Metody Optymalizacji
Laboratorium 2

Adrian Herda

2025-05-08

1. Zadanie 1

Celem modelu jest minimalizacja resztek pozostawionych po wycinaniu desek w tartaku. Tartak tnie deski o długości 22 cali na kawałki o długości $7, 5, 3$ cale.

1.1. Dane

- W - standardowa szerokość deski (w calach),
- w_i - możliwe szerokości desek po cięciu (w calach) dla $i \in \{1, 2, 3\}$,
- d_i - zapotrzebowanie na deski o odpowiednich szerokościach,
- P - zbiór możliwych kombinacji szerokości desek po cięciu jednej deski wraz z resztą pozstałą po cięciu. Każda kombinacja to czwórka (a_1, a_2, a_3, r), gdzie a_i to liczba desek o szerokości w_i pozyskanych z takiego podziału deski, a r to reszta pozostała po cięciu.

1.1.1. Egzemplarz

- $W = 22$,
- $w = [1, 2, 3]$,
- $d = [110, 120, 80]$

1.2. Model

1.2.1. Zmienne decyzyjne

- $x_p \in \mathbb{N} \cup \{0\}$ - liczba desek wyciętych według kombinacji $p \in P$

1.2.2. Ograniczenia

- Ograniczenie na zapotrzebowanie deski o szerokości w_i:

$$\sum_{p \in P} x_p \cdot a_{ip} \geq d_i \text{ , dla } i \in \{1, 2, 3\}$$

gdzie a_{ip} to liczba desek o szerokości w_i w kombinacji p.

1.2.3. Funkcja celu

Funkcja celu minimalizuje resztę pozostałą po cięciu oraz nadmiarowe deski wycięte podczas cięć, równowaażnie można powiedzieć że minimalizuje liczbę użytych desek:

$$\min \sum_{p \in P} x_p$$

1.3. Rozwiązanie

Wykorzustując solver Cbc oraz bibliotekę JuMP w języku Julia, otrzymujemy następujące rozwiązanie:

1.3.1. Funkcja celu

- Ilość wykorzystanych desek: 74,
- Ilość niewykorzystanego materiału (w calach): 18

1.3.2. Wybrane kombinacje

- Wzór $(1, 3, 0, 0) \times 28$
- Wzór $(2, 1, 1, 0) \times 37$
- Wzór $(1, 0, 5, 0) \times 9$

1.4. Wyprodukowane deski

- Ilość desek o szerokości 7 cali: 111
- Ilość desek o szerokości 5 cali: 121
- Ilość desek o szerokości 3 cali: 82

1.5. Podusmowanie

Algorytm wykorzystał tylko kombinacje nie zostawiające reszty po cięciu i zminimalizował ilość nadmiarowych desek.

2. Zadanie 2

Celem modelu jest znalezienie harmonogramu dla pojedynczej maszyny pozwalającego na zminimalizowanie kosztów wykonywania zadań. Koszty wykonywania zadań są obliczane na podstawie czasu ich realizacji oraz ich wagi.

2.1. Dane

- n - liczba zadań,
- p_j - czas potrzebny na wykonanie zadania $j \in [n]$,
- w_j - waga zadania $j \in [n]$,
- r_j - czas najwcześniejszego możliwego rozpoczęcia zadania $j \in [n]$,
- $T = \sum_{j \in [n]} p_j + \max_{j \in [n]} \{r_j\} + 1$ - dodatkowy parametr pomocniczy, który oznacza maksymalny potrzebny czas na wykonanie wszystkich zadań

2.1.1. Wybrany egzemplarz

j - Zadanie	1	2	3	4	5	6	7	8	9	10
p_j - Czas wykonywania	3	2	7	5	6	4	8	3	9	2
w_j - Waga	2	4	1	3	2	5	1	2	4	3
r_j - Czas możliwego rozpoczęcia	0	1	3	5	0	2	6	4	3	0

2.2. Model

2.2.1. Zmienne decyzyjne

- $x_{jt} \in \{0, 1\}$, dla $j \in [n], T]$

  $x_{jt} = \begin{cases} 1 : \text{zadanie } j \text{ ma ostanią jednostkę czasu działania w momencie } t \\ 0 : \text{w przeciwnym przypadku} \end{cases}$

2.2.2. Ograniczenia

- Każde zadanie może być wykonywane tylko raz, czyli może się zakończyć tylko raz:

$$\sum_{(T]}) x_{jt} = 1, \text{ dla } j \in [n]$$

- Każde zadanie może się zaczynać nie wcześniej niż podane w odpowiadającym parametrze r:

$$\sum_{(T]}) x_{jt} \big(t - p_j\big) \geq r_j, \text{ dla } j \in [n]$$

- W dowolnym momencie czasu wykonywane jest maksymalnie jedno zadanie

$$\sum_{j \in [n]} \left(\sum_{t'=t}^{\min\{T, t-1+p_j\}} x_{jt'} \right) \leq 1, \text{ dla } T]$$

2.2.3. Funkcja celu

Funkcja celu minimalizuje sumę ważonych czasów zakończenia zadań:

$$\min \sum_{j \in [n]} w_j \cdot \sum_{(T]})\big(t \cdot x_{jt}\big)$$

gdzie $\sum_{t=1}^{T}\big(t \cdot x_{jt}\big)$ interpretować można jako czas zakończenia zadania j.

2.3. Rozwiązanie

Wykorzustując solver Cbc oraz bibliotekę JuMP w języku Julia, otrzymujemy optymalne rozwiązanie. Wyniki zapisywane są z czasem zaczynającym się od 0-wej sekundy podczas gdy liczone są zaczynając się do 1-ej sekundy

2.3.1. Funkcja celu

$$\sum_{j \in [n]} w_j \cdot \sum_{(T]})\big(t \cdot x_{jt}\big) = 439$$

2.3.2. Momenty zakończeń

Zadanie	Moment zakończenia
1	13
2	3
3	40
4	18
5	33
6	7
7	48
8	10
9	27
10	1

3. Zadanie 3

Celem modelu z tego zadania jest stworzenie harmonogramu dla określonej liczby maszyn tak aby wykonać zadania przeznaczone dla tych maszynw jak najszybszym czasie. Zadania mają czasy wykonywania i ograniczenia poprzedzalności których trzeba przeszkadzać.

3.1. Dane

- m - liczba maszyn,
- n - liczba zadań,
- p_j - czasy wykonywania
- $R = \{(a, b) : a \prec b\}$ - zbiór relacji poprzedzania gdzie $a \prec b$ oznacza ze zadanie b nie może się zacząć dopóki a się nie zakończy
- $T = 1 + \sum_{j \in [n]} p_j$ - dodatkowy parametr pomocniczy oznaczający maksymalny czas w jakim wszystkie zadania mogłyby zostać wykonane przez jedna maszynę

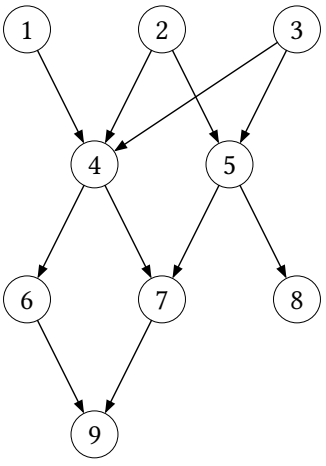
3.1.1. Egzemplarz

- $m = 3$
- $n = 9$
-

Zadanie j	p_j
1	1
2	2
3	1
4	2
5	1
6	1
7	3
8	6
9	2

- $R = \{(1, 4), (2, 4), (2, 5), (3, 4), (3, 5), (4, 6), (4, 7), (5, 7), (5, 8), (6, 9), (7, 9)\}$

3.1.2. Diagram poprzedzania



3.2. Model

3.2.1. Zmienne decyzyjne

- $x_{jti} \in \{0, 1\}$, dla $j \in [n], T], i \in [m]$

  $x_{jti} = \begin{cases} 1 : \text{zadanie } j \text{ ma rozpoczyna swoje działanie w momencie } t \text{ na maszynie } i \\ 0 : \text{w przeciwnym wypadku} \end{cases}$

- $C \geq 0$ - minimalny czas potrzebny na wykonanie wszystkich zadań

3.2.2. Ograniczenia

- Każde zadanie kończy się nie później niż C

$$\sum_{(T]})x_{jti}(t - 1 + p_j) \leq C, \text{ dla } j \in [n] \text{ oraz } i \in [m]$$

- Każde zadanie musi być wykonane, a więc i rozpoczęte, dokładnie raz

$$\sum_{(T]}) \sum_{i \in [m]} x_{jti} = 1, \text{ dla } j \in [n]$$

- Warunki poprzedzania muszą być spełnione - czas zakończenia zadanie a musi być mniejszy niż czas rozpoczęcia zadania b

$$\sum_{(T]}) \sum_{i \in [m]} x_{ati}(t + p_a) \leq \sum_{(T]}) \sum_{i \in [m]} t \cdot x_{bti}, \text{ dla } (a, b) \in R$$

- Zadania nie mogą na siebie nachodzić

$$\sum_{j \in [n]} \left(\sum_{t' = \max\{1, t+1-p_j\}}^{t} x_{jt'm} \right) \leq 1, \text{ dla } T] \text{ oraz } i \in [m]$$

3.2.3. Funkcja celu

Funkcja celu minimalizuje czas potrzebny na wykonanie wszystkich zadań

$$\min C$$

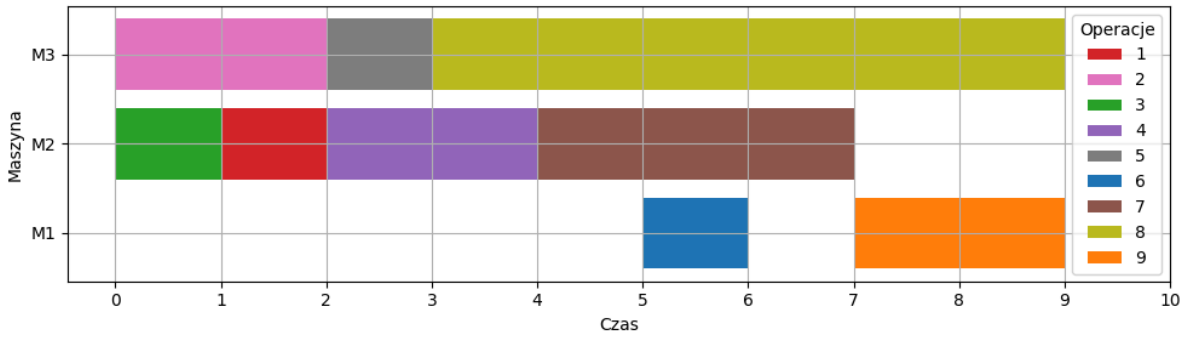
3.3. Rozwiązanie

Wykorzystując solver Cbc oraz bibliotekę JuMP języka Julia do stworzenia modelu, zostało znalezione optymalne rozwiązanie:

3.3.1. Funkcja celu

$$C = 9$$

3.3.2. Harmonogram na diagramie Gantt'a



Rysunek 1: Diagram Gantt'a dla optymalnego rozwiązania $C = 9$

4. Zadanie 4

Zadnie polegało na stworzeniu modelu który znajdzie harmonogram wykonywania zadań przy ograniczonych zasobach nie łamiąc określonych zasad poprzedzania. Każde zadanie ma określony czas trwania, wymaganą ilość zasobu do jego wykonania oraz może zacząć się nie wcześniej niż skończy się jego poprzednik.

4.1. Dane

- P - ilość limitowanych zasobów
- n - ilość zadań do wykonania
- N_p - ilość dostępnego zasobu $p \in [P]$
- δ_j - czas potrzebny do wykonania zadania $j \in [n]$
- r_{jp} - ilość zasobu $p \in [P]$ potrzebna do wykonania zadania $j \in [n]$
- $R = \{(a, b) : a \prec b\}$ - zbiór relacji poprzedzania gdzie $a \prec b$ oznacza ze zadanie b nie może się zacząć dopóki a się nie zakończy
- $T_{\max} = 1 + \sum_{j \in [n]} \delta_j$ - dodatkowy parametr pomocniczy oznaczający maksymalny czas w jakim wszystkie zadania mogłyby zostać wykonane, wykonując wszytskie po kolei i nie dzieląc zasobów pomiędzy zadania

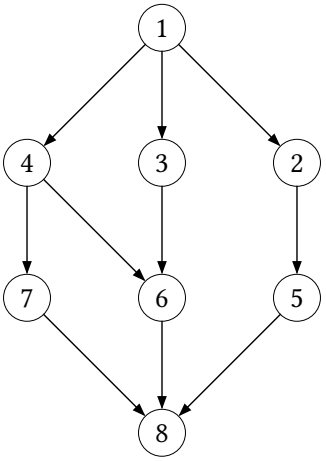
4.1.1. Egzemplarz

- $P = 1$
- $n = 8$
- $N = [30]$
-

Zadanie j	1	2	3	4	5	6	7	8
Czas wykonania δ_j	50	47	55	46	32	57	15	62
Potrzebne zasoby r_j	[9]	[17]	[11]	[4]	[13]	[7]	[7]	[17]

- $R = \{(1,2), (1,3), (1,4), (2,5), (3,6), (4,6), (4,7), (5,8), (6,8), (7,8)\}$

4.1.2. Diagram poprzedzania



4.2. Model

4.2.1. Zmienne decyzyjne

- $x_{jt} \in \{0, 1\}$, dla $j \in [n], T_{\max}]$

  $x_{jti} = \begin{cases} 1 : \text{zadanie } j \text{ ma rozpoczyna swoje działanie w momencie } t \\ 0 : \text{w przeciwnym wypadku} \end{cases}$

- $C \geq 0$ - minimalny czas potrzebny na wykonanie wszystkich zadań

4.2.2. Ograniczenia

- Każde zadanie kończy się nie później niż C

$$\sum_{(T_{\max}]})x_{jt}\big(t - 1 + \delta_j\big) \leq C, \text{ dla } j \in [n]$$

- Każde zadanie musi być wykonane, a więc i rozpoczęte, dokładnie raz

$$\sum_{(T_{\max}]})x_{jt} = 1, \text{ dla } j \in [n]$$

- Warunki poprzedzania muszą być spełnione - czas zakończenia zadanie a musi być mniejszy niż czas rozpoczęcia zadania b

$$\sum_{(T_{\max}]})x_{at}(t + \delta_a) \leq \sum_{(T_{\max}]})t \cdot x_{bt}, \text{ dla } (a, b) \in R$$

- Zadania nie mogą przekraczać limitu zasobów w żadnym momencie czasu

$$\sum_{j \in [n]} \left(\sum_{t' = \max\{1, t+1-\delta_j\}}^{t} \big(x_{jt'} \cdot r_{jp}\big) \right) \leq N_p, \text{ dla } T_{\max}] \text{ oraz } p \in [P]$$

4.2.3. Funkcja celu

Funkcja celu minimalizuje czas potrzebny na wykonanie wszystkich zadań

$$\min C$$

4.3. Rozwiązanie

Wykorzustując solver Cbc oraz bibliotekę JuMP w języku Julia, otrzymujemy następujące optymalne rozwiązanie:

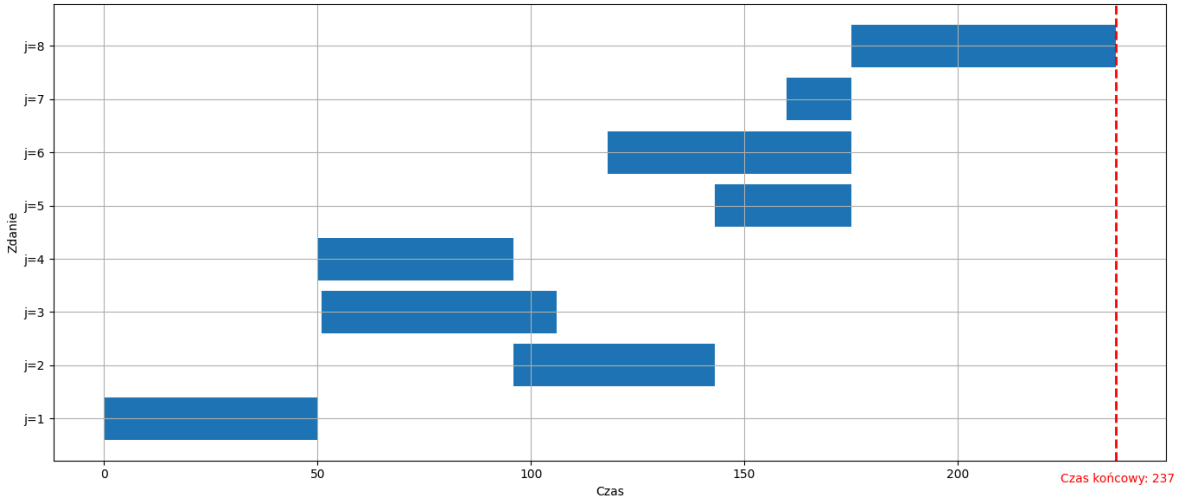
4.3.1. Funkcja celu

$$C = 237$$

4.3.2. Starty i zakończenia zadań

Zadanie	Start	Koniec
1	1	50
2	97	143
3	52	106
4	51	96
5	144	175
6	119	175
7	161	175
8	176	237

4.3.3. Harmonogram na diagramie Gantt'a



Rysunek 2: Diagram Gantt'a dla optymalnego rozwiązania $C = 237$

4.3.4. Wykorzystanie zasobów

Start (włącznie)	Koniec (wyłącznie)	Wykorzystanie zasobów
1	51	$\frac{9}{30}$
51	52	$\frac{4}{30}$

Start (włącznie)	Koniec (wyłącznie)	Wykorzystanie zasobów
52	97	$\frac{15}{30}$
97	107	$\frac{28}{30}$
107	119	$\frac{17}{30}$
119	144	$\frac{24}{30}$
144	161	$\frac{20}{30}$
161	176	$\frac{27}{30}$
176	238	$\frac{17}{30}$