Metody Optymalizacji
Lista 3

Dominik Kaczmarek, nr albumu 261757

2 czerwca 2024

1 Generalized Assignment Problem

1.1 Opis problemu

Dany jest zbiór zadań J oraz maszyn M. Dla każdego zadania $j \in J$ i maszyny $i \in M$ istnieje czas przetwarzania p_{ij} oraz koszt c_{ij}. Każda z maszyna $i \in M$ jest dostępna przez T_i jednostek czasu. Celem jest przypisanie każdego zadania $j \in J$ do jakiejś maszyny $i \in M$ w taki sposób, aby zminimalizować całkowity koszt oraz aby żadna maszyna nie pracowała więcej jednostek czasu niż wynosi jej limit T_i.

1.2 Model (ILP)

1.2.1 Parametry

- M - zbiór maszyn
- J - zbiór zadań,
- $m = |M|$ - liczba maszyn,
- $n = |J|$ - liczba zadań,
- p_{ij} - czas wykonania zadania $j \in J$ maszyną $i \in M$,
- c_{ij} - koszt wykonania zadania $j \in J$ maszyną $i \in M$,
- T_i - jednostki czasu, w których maszyna $i \in M$ może pracować

1.2.2 Zmienne decyzyjne

$$x_{ij} \in \{0,1\} - \begin{cases} 1, & \text{zadanie } j \in J \text{ wykonujemy maszyną } i \in M \\ 0, & \text{w.p.p.} \end{cases}$$

1.2.3 Funkcja celu

Minimalizacja kosztu wykonania wszystkich zadań:

$$C = \min \sum_{i \in M} \sum_{j \in J} c_{ij} x_{ij}$$

1.2.4 Ograniczenia

1. Każde zadanie musi być wykonane:

$$\text{s.t.} \quad \sum_{i \in M} x_{ij} = 1, \quad \forall_{j \in J}$$

2. Maszyna nie może przekroczyć swojego czasu pracy:

$$\text{s.t.} \quad \sum_{j \in J} p_{ij} x_{ij} \leqslant T_i \quad \forall_{i \in M}$$

1.3 Algorytm aproksymacyjny

Problem *Generalized Assignment Problem* (`GPA`) jest probelemem NP-trudnym [3]. Istnieje jednak wielomianowy algorytm autorstwa Shmoysa i Tardosa [2], który zwraca rozwiązanie o koszcie co najwyżej C, gdzie każda maszyna jest używana przez co najwyżej $2T_i$ jednostek czasu. C jest kosztem rozwiązania optymalnego, które wykorzystuje maszynę i przez co najwyżej T_i jednostek czasu (jeśli takie przypisanie jest możliwe).

1.3.1 Redukcja GPA do Bipartite Matching Problem

Problem *Generalized Assignment Problem* możemy zredukować do problemu znalezienia skojarzenia w grafie dwudzielnym [1]. Zaczynamy od pełnego grafu dwudzielnego $G = (V, E)$, gdzie $V = M \cup J$. Krawędź między zadaniem $j \in J$ a maszyną $i \in M$ ma koszt c_{ij}. Problem uogólnionego przydziału można sprowadzić do znalezienia podgrafu F grafu G takiego, że $d_F(j) = 1$ dla każdego zadania $j \in J$, a krawędź incydentna do j wskazuje, do której maszyny i przydzielono zadanie j. Ograniczenia czasowe dla maszyn można modelować poprzez ograniczenie, że $\sum_{e \in \delta(i) \cap F} p_{ij} \leqslant T_i$ dla każdej maszyny i. Wzmacniamy ten model, wykluczając niektóre przypisania, korzystając z następującej obserwacji: jeśli $p_{ij} > T_i$, to żadne optymalne rozwiązanie nie przydziela zadania j do i, w związku z czym możemy usunąć wszystkie takie krawędzie z grafu G.

1.3.2 Relaksacja

Modelujemy `GPA` jako `BMP` i używamy relaksacji do programowania liniowego LP_{ga}. Model LP_{ga} wykorzystamy w alogrytmie aproksymacyjnym. Zauważmy, że nie nakładamy ograniczeń czasowych na wszystkie maszyny, lecz na podzbiór $M' \subseteq M$, który na początku jest równy M. Mamy zmienną x_e dla każdego $e = ij$, oznaczającą, czy zadanie j jest przydzielone do maszyny i.

$$
\begin{aligned}
\min \quad & \sum_{e=(i,j) \in E} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{e \in \delta(j)} x_e = 1 \quad \forall j \in J \\
\text{s.t.} \quad & \sum_{e \in \delta(i)} p_e x_e \leqslant T_i \quad \forall i \in M' \\
\text{s.t.} \quad & x_e \geqslant 0 \quad \forall e \in E
\end{aligned}
$$

1.3.3 Pseudokod

Algorithm 1 Iteracyjny Algorytm Uogólnionego Przydziału [1]

Inicjalizacja: $E(F) \leftarrow \emptyset$, $M' \leftarrow M$
while $J \neq \emptyset$ **do**
    Znajdź optymalne rozwiązanie ekstremalne x dla LP_{ga} i usuń każdą zmienną $x_{ij} = 0$
    if $\exists_{(i,j) \in (M \times J)}(x_{ij} = 1)$ **then**
        $F \leftarrow F \cup \{ij\}$
        $J \leftarrow J \setminus \{j\}$
        $T_i \leftarrow T_i - p_{ij}$
    end if
    if $\exists_{i \in M}((d(i) = 1) \vee (d(i) = 2 \wedge \sum_{j \in J} x_{ij} \geqslant 1))$ **then**
        $M' \leftarrow M' \setminus \{i\}$
    end if
end while
return F

1.4 Wyniki

1.4.1 Oznaczenia w tabelach

- T'_i - całkowity czas pracy maszyny $i \in M$ uzyskany algorytmem aproksymacyjnym (zgodnie z założeniem powinna zachodzić nierówność $T'_i \leqslant 2T_i$)
- C' - optymalny całkowity koszt uzyskany algorytmem aproksymacyjnym
- C - optymalny całkowity koszt problemu bazowego (rozdział 1.2)

Tabela 1: Wyniki dla pliku gap1.txt

Problem	$\max(T'_i/T_i)$	C'	C	C'/C
1	1.3684	188	336	0.5595
2	1.2708	212	327	0.6483
3	1.7045	194	339	0.5723
4	1.2778	201	341	0.5894
5	1.6471	213	326	0.6534

Tabela 2: Wyniki dla pliku gap2.txt

Problem	$\max(T'_i/T_i)$	C'	C	C'/C
1	1.1667	268	434	0.6175
2	1.2391	247	436	0.5665
3	1.1556	243	420	0.5786
4	1.2692	293	419	0.6993
5	1.3111	245	428	0.5724

Tabela 3: Wyniki dla pliku gap3.txt

Problem	$\max(T'_i/T_i)$	C'	C	C'/C
1	1.1562	310	580	0.5345
2	1.5849	299	564	0.5301
3	1.1964	313	573	0.5462
4	1.1719	312	570	0.5474
5	1.375	316	564	0.5603

Literatura

[1] R. Ravi. *Approximation Algorithms and Metaheuristics.* Accessed: 2024-06-02.

[2] David Shmoys and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, 62(3):461–474, 1993.

[3] Wikipedia. Generalized assignment problem — wikipedia, the free encyclopedia. Accessed: 2024-06-02.

Tabela 4: Wyniki dla pliku gap4.txt

Problem	$\max(T_i'/T_i)$	C'	C	C'/C
1	1.2222	360	656	0.5488
2	1.2097	380	644	0.5901
3	1.2179	389	673	0.578
4	1.2895	378	647	0.5842
5	1.2	388	664	0.5843

Tabela 5: Wyniki dla pliku gap5.txt

Problem	$\max(T_i'/T_i)$	C'	C	C'/C
1	1.4857	307	563	0.5453
2	1.325	285	558	0.5108
3	1.5714	275	564	0.4876
4	1.4688	280	568	0.493
5	1.3939	303	559	0.542

Tabela 6: Wyniki dla pliku gap6.txt

Problem	$\max(T_i'/T_i)$	C'	C	C'/C
1	1.3556	393	761	0.5164
2	1.2449	406	759	0.5349
3	1.4444	392	758	0.5172
4	1.6341	385	752	0.512
5	1.46	425	747	0.5689

Tabela 7: Wyniki dla pliku gap7.txt

Problem	$\max(T_i'/T_i)$	C'	C	C'/C
1	1.5079	484	942	0.5138
2	1.4643	486	949	0.5121
3	1.2982	493	968	0.5093
4	1.2931	508	945	0.5376
5	1.2759	485	951	0.51

Tabela 8: Wyniki dla pliku gap8.txt

Problem	$\max(T_i'/T_i)$	C'	C	C'/C
1	1.1923	411	1133	0.3628
2	1.1458	404	1134	0.3563
3	1.3061	414	1141	0.3628
4	1.4375	397	1117	0.3554
5	1.2105	424	1127	0.3762

Tabela 9: Wyniki dla pliku gap9.txt

Problem	$\max(T_i'/T_i)$	C'	C	C'/C
1	1.25	357	709	0.5035
2	1.4828	374	717	0.5216
3	1.6129	395	712	0.5548
4	1.4615	375	723	0.5187
5	1.4706	404	706	0.5722

Tabela 10: Wyniki dla pliku gap10.txt

Problem	$\max{(T_i'/T_i)}$	C'	C	C'/C
1	1.4043	479	958	0.5
2	1.5814	503	963	0.5223
3	1.3636	509	960	0.5302
4	1.4082	477	947	0.5037
5	1.1569	468	947	0.4942

Tabela 11: Wyniki dla pliku gap11.txt

Problem	$\max{(T_i'/T_i)}$	C'	C	C'/C
1	1.2787	665	1139	0.5838
2	1.4545	648	1178	0.5501
3	1.3506	665	1195	0.5565
4	1.3143	698	1171	0.5961
5	1.2273	626	1171	0.5346

Tabela 12: Wyniki dla pliku gap12.txt

Problem	$\max{(T_i'/T_i)}$	C'	C	C'/C
1	1.1795	752	1451	0.5183
2	1.1806	724	1449	0.4997
3	1.2424	761	1433	0.5311
4	1.2714	728	1447	0.5031
5	1.2778	769	1446	0.5318

Tabela 13: Wyniki dla pliku gapa.txt

Problem	$\max{(T_i'/T_i)}$	C'	C	C'/C
1	1.0556	1529	—	—
2	1.0253	2981	—	—
3	1.1406	1519	—	—
4	1.0164	2943	—	—
5	1.09	1472	—	—
6	1.1407	3029	—	—

Tabela 14: Wyniki dla pliku gapb.txt

Problem	$\max{(T_i'/T_i)}$	C'	C	C'/C
1	1.0287	1051	—	—
2	1.0325	2475	—	—
3	1.3511	1335	—	—
4	1.1029	2471	—	—
5	1.25	1353	—	—
6	1.3282	2578	—	—

Tabela 15: Wyniki dla pliku gapc.txt

Problem	$\max{(T_i'/T_i)}$	C'	C	C'/C
1	1.0603	1187	—	—
2	1.0199	2464	—	—
3	1.2321	1210	—	—
4	1.075	2440	—	—
5	1.3571	1240	—	—
6	1.1301	2419	—	—

Tabela 16: Wyniki dla pliku gapd.txt

Problem	$\max(T_i'/T_i)$	C'	C	C'/C
1	1.0777	4173	—	—
2	1.0289	8213	—	—
3	1.1759	4072	—	—
4	1.077	8225	—	—
5	1.4444	4290	—	—
6	1.2038	8429	—	—