

Problem podziału na 3 podzbiory

Krzysztof Król Szymon Dudycz

23 marca 2016

Opis problemu

Zbiór liczb należy podzielić na 3 podzbiory tak, aby:

- ▶ każda liczba należała do dokładnie jednego podzbioru
- ▶ sumy liczb elementów każdego podzbioru jak najmniej się różniły

Model matematyczny

Wejście problemu: Dane jest n liczb naturalnych a_1, \dots, a_n

Wyjście problemu:

Dla każdej liczby a_i i zbioru j tworzymy zmienną $x_{i,j}$ oznaczającą czy a_i została przydzielona do zbioru j . Wtedy problem można wyrazić za pomocą następującego programu całkowitoliczbowego:

$$\begin{aligned} \min \quad & \left| \sum_{i=1}^n x_{i,1} a_i - \sum_{i=1}^n x_{i,2} a_i \right| + \left| \sum_{i=1}^n x_{i,1} a_i - \sum_{i=1}^n x_{i,3} a_i \right| \\ & + \left| \sum_{i=1}^n x_{i,2} a_i - \sum_{i=1}^n x_{i,3} a_i \right| \\ x_{i,1} + x_{i,2} + x_{i,3} = 1 \quad & \forall i \in \{1, \dots, n\} \\ x_{i,j} \in \{0, 1\} \quad & \forall i \in \{1, \dots, n\} \quad \forall j \in \{1, 2, 3\} \end{aligned}$$

Górne ograniczenie

Górne ograniczenie można wyznaczyć algorytmem Longest Processing Time. Można udowodnić, że ten algorytm osiąga górne ograniczenie nie gorsze od $11/9 * \textit{optimal}$ dla problemu podziału ciągu, w którym chcemy minimalizować sumę elementów największego podzbioru.

LPT:

$s_1 \leftarrow \emptyset;$

$s_2 \leftarrow \emptyset;$

$s_3 \leftarrow \emptyset;$

$m \leftarrow$ posortowany ciąg wejściowy w nierosnącej kolejności;

$iter \leftarrow 0;$

while $iter < n$ **do**

$s \leftarrow \min(s_1, s_2, s_3);$

$s \leftarrow s \cup \{m_{iter}\};$

$iter \leftarrow iter + 1;$

end

Dolne ograniczenie

Jakieś pomysły?

Algorytmy

- ▶ Algorytm pseudo wielomianowy $O(M^2n)$ gdzie M to suma elementów ciągu wejściowego
- ▶ Heurystyka Karmakar-Karp
- ▶ Meta-heurystyka - symulowane wyżarzanie