

Algorytmy online: lista 6

Zadanie 1. Rozważmy problem szeregowania na m identycznych maszynach (o takiej samej prędkości). Celem jest minimalizacja obciążenia najbardziej obciążonej maszyny.

- a) (2 pkt.) Pokaż, że strategia zachłanna (przypisujemy zadanie do najmniej obciążonej maszyny) ma współczynnik ścisłej konkurencyjności równy $2 - 1/m$ (tj. pokaż dolne i górne ograniczenie).
- b) (2 pkt.) Pokaż, że dla dowolnego $m \geq 2$, ścisła konkurencyjność dowolnego randomizowanego algorytmu wynosi co najmniej $4/3$.

Zadanie 2. (2 pkt.) Pokaż, że ścisła konkurencyjność problemu ONLINE BIDDING wynosi co najmniej 4. Możesz skorzystać z następującego schematu:

- a) Ustal nieskończony ciąg $0 < d_1 < d_2 < d_3 \dots$ definiujący algorytm i oznacz konkurencyjność algorytmu przez R . Zdefiniuj $s_j = \sum_{i=1}^j d_i$ i pokaż, że dla każdego j musi zachodzić $s_{j+2} \leq R \cdot (s_{j+1} - s_j)$.
- b) Zdefiniuj $b_j = s_{j+1}/s_j$ i pokaż, że

$$(b_{j+1} - b_j) \cdot b_j \leq -b_j^2 + R \cdot b_j - R . \quad (1)$$

- c) Pokaż, że dla $R < 4$ prawa strona (1) jest ujemna, a zatem ciąg $\{b_j\}_j$ jest malejący.
- d) Zaobserwuj, że ciąg $\{b_j\}_j$ jest zbieżny (dlaczego?). Niech $b = \lim_{j \rightarrow \infty} b_j$. Do czego zbiega lewa i prawa strona (1)?

Zadanie 3 (2 pkt.) Rozważamy szeregowanie zadań na m powiązanych maszynach, a funkcją kosztu jest maksymalne obciążenie maszyny. Przedstawiony na wykładzie algorytm SLOWFIT(λ) ma następujące własności.

- Przypisuje zadania do maszyn, tak że maksymalne obciążenie wynosi co najwyżej 2λ . Być może dla pewnego zadania zamiast numeru maszyny algorytm zwróci komunikat „FAIL”.
- Jeśli zostanie uruchomiony na wejściu σ , dla którego $\text{OPT}(\sigma) \leq \lambda$, to wszystkie zadania zostaną przypisane do maszyn (komunikat „FAIL” nie wystąpi).

Pokaż, jak za jego pomocą skonstruować 8-konkurencyjny algorytm.

Zadanie 4. (2 pkt.) (Problem inkrementalnych median). Dane są dwa zbiory: K (klienci) i M (miejsca na zbudowanie fabryk) oraz funkcja odległości $d(k, m)$ zdefiniowana dla dowolnej pary $(k, m) \in K \times M$.¹ Mówimy, że zbiór fabryk $F \subseteq M$ ma koszt

$$\text{COST}(F) \triangleq \sum_{k \in K} d(k, F) \triangleq \sum_{k \in K} \min_{f \in F} d(k, f) .$$

Dla rozmiaru $s \in \{1, \dots, |M|\}$ minimalny koszt zbioru s fabryk oznaczamy przez $\text{OPT}(s)$. Przykładowo $\text{OPT}(|M|)$ jest kosztem postawienia fabryki w każdym dostępnym miejscu.

¹Możesz założyć, że funkcja d spełnia nierówność trójkąta, choć nie jest to potrzebne w tym zadaniu.

Zadaniem algorytmu jest rozbudowa zbioru fabryk, tj. skonstruowanie ciągu zbiorów $F_1 \subseteq F_2 \subseteq \dots \subseteq F_{|M|-1} \subseteq F_{|M|}$, takiego że dla każdego $i \in \{1, \dots, |M|\}$ zachodzą warunki

- a)** $\text{COST}(F_i) \leq \text{OPT}(i)$,
- b)** $|F_i| \leq 4 \cdot i$.

Wskazówka: to zadanie jest proste, tylko ma długą treść. Być może do konstrukcji ciągu F_i konieczne będzie rozwiązanie jakiegoś NP-trudnego problemu, ale nie należy się tym martwić.

Marcin Bienkowski