

Metody Optymalizacji - Lista 1

Janusz Witkowski 254663

2 kwietnia 2023

1 Zadanie 1

1.1 Model

Mamy daną szerokość bazowej deski $b \in \mathbb{Z}_+$, żądane szerokości desek $w \in \mathbb{Z}_+^n$, oraz popyt na takowe deski $d \in \mathbb{Z}_+^n$. Z tych danych można wyznaczyć wszystkie sposoby pocięcia deski $c \in \mathbb{N}^{n \times m}$.

1.1.1 Zmienne decyzyjne

Należy wyznaczyć liczby bazowych desek przeznaczonych do poszczególnych cięć: $x \in \mathbb{N}^m$

1.1.2 Ograniczenia

Jeśli jesteśmy w stanie zagwarantować poprawność cięć c , wystarczy nam jedno ograniczenie:

1. **Nasyć popyt** - wyprodukuj co najmniej tyle desek każdego typu ile wynosi podaż.

$$(\forall i \in \{1, \dots, n\}) \left(\sum_{j=1}^m c_{ij} \cdot x_j \right) \geq d_i$$

1.1.3 Funkcja celu

Chcemy zminimalizować liczbę niewykorzystanych desek, jak i drewno które zostanie nam z nieidealnych cięć:

$$\min \left(\sum_{i=1}^n \left(\sum_{j=1}^m c_{ij} \cdot x_j - d_i \right) \cdot w_i \right) + \left(\sum_{j=1}^m x_j \cdot \left(b - \sum_{i=1}^n w_i \cdot c_{ij} \right) \right)$$

1.2 Wyniki

Dla $b = 22$, $w = [7, 5, 3]$ oraz $d = [110, 120, 80]$ mamy następujące cięcia oraz rozwiązanie:

Nr	1	2	3	4	5	6	7	8	9	10	11	12
Cięcia	3/0/0	2/1/1	2/0/2	1/3/0	1/2/1	1/1/3	1/0/5	0/4/0	0/3/2	0/2/4	0/1/5	0/0/7
Rozwiązanie	0	37	0	28	0	0	9	0	0	0	0	0

Stosując wskazane cięcia, uzyskamy wartość funkcji celu 18 oraz następujące liczby desek:

Deski szerokości 7	Deski szerokości 5	Deski szerokości 3	Drewno z niewykorzystanych desek	Drewno z odpadków
111	121	82	18	0

2 Zadanie 2

2.1 Model

Mamy daną liczbę zadań $n \in \mathbb{N}$, czasy wykonywania zadań $p \in \mathbb{N}^n$, momenty gotowości zadań $r \in \mathbb{N}^n$ oraz wagi zadań $w \in \mathbb{R}^n$. Definiujemy dodatkowo tzw. *horyzont* $T = 1 + \max r + \sum_{i=1}^n p_i$, który posłuży nam do ustalenia tabeli czasu w grafiku maszyny.

2.1.1 Zmienne decyzyjne

Definiujemy zmienną decyzyjną w postaci tabeli czasu: $C \in \{0, 1\}^{n \times T}$. Jeżeli w momencie $t = 1 \dots T$ rozpoczyna się zadanie $i = 1 \dots n$, to $C_{it} = 1$. W przeciwnym przypadku $C_{it} = 0$. Zadbamy o poprawność tego grafiku w ograniczeniach.

2.1.2 Ograniczenia

1. **Nie powtarzaj się** - każde zadanie może zostać rozpoczęte wyłącznie raz.

$$(\forall i \in \{1, \dots, n\}) \left(\sum_{t=1}^T C_{it} \right) = 1$$

2. **Cierpliwość jest cnotą** - nie zaczynaj zadania jeśli nie jest gotowe.

$$(\forall i \in \{1, \dots, n\}) \left(\sum_{t=1}^T C_{it} \cdot t \right) \geq r_i$$

3. **Skup się na jednym** - maszyna nie może przetwarzać naraz więcej niż jednego zadania.

$$(\forall t \in \{1, \dots, T\}) \left(\sum_{i=1}^n \sum_{s=\max(1, t-p_i+1)}^t C_{is} \right) \leq 1$$

2.1.3 Funkcja celu

Chcemy zminimalizować ważone opóźnienie wykonania wszystkich zadań:

$$\min \sum_{i=1}^n \sum_{t=1}^T w_i \cdot C_{it} \cdot (t + p_i)$$

2.2 Wyniki

Dla przykładowych danych $n = 5$, $p = [3; 2; 4; 5; 1]$, $r = [2; 1; 3; 1; 0]$, $w = [5, 1, 5, 6, 1]$ mamy następujący grafik:

Czas	0-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16
M1	5	4				1			3				2			

z funkcją celu o wartości 162.

3 Zadanie 3

3.1 Model

Mamy daną liczbę zadań $n \in \mathbb{Z}_+$, liczbę maszyn $m \in \mathbb{Z}_+$, czasy wykonywania zadań $p \in \mathbb{N}^n$, oraz macierz reprezentującą graf relacji poprzedzania się zadań $r \in \{0, 1\}^{n \times n}$, gdzie $r_{ij} = 1$ wtedy i tylko wtedy, gdy zadanie i musi się zakończyć przed rozpoczęciem zadania j . Ponadto, jak w poprzednim zadaniu użyjemy *horyzontu*, czyli $T = 1 + \sum_{i=1}^n p_i$.

3.1.1 Zmienne decyzyjne

Główną zmienną decyzyjną będzie tabela czasu rozpoczęć zadań na poszczególnych maszynach $S \in \{0,1\}^{n \times m \times T}$. Użyjemy również zmiennej pomocniczej $c \in \mathbb{N}$, która pomoże w wyznaczeniu maksymalnego opóźnienia.

3.1.2 Ograniczenia

1. **Definicja maksimum** - zmienna pomocnicza jest ograniczeniem górnym opóźnienia na każdej maszynie.

$$(\forall i \in \{1, \dots, n\}) \left(\sum_{j=1}^m \sum_{t=1}^T (t + p_i) \cdot S_{ijt} \right) \leq c$$

2. **Nie powtarzaj się** - każde zadanie może zostać rozpoczęte wyłącznie raz.

$$(\forall i \in \{1, \dots, n\}) \left(\sum_{j=1}^m \sum_{t=1}^T S_{ijt} \right) = 1$$

3. **Skup się na jednym** - żadna maszyna nie może przetwarzać naraz więcej niż jednego zadania.

$$(\forall t \in \{1, \dots, T\}, j \in \{1, \dots, T\}) \left(\sum_{i=1}^n \sum_{s=\max(1, t-p_i+1)}^t S_{ijs} \right) \leq 1$$

4. **Uszanuj pierwszeństwo** - zadania poprzedzające inne zadania muszą się wcześniej skończyć.

$$(\forall i_1, i_2 \in \{1, \dots, n\}) r_{i_1 i_2} \left(\sum_{j=1}^m \sum_{t=1}^T t \cdot S_{i_2 j t} - (t + p_i) \cdot S_{i_1 j t} \right) \geq 0$$

3.1.3 Funkcja celu

Chcemy zminimalizować maksymalne opóźnienie reprezentowane przez c :

$$\min c$$

3.2 Wyniki

Dla danych $n = 9, m = 3, p = [1; 2; 1; 2; 1; 1; 3; 6; 2], r = [(1, 4); (2, 4); (2, 5); (3, 4); (3, 5); (4, 6); (4, 7); (5, 7); (5, 8); (6, 9); (7, 9)]$ program zwrócił wartość funkcji celu $c = 9$ oraz następujący grafik:

t	0-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10
M1	1	3	4		7			9		
M2	2				6					
M3			5		8					

Grafik ten spełnia założenia relacji poprzedzania oraz, mimo innego układu zadań, ma to samo opóźnienie co przykład podany w poleceniu.

4 Zadanie 4

4.1 Model

Mamy dane: liczbę odnawialnych zasobów $p \in \mathbb{N}$, limity przechowywania zasobów $N \in \mathbb{Z}_+$, liczbę czynności $n \in \mathbb{Z}_+$, czasy wykonywania czynności $t \in \mathbb{N}^n$, wektory zapotrzebowań czynności na zasoby $r \in \mathbb{N}^{n \times p}$, oraz macierz reprezentującą graf relacji poprzedzania czynności $g \in \{0,1\}^{n \times n}$, gdzie $g_{ij} = 1$ wtedy i tylko wtedy, gdy zadanie i musi się zakończyć przed rozpoczęciem zadania j . Tutaj znowu skorzystamy z *horyzontu*: $T = 1 + \sum_{i=1}^n t_i$

4.1.1 Zmienne decyzyjne

Główną zmienną decyzyjną będzie tabela rozpoczęć czynności: $S \in \{0, 1\}^{n \times T}$. Użyjemy również zmiennej pomocniczej $c \in \mathbb{N}$, która pomoże w wyznaczeniu maksymalnego opóźnienia.

4.1.2 Ograniczenia

1. **Definicja maksimum** - zmienna pomocnicza jest ograniczeniem górnym opóźnienia na każdej maszynie.

$$(\forall i \in \{1, \dots, n\}) \left(\sum_{u=1}^T (u - 1 + t_i) \cdot S_{iu} \right) \leq c$$

2. **Nie powtarzaj się** - każde zadanie może zostać rozpoczęte wyłącznie raz.

$$(\forall i \in \{1, \dots, n\}) \left(\sum_{u=1}^T S_{iu} \right) = 1$$

3. **Uszanuj pierwszeństwo** - zadania poprzedzające inne zadania muszą się wcześniej skończyć.

$$(\forall i_1, i_2 \in \{1, \dots, n\})_{g_{i_1 i_2}} \left(\sum_{u=1}^T (u - 1) \cdot S_{i_2 u} - (u - 1 + t_{i_1}) \cdot S_{i_1 u} \right) \geq 0$$

4. **Ograniczony budżet** - w danej chwili liczba wykorzystywanych zasobów nie może przekroczyć limitu.

$$(\forall e \in \{1, \dots, p\}, u \in \{1, \dots, T\}) \left(\sum_{i=1}^n \left(\sum_{s=\max(1, u-t_i+1)}^u S_{is} \right) \cdot r_{ie} \right) \leq N_e$$

4.1.3 Funkcja celu

Chcemy zminimalizować maksymalne opóźnienie reprezentowane przez c :

$$\min c$$

4.2 Wyniki

Dane z polecenia w zadaniu: $p = 1$, $N = [30]$, $n = 8$, $t = [50; 47; 55; 46; 32; 57; 15; 62]$, $r = [[9; 17; 11; 4; 13; 7; 7; 17]]$, $g = [(1, 2); (1, 3); (1, 4); (2, 5); (3, 6); (4, 6); (4, 7); (5, 8); (6, 8); (7, 8)]$.

Wynikiem obliczeń dla tych danych jest poniżej przedstawiony grafik otrzymany w 24.4256 sekund z wartością funkcji celu 237:

Czynność	Start	Finisz
1	0	50
2	96	143
3	52	107
4	50	96
5	143	175
6	115	172
7	143	158
8	175	237

i rzeczywiście, w żadnej chwili nie ma przeciążenia zasobu ponad limit.