

WYDZIAŁ PODSTAWOWYCH PROBLEMÓW TECHNIKI
POLITECHNIKA WROCŁAWSKA

METODY OPTYMALIZACJI

LISTA 2

KAMIL SIKORSKI
NR INDEKSU: 221481

Przedmiot prowadzony przez

Pawła Zielińskiego



Politechnika
Wrocławska

WROCŁAW 2019

Spis treści

1	Zadanie 1	1
1.1	Opis problemu	1
1.2	Rozwiązanie	1
1.3	Wyniki i interpretacja	1
2	Zadanie 2	2
2.1	Opis problemu	2
2.2	Rozwiązanie	2
2.3	Wyniki i interpretacja	2
3	Zadanie 3	4
3.1	Opis problemu	4
3.2	Rozwiązanie	4
3.3	Wyniki	4

Zadanie 1

1.1 Opis problemu

Zadanie polega na zminimalizowaniu ilości odpadów w tartaku. Firma otrzymuje zamówienie na deski różnej szerokości, które są wycinane z desek o standardowej szerokości. Odpadem nazywane są deski, które przekraczają ilość zapotrzebowania oraz takie które nie mają odpowiedniego rozmiaru. Danymi do zadania są:

- rozmiar deski standardowej (22 cali),
- zamówienia na deski o rozmiarach 7, 5, 3 cali (110 o szerokości 7 cali, 120 o szerokości 5 cali oraz 80 o szerokości 3 cali).

1.2 Rozwiązanie

Do rozwiązania zadania napisano program z parametrami:

- szerokość standardowej deski
- zapotrzebowanie na deski o szerokości 3 cali
- zapotrzebowanie na deski o szerokości 5 cali
- zapotrzebowanie na deski o szerokości 7 cali

Program wyznacza możliwe sposoby podzielenia jednej deski na 3,5,7 calowe deski oraz odpad. Następnie minimalizuje odpad.

1.3 Wyniki i interpretacja

Plan podziału desek przedstawiony jest w tabeli 1.1. Łączna suma nadmiaru to 18 cali.

ilość	7cal	5cal	3cal	odpad
9	1	0	5	0
28	1	3	0	0
37	2	1	1	0
suma	111	121	82	0

Tabela 1.1: Podział desek

Zadanie 2

2.1 Opis problemu

Zadanie polega na zminimalizowaniu kosztu wykonania zadań $J = \{1, 2, \dots, n\}$. Każde zadanie posiada czas wykonania p_j , wagę zadania w_j oraz moment gotowości (moment przed który zadanie nie może być wykonane) r_j . Funkcja kosztu: $\sum_{j \in J} w_j C_j$, gdzie C_j jest czasem zakończenia zadania j .

2.2 Rozwiązanie

Do rozwiązania problemu stworzono program z parametrami:

- p_j gdzie $j \in J$ opisuje czas wykonania zadania;
- w_j gdzie $j \in J$ opisuje wagę zadania;
- r_j gdzie $j \in J$ opisuje moment dostępności zadania;

Model rozwiązujący posiada:

- Zmienną $x_{j,h}$ gdzie $j \in J, h \in Czas$ opisującą czy zadanie j uruchomi się w momencie h ;
- Ogarniczenie $\forall j \in J \sum_{h \in Czas} x[j, h] == 1$ ogranicza dokładnie jedno uruchomienie każdego zadania;
- Ogarniczenie $\forall j \in J \sum_{h \in Czas} x[j, h] \geq r_j$ ogranicza moment dostępności zadania;
- Ogarniczenie $\forall h \in Czas \sum_{j \in J} x[j, h] \leq 1$ ogranicza nakładanie się zadań w tym samym czasie.

2.3 Wyniki i interpretacja

W celu sprawdzenia rozwiązania uruchomiono program z parametrami podanych w tabeli 2.1.

Zadanie	Czas	dostępność	waga
1	1	5	50
2	2	0	1
3	3	0	2
4	4	0	3
5	5	0	5

Tabela 2.1: Przykładowe dane

Koszt rozwiązania dla danego problemu wyniósł 414, co zaprezentowane jest w tabeli 2.2.

Czas	Zadanie
1	4
5	1
6	5
11	3
14	2

Tabela 2.2: Rozwiązanie

Zadanie 3

3.1 Opis problemu

Zadanie polega na rozwiązaniu problemu minimalizacji czasu wykonywania zadań $J = \{1, 2, \dots, n\}$ na m maszynach. Dla każdego zadania $i \in J$ dany jest czas potrzebny do wykonania p_i . Zbiór zadań jest uporządkowany za pomocą relacji poprzedzenia (zadanie i może zostać wykonane po wykonaniu zadań j_1, j_2).

3.2 Rozwiązanie

Do rozwiązania problemu stworzono program z parametrami:

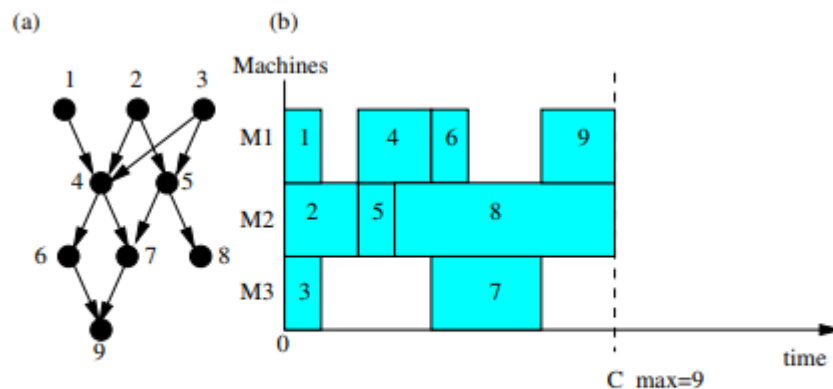
- p_i gdzie $i \in J$ czas wykonania zadania i ;
- r_i gdzie $i \in J$ zadania poprzedzające zadanie i ;
- m ilość maszyn

Model rozwiązujący posiada:

- Zmienną $x_{i,h,m}$ gdzie $i \in J, h \in Czas, m \in Maszyny$ opisującą czy zadanie i uruchomi się w momencie h na maszynie m ;
- Zmienną ms opisującą zakończenie ostatniego zadania;
- Ograniczenie $\forall i \in J \sum_{m \in Maszyny} \sum_{h \in Czas} x[i, h, m] = r_j$ ogranicza uruchomienie każdego zadania tylko raz;
- Ograniczenie $\forall_{m \in Maszyny} \sum_{h \in Czas} \sum_{i \in J} x[i, h, m] \leq 1$ ogranicza nakładanie się zadań w tym samym czasie na tej samej maszynie;
- Ograniczenie $\forall i \in J (\forall j \in r_i \sum_{m \in Maszyny} \sum_{h \in Czas} x[j, h, m] * (h + p[j]) \leq \sum_{m \in Maszyny} \sum_{h \in Czas} x[i, h, m] * h)$ ogranicza wykonywanie zadań po sobie;
- Ograniczenie $\forall_{m \in Maszyny} \sum_{i \in J} \sum_{h \in Czas} x[i, h, m] * (h + p[i]) \leq ms$ ograniczającą zakończenie się ostatniego zadania;
- Funkcję celu $min \rightarrow ms$.

3.3 Wyniki

Rozwiązanie przykładu podanego w zadaniu 3.1. Rozwiązanie zostało zaprezentowane w tabeli 3.1



Rysunek 1: (a) Relacje poprzedzania. (b) Wizualizacja dopuszczalnego harmonogramu z całkowitym czasem potrzebnym do wykonania wszystkich zadań równym 9.

Rysunek 3.1: Dane i rozwiązanie

maszyna	czas		
	1	2	3
1	1	-	2
2	-	3	2
3	-	5	4
4	-	8	4
5	-	8	7
6	-	8	7
7	6	8	7
8	-	8	9
9	-	8	9

Tabela 3.1: Wyniki rozwiązania