

# 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.** Rozważmy problem szeregowania zadań na  $m$  *niepowiązanych* maszynach. W kroku  $t$  otrzymujemy zadanie zdefiniowane wektorem  $p_t = (p_{t,1}, p_{t,2}, \dots, p_{t,m})$ . Jeśli przypiszemy je maszynie  $j$ , to jej obciążenie  $\ell_j$  zwiększy się o  $p_{t,j}$ . Celem jest minimalizacja  $\max_{j=1}^m \ell_j$ .

Podobnie jak w przypadku algorytmu dla powiązanych maszyn chcemy stworzyć procedurę  $\text{ALG}_\lambda$ , która będzie przypisywać zadania *przy założeniu*, że  $\text{OPT} \leq \lambda$  (i będzie gwarantować, że obciążenie każdej maszyny to  $O(\lambda \cdot \log m)$ ).

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

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

Niech  $\mathcal{P}_k$  będzie liniową relaksacją  $\mathcal{P}_k^{\text{int}}$ , w której 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  $\mathcal{D}_k$ .

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

Algorytm  $\text{ALG}_\lambda$  działa następująco. Na początku przypisuje  $x_j \leftarrow 1/(2\lambda \cdot m)$  dla każdego  $j \in \{1, \dots, m\}$ . W kroku  $k$  algorytm ustala zbiór  $\mathcal{S}_k \leftarrow \{j : p_{k,j} \leq \lambda\}$  i wykonuje następujące czynności.

- Jeśli  $\mathcal{S}_k = \emptyset$ , to zwróć FAIL.
- W przeciwnym przypadku niech  $j^*$  będzie maszyną z  $\mathcal{S}_k$  minimalizującą  $p_{k,j^*} \cdot x_{j^*}$ . Następnie:
  - przypisz zadanie  $k$  do maszyny  $j^*$  (tj.  $y_{k,j^*} \leftarrow 1$  oraz  $y_{k,j} \leftarrow 0$  dla  $j \neq j^*$ ),
  - $z_k \leftarrow 1 - p_{k,j^*} \cdot x_{j^*}$ ,
  - $x_{j^*} \leftarrow x_{j^*} \cdot \left(1 + \frac{p_{k,j^*}}{2\lambda}\right)$ .
- Jeśli istnieje maszyna  $j$ , dla której  $x_j > 1/\lambda$ , to zwróć FAIL.

Niech  $k$  będzie pierwszym krokiem, w którym ALG zwróci FAIL, lub ostatnim krokiem wejścia, jeśli ALG nigdy nie zwróci FAIL. Rozwiązanie generowane przez ALG przypisuje wszystkie  $k$  zadań do maszyn, ale odpowiadające mu rozwiązanie dla  $\mathcal{P}_k^{\text{int}}$  być może nie jest dopuszczalne (obciążenia maszyn mogą być większe od  $\lambda$ ).

- a) (2 pkt.) Pokaż, że dla dowolnego kroku  $t \in \{0, \dots, k\}$ , ALG generuje dopuszczalne rozwiązanie dla  $\mathcal{D}_t$  i zachodzi

$$\text{ALG}(\mathcal{D}_t) = t + 1 - \lambda \cdot \sum_{j=1}^m x_j^{(t)},$$

gdzie  $x_j^{(k)}$  to wartość zmiennej  $x_j$  po kroku  $k$ .

- b) (2 pkt.) Pokaż, że jeśli ALG zwróci FAIL w kroku  $k$ , to OPT nie jest w stanie przypisać wszystkich  $k$  zadań tak, żeby obciążenie każdej maszyny wynosiło co najwyżej  $\lambda$ .
- c) (2 pkt.) Pokaż, że dla dowolnego kroku  $t \in \{1, \dots, k\}$  oraz dla dowolnej maszyny  $j$  zachodzi

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

Jak z tego wynika  $\ell_j = O(\lambda \cdot \log m)$ ?

- d) (1 pkt.) Jak z poprzednich punktów wynika następujące twierdzenie: „Jeśli pierwsze  $k$  zadań można przypisać z obciążeniem co najwyżej  $\lambda$  dla każdej maszyny, to ALG przypisze te zadania generując obciążenie  $O(\lambda \cdot \log m)$  dla każdej maszyny i nie zwróci FAIL”.

*Marcin Bieńkowski*