Metody optymalizacji Lista 2

Radosław Wojtczak, numer indeksu: 254607

07.05.2023

Spis treści

Zadanie 1

1.1 Wprowadzenie

Zadanie to przedstawiało tartak, który mając podaną szerokość deski określoną w tej pracy przez zmienną *size* mógł dokonać jej podziału na mniejsze części, odgórnie zdefiniowane przy pomocy zmiennej *cuts*. Celem tartaku było zaspokojenie zapotrzebowań na podziały określone w pracy przez zmienną *demands* tak, aby odpad (czyli pozostałość po oryginalnej desce, której nie da się przedstawić poprzez podział na mniejsze części) były jak najmniejsze. W treści zadania zmienne otrzymały odpowiednie wartości:

- size = 22
- $cuts = \{3, 5, 7\}$
- $demand = \{80, 110, 120\}$

1.2 Model

Model został zaprojektowany w sposób następujacy: dla zadanego rozmiaru oraz możliwości cięć generowane są wszystkie kombinacje cięć. Wygenerowane schematy przetrzymywane są w specjalnej macierzy z dodatkową kolumną nazywaną **odpadem**, która wyliczana jest jako rożnica rozmiaru oryginalnej deski i sumy rozmiarów cięć pomniejszych desek. Tak spreparowana macierz pozwala nam w prosty sposób wyłuskiwać istotne dla nas informacje z persepktywy rozpatrywanego zadania.

1.2.1 Zmienne decyzyjne

$$x_i : i \in 1, ..., n \tag{1.1}$$

gdzie n oznacza liczbę wszystkich permutacji spełniających założenia zadania.

1.2.2 Ograniczenia

- Wszystkie zapotrzebowania muszą zostać zaspokojone

$$\forall_{j \in cuts}(\sum_{a=1}^{n} conf_{a,j} * x_a) \geq demand_j \tag{1.2}$$

gdzie **conf** jest wcześniej opisywaną macierzą możliwych *konfiguracji* cięć.

1.2.3 Funkcja celu

$$(\sum_{a=1}^{n} conf_{a,o} * x_a) + \sum_{j \in cuts} ((\sum_{a=1}^{n} conf_{a,j} * x_a) - d_j) * cuts_j \tag{1.3}$$

gdzie indeks o oznacza kolumnę zawierająca informację o odpadzie. Funkcję celu można podzielić na dwa etapy. W pierwszym wyliczany jest odpad, a w drugim-nadwyżka wyprodukowanych desek. Ostatecznie dążymy do minimalizacji sumy dwóch wyodrębnionych możliwości.

1.3 Wyniki

Po uruchomieniu algorytmu dla zadanych w zadaniu paremtrów otrzymano następującą odpowiedź: **Odpad: 18**. Ponadto poinformował użytkownika o podziale oryginalnej deski:

- 37 podziałów typu $[7, 7, 5, 3]$
- 28 podziałów typu $[7, 5, 5, 5]$
- 9 podziałów typu $[7, 3, 3, 3, 3, 3]$

Zadanie 2

2.1 Wprowadzenie

Celem zadania jest znalezienie odpowiedniego harmonogramu, według którego należy wykonać zadania na jednej maszynie. Problem ów jest znany w literaturze pod nazwą "single machine scheduling". Dokładniej formuując problem, mamy zadany zbiór $J = \{1, .., n\}$ składający się z n zadań. Dla każdego zadania $j \in J$ definiowane są następujące własności:

- Czas potrzebny na wykonanie zadania (p_j)
- Priorytet zadania, określany też jako waga (w_j)
- Moment gotowości zadania (r_j)

Celem programu jest utworzenie harmonogramu minimalizującego iloczyn czasu zakończenia danego zadania oraz jego priorytetu.

2.2 Model

W ramach utworzenia modelu zdefiniowano następujące zmienne:

- (max_T = maximum(r) + sum(p) + 1) - maksymalnie czas pracy maszyny
- jobs = $\{1, ..., n\}$ - wektor przechowujący indeksy poszczególnych prac
- times = $\{1, ..., max\_T\}$ - wektor przechowujący stany maszyny w danych momentach

To na co należy zwrócić uwagę to fakt, że w ramach rozpatrywanego zadania traktujemy czas w sposób dyskretny. Zdefiniowawszy wszystkie zmienne pomocnicze poniżej przedstawiono definijcę zmiennych decyzyjnych.

2.2.1 Zmienne decyzyjne

$$x_{jt} : j \in jobs, t \in times, x_{jt} \in \{0, 1\} \tag{2.1}$$

Zmienną decyzyjną jest macierz przechowująca momenty rozpoczęcia każdego z zadań. Rozpoczęcie zadania oznaczane jest przy pomocy liczby **1**, pozostałe wartości są zerowane.

2.2.2 Ograniczenia

- Każda praca może mieć tylko jeden moment rozpoczęcia

$$(\forall_{j \in jobs})(\sum_{t \in times} x_{jt} = 1) \tag{2.2}$$

- j-te zadanie może rozpocząć nie wcześniej, niż wskazuje na to wartość r_j

$$(\forall_{j \in jobs})(\sum_{t \in times} x_{jt} * t \geq r_j) \tag{2.3}$$

- Maksymalnie jedno zadanie może być wykonywane w danej jednostce czasu

$$(\forall_{t \in times})(\sum_{j \in jobs, s \in max\{0, t-p_j+1\}, t} x_{js} <= 1) \tag{2.4}$$

2.2.3 Funkcja celu

Funkcją celu jest minimalizacja iloczynu czasu zakończenia danego zadania oraz jego priorytetu.

$$\sum_{t \in times, j \in jobs} w_j * (t + p_j) * x_{jt} \tag{2.5}$$

2.3 Wyniki

Zaimplementowany model został zaimplementowany dla następujących danych:

- $n = 6$
- $p = [2, 3, 4, 1, 3, 2]$
- $w = [3, 2, 1, 4, 4, 6]$
- $r = [2, 1, 1, 0, 0, 4]$

Wyniki prezentują się w sposób następujący:

Zadanie	Czas rozpoczęcia
1	6
2	8
3	11
4	0
5	1
6	4

Table 2.1: Numer zadania wraz z momentem rozpoczęcia dla wybranego egzemplarza problemu

Ponadto funkcja celu została obliczona i jej wartość wynosi **117**.

Zadanie 3

3.1 Wprowadzenie

Zadanie to jest naturalnym rozszerzeniem poprzedniego zadania. Tym razem mając zbiór zadań $J = \{1, .., n\}$ składający się z n zadań, tworząc harmonogram mamy do dyspozycji m maszyn. Ponadto zrezygnowano z wag oraz momentu gotowości zadań na rzecz relacji pierwszeństwa. Jeśli dwa zadania (i, j) są ze sobą w relacji to zadanie j nie może się rozpocząć przed ukończeniem zadania i. Zgodnie z treścią zadania, relacja ta jest oznaczana przy pomocy symbolu $\rightarrow$ (i $\rightarrow$ j). Celem zadania jest znalezienie dopuszczalnego harmonogramu, który minimalizuje całkowity czas potrzebny do wykonania wszystkich zadań. Czas ten oznaczono jako c_max.

3.2 Model

W celu wykonania zadania, podobnie jak w poprzednim zadaniu, wprowadzono dodatkowe zmiennej:

- (max_T = sum(p) + 1) - maksymalnie czas pracy maszyny
- jobs = $\{1, ..., n\}$ - wektor przechowujący indeksy poszczególnych prac
- times = $\{1, ..., max\_T\}$ - wektor przechowujący stany maszyny w danych momentach

3.2.1 Zmienne decyzyjne

$$x_{jt} : j \in jobs, t \in times, x_{jt} \in \{0, 1\} \tag{3.1}$$

Zmienną decyzyjną jest macierz przechowująca momenty rozpoczęcia każdego z zadań. Rozpoczęcie zadania oznaczane jest przy pomocy liczby **1**, pozostałe wartości są zerowane. Ponadto zmienną decyzyjną jest zmienna $c\_max \geq 0$ informująca o czasie niezbędnym do wykoniania wszystkich zadań.

3.2.2 Ograniczenia

- Wymagany czas nie może być mniejszy niż suma rozpoczęcia ostatniej pracy na maszynie oraz czasu jej trwania

$$(\forall_{j \in jobs})(\sum_{t \in times} x_{jt} * (p_j + t)) <= c\_max \tag{3.2}$$

- Każda praca może mieć tylko jeden moment rozpoczęcia

$$(\forall_{j \in jobs})(\sum_{t \in times} x_{jt} = 1) \tag{3.3}$$

- Zachowanie relacji pierwszeństwa między zadaniami

$$(\forall_{(j,relation) \in precedence})(\forall_{k \in relation})(\sum_{t \in times} (p_j + t) * x_{jt} \leq \sum_{t \in times} t * x_{kt}) \tag{3.4}$$

- W jednej jednostce czasu nie może być więcej aktywny zadań niż dostepnych maszyn

$$(\forall t \in times)(\sum_{j \in jobs} \sum_{s=max\{0,t-p_j+1\}}^{t} x_{js}) <= m \tag{3.5}$$

3.2.3 Funkcja celu

Funkcją celu jest minimalizacja zmiennej c_max.

$$minc_max \tag{3.6}$$

3.3 Wyniki

Działanie programu zostało przetestowane dla następujących danych zaczerpniętych z treści zadania

- n = 9
- m = 3
- p = [1, 2, 1, 2, 1, 1, 3, 6, 2]
- precedence = 1 → [4], 2 → [4,5], 3 → [4,5], 4 → [6,7], 5 → [7,8], 6 → [9], 7 → [9]

Poniżej przedstawiono wyniki w formie tabeli:

Maszyna	Slot1	Slot2	Slot3	Slot4	Slot5	Slot6	Slot7	Slot8	Slot9
1	3	1	4	4	0	0	6	9	9
2	2	2	5	0	7	7	7	0	0
3	0	0	0	8	8	8	8	8	8

Table 3.1: Maszyna oraz zadanie, które jest na niej wykonywane w danym slocie czasu. 0 oznacza brak zadania