

Metody optymalizacji

Stepan Yurtsiv, 246437

9 maja 2022r.

1 Zadanie 1

Celem danego zadania jest wyznaczenie serwerów, z których należy odczytać dane o określonych cechach, aby zminimalizować czas.

1.1 Model

Funkcja celu:

$$\min \sum_{j \in [n]} x_j * T_j$$

gdzie

- n - liczba serwerów
- T_j - czas odczytu danych z serwera $j \in [n]$
- x_j - zmienna binarna określająca, czy serwer j powinien zostać przeszukany, $x_j \in \{0, 1\}$, $j \in [n]$

Ograniczenia:

- $\forall i \in [m] \sum_{j \in [n]} Q_{ij} * x_j \geq 1$, gdzie m - liczba cech, a Q_{ij} określa, czy dane cechy i są na serwerze j . Dane ograniczenie zapewnia, że dane każdej cechy zostaną przeczytane co najmniej raz.

1.2 Wyniki

Zdefiniowano następujący egzemplarz problemu

Serwer	Czas
1	4
2	2
3	5
4	1
5	6

Tablica 1: Czas odczytu danych dla każdego serwera

Cecha / Serwer	1	2	3	4	5
1	0	0	0	0	1
2	1	1	1	1	0
3	1	0	0	1	0
4	0	1	1	1	0
5	1	0	0	0	0
6	0	0	0	1	0
7	0	0	1	1	0

Tablica 2: Obecność danych cech na każdym serwerze

Dla tych danych optymalnym rozwiązaniem jest pobranie danych z serwerów 1, 4 i 5. Sumaryczny czas wyniesie 11 jednostek.

2 Zadanie 2

Zadanie 2 polega na ułożeniu sekwencyjnego programu, składającego się z określonego zbioru funkcji I . Należy dobrać odpowiednie podprogramy, aby cały program zajmował nie więcej niż M komórek pamięci, a czas jego wykonania był minimalny.

2.1 Model

Funkcja celu:

$$\min \sum_{i \in I, j \in [m]} x_{ij} * T_{ij}$$

gdzie

- m - liczba podprogramów do obliczenia funkcji
- I - zbiór funkcji do policzenia. $i \in I \leq n$, gdzie n to liczba wszystkich możliwych funkcji
- T_{ij} - czas działania podprogramu j dla funkcji i
- x_{ij} - zmienna binarna określająca, czy podprogram j zostanie użyty do policzenia funkcji i . $x_{ij} \in \{0, 1\}$, $i \in I$, $j \in [m]$

Ograniczenia:

- $\sum_{i \in I, j \in [m]} x_{ij} * R_{ij} \leq M$, gdzie R_{ij} to ilość pamięci, wymagana przez podprogram j . Dane ograniczenie zapewnia, że cały program zużywa maksymalnie M komórek pamięci
- $\forall i \in I \sum_{j \in [m]} x_{ij} = 1$ - zapewnia wybranie dokładnie jednego podprogramu dla danej funkcji

2.2 Wyniki

Zdefiniowano następujący egzemplarz problemu: $I = \{1, 2, 4\}$, $M = 15$

Funkcja / Podprogram	1	2	3	4
1	3	2	1	9
2	1	2	3	5
3	4	5	2	8
4	1	8	2	8

Tablica 3: Wymgania pamięciowe

Funkcja / Podprogram	1	2	3	4
1	4	6	9	2
2	10	8	7	5
3	3	5	2	1
4	15	2	8	3

Tablica 4: Wymagania czasowe

Otrzymano następujące rozwiązanie optymalne:

- Czas wykonania programu: 13
- Zużycie pamięci: 15

Funkcja / Podprogram	1	2	3	4
1		x		
2				x
4		x		

Tablica 5: Których podprogramów użyć

3 Zadanie 3

Dane zadanie polega na stworzeniu harmonogramu wykonania zadań na procesorach.

3.1 Model

Niech:

- n - liczba procesorów
- m - liczba zadań do wykonania
- T_{ji} - czas wykonania zadania j na procesorze i
- O_{jik} - zmienna pomocnicza binarna, wyznaczająca kolejność zadań. $O_{jik} = 1$ oznacza że zadanie i zaczyna się szybciej od zadania k na procesorze j
- S_{ji} - czas rozpoczęcia zadania i na procesorze j
- C_{max} - czas zakończenia wszystkich zadań

Funkcja celu:

$$\min C_{max}$$

Ograniczenia:

- Wszystkie zadania muszą być zakończone przed zakończeniem ostatniego

$$\forall(i \in [m]) S_{ni} + T_{ni} \leq C_{max}$$

- Każdy procesor może wykonywać tylko jedno zadania naraz

$$\begin{aligned} \forall(j \in [n], i, k \in [m]) S_{jk} &\geq S_{ji} + T_{ji} \text{ , jeżeli } O_{jik} = 1 \\ \forall(j \in [n], i, k \in [m]) S_{ji} &\geq S_{jk} + T_{jk} \text{ , jeżeli } O_{jik} = 0 \end{aligned}$$

- Wykonanie zadania może się zacząć tylko po jego zakończeniu na poprzednim procesorze

$$\forall(i \in [m], j \in [n-1]) S_{(j+1)i} \geq S_{ji} + T_{ji}$$

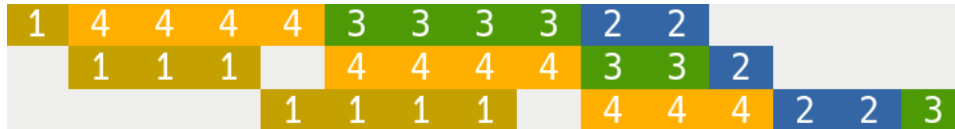
3.2 Wyniki

Zdefiniowano następujący egzemplarz problemu: $n = 3$

Procesor / Zadanie	1	2	3	4
1	1	2	4	4
2	3	1	2	4
3	4	2	1	3

Tablica 6: Czas wykonania zadań na każdym z procesorów

Optymalnym jest następujący harmonogram:



$$C_{max} = 15$$