PD wykład 6

Niech $c \in \boldsymbol{R}^n, A \in \boldsymbol{R}^{m,n}, b \in \boldsymbol{R}^m$

$$(IP) \quad z = \max\{ c^T x \colon Ax \leq b, x \geq 0, x \in \boldsymbol{Z}^n\}$$

Ograniczenia na funkcję celu

1. Ograniczenia dolne

- poprzez każde rozwiązanie dopuszczalne (algorytmy heurystyczne)

2. Ograniczenia górne

- dualność
- relaksacje

Ad. 1. Algorytmy zachłanne

Przykład. Problem plecakowy

$$\max \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_j x_j \leq b$$

$$x_j \in \{0,1\}$$

Sortowanie według $\frac{c_j}{a_j}$

Ad. 1. Przegląd lokalny (local search) – znajduje ekstremum lokalne poprzez przeszukiwanie sąsiedztwa kolejnych coraz lepszych rozwiązań.

Przykład: problem lokalizacyjno transportowy (UFL)

dane:

zbiór $N = \{1, \dots, n\}$ potencjalnych lokalizacji (np. fabryk)
zbiór $M = \{1, \dots, m\}$ odbiorców
f_j - koszt budowy fabryki w lokalizacji *j*
c_{ij} - sumaryczny koszt transportu, jeśli całość wszystkich zamówień (w analizowanym okresie) odbiorcy *i* jest realizowana przez fabrykę *j*

zmienne:

$$y_j = \begin{cases} 1, & jeśli\ ma\ być\ otwarta\ fabryka\ w\ lokalizacji\ j \\ 0, & w\ p.p. \end{cases}$$

x_{ij} - % udział całkowitego zapotrzebowania odbiorcy *i* realizowany przez fabrykę *j*

$$
(c_{ij}) = \begin{bmatrix} 6 & 2 & 3 & 4 \\ 1 & 9 & 4 & 11 \\ 15 & 2 & 6 & 3 \\ 9 & 11 & 4 & 8 \\ 7 & 23 & 2 & 9 \\ 4 & 3 & 1 & 5 \end{bmatrix}
$$

$$
f_j = (21,16,11,24)
$$

$$
c(S) = \sum_{i=1}^{6} \min_{j \in S} c_{ij} + \sum_{j \in S} f_j \qquad \text{gdzie} \quad S \subseteq N
$$

Sąsiedztwo $Q(S)$ zbioru S możemy zdefiniować

$$
Q(S) = \{T \subseteq N : T = S \setminus i, i \in S \quad lub \quad T = S \cup j, j \in N \setminus S\}
$$

Inne

Programowanie dynamiczne (będzie więcej wkrótce)
Algorytmy randomizowane
Algorytmy genetyczne
Sieci neuronowe
…

Ad. 2.

Def. Problem

$$(P') \qquad z' = \max\{f(x) \colon x \in T\}$$

jest ***relaksacją*** problemu

$$(P\ ) \qquad z\ = \max\{c(x) \colon x \in X\}$$

jeśli

- $X \subseteq T$ oraz
- $c(x) \leq f(x) \qquad \forall x \in X$

Twierdzenie.

Jeśli P’ jest relaksacją dla P, to $z' \geq z$.

Def. Dla problemu

$$(IP) \quad z = \max\{ c^T x : Ax \leq b, x \geq 0, x \in \boldsymbol{Z}^n\}$$

jego ***liniową relaksacją*** jest problem programowania liniowego

$$(LP) \quad z_L = \max\{ c^T x : Ax \leq b, x \geq 0\}.$$

Uwaga! $c \in Z^m \Rightarrow \quad z \leq \lfloor z_L \rfloor$.

Zalety: rozwiązanie spełnia wszystkie ograniczenia
Wady: rozwiązanie nie jest całkowite

Def. Dla problemu

$$(IP) \quad z = \max\{\, c^T x : Ax \leq b, x \geq 0, x \in \boldsymbol{Z}^n\}$$

oraz wektora $u \geq 0, u \in \boldsymbol{R}^m$, jego ***relaksacją Lagrange'a*** jest problem

$$(IP) \quad z_{L,u} = \max\{c^T x + u(b - Ax) : x \geq 0, x \in \boldsymbol{Z}^n\}.$$

Zalety: rozwiązanie jest całkowite

Wady: rozwiązanie nie spełnia wszystkich ograniczeń

Uwaga! Rozwiązanie nie jest zbyt odległe od obszaru dopuszczalnego.

Relaksacje kombinatoryczne.

Przykład. Symetryczny problem komiwojażera STSP

$$z_{STSP} = \min\left\{\sum_{e\in T} c_e : T\ jest\ cyklem\ Hamiltona\right\} \geq z' = \min\left\{\sum_{e\in T} c_e : T\ jest\ 1-drzewem\right\}$$

Uwaga! Każda relaksacja musi mieć niską złożoność obliczeniową

(algorytm Kruskala dla problemu minimalnego drzewa rozpinającego).