METODY OPTYMALIZACJI – LABORATORIUM 2

Adrian Mucha, Politechnika Wrocławska 03/05/2020

1 Chmura rozproszonych danych

1.1 Model

Dane są następujące parametry:

- T_j - wektor zawierający czasy potrzebne na przeszukanie j-tego serwera.
- q_{ij} - macierz zawierająca informację o tym, które cechy i zawiera j-ty serwer (1 - obecność informacji, 0 - brak informacji).
- k - ilość serwerów (wnioskowana na podstawie wektora T_j)
- n - ilość cech (wnioskowana na podstawie wektora q_{ij})

1.1.1 Zmienne decyzyjne

Zmienną decyzyjną jest wektor $\mathbf{x}$ o długości k odpowiadającej liczności serwerów. Wektor decyduje czy w ostatecznym przeszukiwaniu uwzględniany jest j-ty serwer ($x_j = 1$) czy nie ($x_j = 0$).

1.1.2 Ograniczenia

Przynajmniej jeden wybrany serwer j zawiera dostęp do cechy i-tej

$$\forall_{i \in [n]} (\sum_{j=1}^{k} x_j \cdot q_{ij} \geq 1)$$

1.1.3 Funkcja kosztu

Dążymi do minimalizowania czasów dostępu do wszystkich serwerów tak aby odczytać wszystkie cechy. Koszt opisuje następująca funkcja

$$f(T, x) = \mathit{min} \sum_{j=1}^{k} T_j \cdot x_j$$

1.2 Przykładowe dane

Dla następujących danych:

- $T = [1, 2, 5, 5]$

- $q = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$

  solver GLPK znalazł następujące rozwiązanie o koszcie $f(T,x) = 6$

$$x = [1, 0, 0, 1]$$

2 Biblioteka podprogramów

2.1 Model

Dane są następujące parametry, znajdź optymalny zestaw podprogramów P_{ij} rozwiązujący program P składający się z funkcji I minimalizując czas przy jednoczesnym ograniczeniu pamięci:

- m - ilość funkcji
- n - ilość podprogramów
- M - górne ograniczenie pamięci programu P
- P_{ij} - biblioteka j-tych podprogramów obliczających i-tą funkcję, $i \in [m], j \in [n]$
- r_{ij} - pamięć zużywana przez podprogram P_{ij}
- t_{ij} - czas wykonania podprogramu P_{ij}
- $I \subseteq \{1, \ldots, m\}$ - funkcje składające się na program P

2.1.1 Zmienne decyzyjne

Zmienną decyzyjną jest macierz $\mathbf{x}$ o wymiarach $m \times n$ przechowującą informację o wykorzystanych podprogramach w celu wykonania programu P. Mianowicie $x_{ij} = 1$ oznacza wykorzystanie podprogramu P_{ij} do obliczenia i-tej funkcji, oraz $x_{ij} = 0$ w p.p.

2.1.2 Ograniczenia

- Uwzględniaj tylko zlecone funkcje, jeden podprogram na funkcję

$$\forall_{i \in I} \sum_{j=1}^{n} x_{ij} = 1$$

- Nie wykorzystaj więcej pamięci niż M podczas całej pracy programu P

$$\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \cdot r_{ij} \leq M$$

2.1.3 Funkcja kosztu

Minimalizujemy czas pracy programu P

$$f(t,x) = \textit{min} \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \cdot t_{ij}$$

2.2 Przykładowe dane

Dla parametrów

- $m = 3$
- $n = 4$
- $M = 10$
- $I = [1, 3]$

wygenerowano bibliotekę funkcji P_{ij} w następujący sposób

$$r_{ij} = i \cdot j, \quad t_{ij} = \left\lfloor \frac{100}{r_{ij}} \right\rfloor$$

solver GLPK znalazł następujące rozwiązanie o koszcie (czasie) $f(T, x) = 41$

$$x = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

3 Gantt

3.1 Model

- m - ilość zadań
- n - ilość procesorów
- d_{ij} - macierz $m \times n$ zawierajaca czasy wykonania i-tego zadania na j-tym procesorze, $i \in [m], j \in [n]$

3.1.1 Zmienne decyzyjne

Zdefiniowano następujące zmienne decyzyjne

- macierz $\mathbf{t}$ o wymiarach $m \times n$ przechowująca czasy rozpoczęcia i-tego zadania na j-tym procesorze.
- C_{max} - czas zakonczenia wszystkich zadań (w szczególności ostatniego zadania na trzecim procesorze)
- x_{jik} - pomocnicza zmienna określająca kolejność wykonywanych zadań, $j \in [n],\ i, k \in [m]$

3.1.2 Ograniczenia

1. moment rozpoczecia i-tego zadania na $j+1$-tym procesorze może zacząć się dopiero gdy i-te zadanie skończy się na j-tej

$$\forall_{i\in[m],j\in[n],i<n}\ t_{i,j+1} \geq t_{ij} + d_{ij}$$

2. tylko jedno zadanie wykonywane jest w danym momencie na j-tym procesorze (B oznacza bardzo dużą liczbę, pełniącą funkcję strażnika)

$$t_{ij} - t_{kj} + B \cdot x_{jik} \geq d_{kj}$$

$$t_{kj} - t_{ij} + B \cdot (1 - x_{jik}) \geq d_{ij}$$

3. C_{max} jest ostatnim wykonanym zadaniem na trzecim procesorze

$$t_{in} + d_{in} \leq C_{\text{max}}$$

3.1.3 Funkcja kosztu

Minimalizujemy C_{max}

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

dla pewnej permutacji zadań π, gdzie $\pi(n)$ oznacza ostatnie w kolejności wykonywania zadanie.

3.2 Przykładowe dane

- a - czas trwania zadań na procesorze 1
- b - czas trwania zadań na procesorze 2
- c - czas trwania zadań na procesorze 3

$$a = [3, 9, 9, 4, 6, 6, 7], \quad b = [3, 3, 8, 8, 10, 3, 10], \quad c = [2, 8, 5, 4, 3, 1, 3]$$

$$d = [a\, b\, c]$$

otrzymujemy wynik (czasy rozpoczęcia zadań), w danej kolumnie, ułożone zostały spermutowane czasy rozpoczęcia zadań. Aby otrzymać permutację etykiet, należy posortować czasy w kolumnie i odczytać kolejność zadań

$$t = \begin{pmatrix} 41 & 44 & 48 \\ 17 & 32 & 35 \\ 26 & 35 & 43 \\ 0 & 4 & 12 \\ 11 & 22 & 32 \\ 35 & 47 & 50 \\ 4 & 12 & 29 \end{pmatrix}$$

gdy posortujemy pierwszą (dowolną) kolumnę rosnąco otrzymamy następującą permutację zadań $Z = [4, 7, 5, 2, 3, 6, 1]$ w której $C_{\text{max}} = 51$