Metody Optymalizacji
Lista 2

Dominik Kaczmarek, nr albumu 261757

5 maja 2024

1 Zadanie

1.1 Opis problemu

Załóżmy, że chcemy zebrać dane liczbowe na temat m różnych cech populacji. Ogromna ilość zebranej informacji pamiętana jest w chmurze w n różnych miejscach (serwerach). Niech T_j oznacza czas potrzebny na przeszukanie j-tego miejsca, $j = 1, \ldots, n$, przy czym zakładamy, że nie zależy on od liczby cech, których charakterystyki liczbowe zamierza się w danym momencie odczytać. Dane dotyczące niektórych cech zapisane są w więcej niż jednym miejscu, tzn. niektóre miejsca zawierają duplikaty informacji. Niech $q_{ij} = 1$ jeśli dane na temat cechy i zapisane są w miejscu j, oraz $q_{ij} = 0$ w przeciwnym przypadku. W ten sposób, np. $q_{13} = q_{18} = q_{19} = 1$ oznacza, że dane na temat cechy 1 zapisane są w miejscach 3, 8 i 9. Wyznaczyć te spośród n miejsc, które należy przeszukać, aby zminimalizować łączny czas odczytania danych dotyczących wszystkich cech.

1.2 Model

1.2.1 Parametry

- n - liczba serwerów,
- m - liczba cech populacji,
- T_j - czas przeszukania serwera $j = 1, \ldots, n$
- q_{ij} - czy cecha i znajduje się na serwerze j

1.2.2 Zmienne decyzyjne

- $x_i \in \{0, 1\}$ - 1 jeśli przeszukujemy serwer i, 0 w.p.p.

1.2.3 Funkcja celu

Minimalizacja łącznego czasu przeszukania serwerów:

$$\min \sum_{j=1}^{n} T_j x_j$$

1.2.4 Ograniczenia

Każda cecha musi być obsługiwana przez co najmniej jeden serwer:

$$\text{s.t.} \quad \sum_{j=1}^{n} x_j q_{ij} \geqslant 1, \quad \forall_{i=1,\ldots,m}$$

2 Zadanie

2.1 Opis problemu

Niech P_{ij} będzie j-tym podprogramem obliczania funkcji i należącym do biblioteki podprogramów ($i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$). Podprogram P_{ij} zajmuje r_{ij} komórek pamięci i potrzebuje na jego wykonanie t_{ij} jednostek czasu. Należy ułożyć program (sekwencyjny) P obliczający zadany zbiór funkcji I, $I \subseteq \{1, \ldots, m\}$. Zatem należy dobrać tak podprogramy P_{ij} wchodzące w skład P, obliczające wszystkie funkcje z I, aby cały program zajmował nie więcej niż M komórek pamięci, a czas jego wykonania był minimalny.

2.2 Model

2.2.1 Parametry

- n - liczba podprogramów,
- m - liczba funkcji,
- t_{ij} - czas wykonania funkcji i podprogramem j ($i = 1, \ldots, m$, $j = 1, \ldots, n$),
- r_{ij} - liczba komórek pamięci zajmowana do obliczenia funkcji i podprogramem j ($i = 1, \ldots, m$, $j = 1, \ldots, n$),
- M - posiadana liczba komórek pamięci

2.2.2 Zmienne decyzyjne

- $x_{ij} \in \{0, 1\}$ - 1 jeśli liczymy funkcje i podprogramem j, 0 w.p.p.

2.2.3 Funkcja celu

Minimalizacja łącznego czasu obliczenia wszystkich funkcji:

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} t_{ij} x_{ij}$$

2.2.4 Ograniczenia

1. Każda funkcja musi zostać wykonana:

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} \geqslant 1, \quad \forall_{i=1,\ldots,m}$$

2. Nie możemy przekroczyć posiadanej pamięci:

$$\text{s.t.} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} r_{ij} x_{ij} \leqslant M,$$

3 Zadanie

3.1 Opis problemu

Dany jest zbiór zadań $Z = \{1, \ldots, n\}$, które mają być wykonywane na trzech procesorach $P1$, $P2$ i $P3$. Zakładamy, że:

1. każdy procesor może wykonywać w danym momencie tylko jedno zadanie,
2. każde zadanie musi być wykonywane najpierw na procesorze $P1$ następnie na procesorze $P2$ i na końcu na procesorze $P3$,
3. kolejność wykonywania zadań na wszystkich trzech procesorach jest taka sama.

Dla każdego zadania $i \in Z$ są zadane czasy trwania a_i, b_i oraz c_i odpowiednio na procesorach $P1$, $P2$ i $P3$. Wszystkie dane są dodatnimi liczbami całkowitymi. Każdy harmonogram jest jednoznacznie określony przez pewną permutację $\pi = (\pi(1), \ldots, \pi(n))$ zadań należących do zbioru Z. Niech $C_\pi(k)$ oznacza czas zakończenia k-go zadania na procesorze $P3$ dla permutacji π. Celem jest wyznaczenie permutacji π takiej, że:

$$C_{\max} = C_\pi(n) \to \min$$

Uogólnić model dla m procesorów.

3.2 Model

3.2.1 Parametry

- n - liczba zadań,
- m - liczba procesorów,
- $Z = \{1, \ldots, n\}$ - zbiór zadań
- $P = \{1, \ldots, m\}$ - zbiór procesorów
- t_{ij} - czas wykonania zadania $i \in Z$ procesorem $j \in P$,
- B - suma czasów t_{ij} $(\sum_i^n \sum_j^m t_{ij})$

3.2.2 Zmienne decyzyjne

- x_{ij} - czas rozpoczęcia wykonywania zadania $i \in Z$ na procesorze $j \in P$,
- y_{ik} - $\begin{cases} 1, & \text{zadanie } k \text{ jest wykonywane po zadaniu } i. \\ 0, & w.p.p \end{cases}$ $\quad \forall_{i,k \in Z, i<k}$

3.2.3 Funkcja celu

Minimalizacja czasu ukończenia wszystkich zadań:

$$\min \max_{i \in Z} \{x_{im} + t_{im}\}$$

3.2.4 Ograniczenia

1. Każde zadanie musi być wykonane po kolei na procesorach $1, 2, 3, \ldots, m$:

$$\text{s.t.} \quad x_{ij} + t_{ij} \leqslant x_{i(j+1)}, \quad \forall_{i \in Z} \forall_{j=1,\ldots,m-1}$$

2. Każdy procesor wykonuje tylko jedno zadanie w jednej chwili:

$$\text{s.t.} \quad x_{kj} + t_{kj} \leqslant x_{ij} + B \cdot y_{ik}, \quad \forall_{j \in P} \forall_{i,k \in Z, i<k}$$

$$\text{s.t.} \quad x_{ij} + t_{ij} \leqslant x_{kj} + B \cdot (1 - y_{ik}), \quad \forall_{j \in P} \forall_{i,k \in Z, i<k}$$

3.3 Przykład

3.3.1 Dane

- $m = 3$
- $n = 7$
- $B = \sum_{i=1}^{7} \sum_{j=1}^{3} t_{ij} = 115$

3.3.2 Wyniki

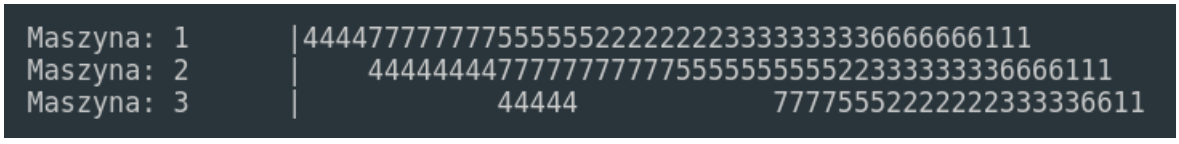
Minimalny czas, po którym ostatnia maszyna skończy ostatnie zadanie to: **51**.

Tabela 1: Czasy wykonywania i-tego zadania na j-tej maszynie (t_{ij})

Zadanie	**Maszyna 1**	**Maszyna 2**	**Maszyna 3**
1	3	3	2
2	9	3	8
3	9	8	5
4	4	8	4
5	6	10	3
6	6	3	1
7	7	10	3

Tabela 2: Czas rozpoczęcia wykonywania zadania i na j-tym procesorze.

Zad/Proc	1	2	3
1	41	46	49
2	17	32	35
3	26	35	43
4	0	4	12
5	11	22	32
6	35	43	48
7	4	12	29



Rysunek 1: Diagram Gantt'a otrzymanego rozwiązania