastąpiony przez $y_{t,j} \geq 0$ (Warunek $y_{t,j} \leq 1$ jest zbędny, bo implikowany przez $\sum_{j=1}^m y_{t,j} \leq 1$). Programem dualnym do $\mathcal{P}_k$ jest następujące zagadnienie minimalizacyjne $\mathcal{D}_k$:

$$
\begin{aligned}
\text{minimalizuj:} \quad & \sum_{j=1}^{m} \lambda \cdot x_j + \sum_{t=1}^{k} z_t \\
\text{przy warunkach:} \quad & z_t + p_{t,j} \cdot x_j \geq 1 && \text{dla każdego } t \in \{1, \ldots, k\}, j \in \{1, \ldots, m\}. \\
& x_j \geq 0 && \text{dla każdego } j \in \{1, \ldots, m\}, \\
& z_t \geq 0 && \text{dla każdego } t \in \{1, \ldots, k\}.
\end{aligned}
$$

Algorytm ALG$_\lambda$. Na początku niech $x_j = 1/(2\lambda \cdot m)$ dla każdego $j \in \{1, \ldots, m\}$. W kroku k niech $\mathcal{S}_k = \{j : p_{k,j} \leq \lambda\}$. Następnie:

- Jeśli $\mathcal{S}_k = \emptyset$ lub jeśli $x_j > 1/\lambda$ dla pewnego j, to zwróć `FAIL`.
- W przeciwnym przypadku niech j^* będzie maszyną z $\mathcal{S}_k$ minimalizującą $p_{k,j^*} \cdot x_{j^*}$.
  - **a)** przypisz zadanie k do maszyny j^*
  - **b)** $z_t \leftarrow 1 - p_{k,j^*} \cdot x_{j^*}$
  - **c)** $x_{j^*} \leftarrow x_{j^*} \cdot \left(1 + \frac{p_{k,j^*}}{2 \cdot \lambda}\right)$

a) **(2 pkt.)** Załóżmy, że $\textsc{Alg}$ nie zwróci `FAIL`. Ustalmy j. Pokaż, że w dowolnym momencie zachodzi

$$x_j \geq \frac{1}{2\lambda \cdot m} \cdot C^{\ell_j/\lambda}$$

dla pewnego $C > 1$.[1] Wykorzystaj tę nierówność do pokazania, że $\ell_j = O(\log m) \cdot \lambda$.

[1] Wskazówka: hasło do wyszukiwarki „useful inequalities".

b) (2 pkt.) Załóżmy, że ALG nie zwróci `FAIL`. Pokaż, że w dowolnego kroku k, ALG generuje dopuszczalne rozwiązanie dla $\mathcal{D}_k$ i zachodzi

$$\textsc{Alg}(\mathcal{D}_k) = k + 1 - \lambda \cdot \sum_{j=1}^{m} x_j .$$

c) (3 pkt.) Pokaż, że jeśli ALG zwróci `FAIL`, to OPT nie jest w stanie przypisać wszystkich zadań tak, żeby obciążenie każdej maszyny było co najwyżej λ.

Marcin Bieńkowski