

Metody optymalizacji

Zestaw nr 3

Tomasz Jankowiak 249006

12.06.2023

1 Wstęp

Celem zadania było zapoznanie się z 2-aproksymacyjnym algorytmem Vaziraniego opartym na programowaniu liniowym dla problemu szeregowania zadań na niezależnych maszynach z kryterium minimalizacji długości uszeregowania (ang. Scheduling on Unrelated Parallel Machines and Makespan Criterion) oraz zaimplementowanie go w języku Julia z użyciem pakietu JuMP.

2 Opis problemu

Rozpatrujemy problem szeregowania zadań na wielu maszynach. Poszukujemy harmonogramu, minimalizującego czas zakończenia ostatniego zadania.

Dane

- zbiór zadań $J=\{1,\dots,n\}$,
- zbiór maszyn $M=\{1,\dots,m\}$,
- czasy wykonania poszczególnych zadań na maszynach p_{ij} , $j \in J$, $i \in M$.

Zmienne decyzyjne

$$x_{ij} = \begin{cases} 1 & \text{jeśli zadanie } j \text{ jest szeregowane na maszynie } i, \\ 0 & \text{w przeciwnym przypadku.} \end{cases}$$

Ograniczenia

- Każde zadanie jest szeregowane na dokładnie jednej maszynie:

$$\sum_{i \in M} x_{ij} = 1, \quad j \in J$$

- Funkcja celu - każda maszyna kończy pracę najpóźniej w chwili t :

$$\sum_{j \in J} x_{ij} p_{ij} \leq t, \quad i \in M$$

- Binarność zmiennych decyzyjnych:

$$x_{ij} \in \{0, 1\}, \quad i \in M, j \in J.$$

Funkcja celu

Minimalizacja makespanu t , czyli czasu zakończenia ostatniego zadania

$$t \rightarrow \min,$$
$$t = \max_{i \in M} \sum_{j \in J} x_{ij} p_{ij}$$

3 Algorytm

Algorithm 17.5 (Scheduling on unrelated parallel machines)

1. By a binary search in the interval $[\alpha/m, \alpha]$, find the smallest value of $T \in \mathbb{Z}^+$ for which $LP(T)$ has a feasible solution. Let this value be T^* .
2. Find an extreme point solution, say x , to $LP(T^*)$.
3. Assign all integrally set jobs to machines as in x .
4. Construct graph H and find a perfect matching \mathcal{M} in it (e.g., using the procedure of Lemma 17.7).
5. Assign fractionally set jobs to machines according to matching \mathcal{M} .

Wprowadzamy parametr $T \in \mathbb{Z}^+$, będący naszą przypuszczalną dolną granicą optymalnego makespanu. Pozwoli nam to wyciąć wszystkie pary zadanie-maszyna, w których $p_{ij} > T: \forall i \in M, j \in J$: jeśli $p_{ij} > T$ to $x_{ij} = 0$.

1. Wykorzystując poszukiwanie binarne, znajdujemy najmniejszą wartość T , dla której istnieje potencjalne rozwiązanie. Jest to dolna granica rozwiązania optymalnego.
2. Konstruujemy rozwiązanie bazowe dopuszczalne x .
3. Zadania przyporządkowane do jednej maszyny przypisujemy do tej maszyny jak w x .
4. Zadania przyporządkowane do kilku maszyn przypisujemy do jednej, w szczególności jeśli maszyna ma tylko jedno zadanie, to do niej.

4 Wyniki

Dane	ALG C_{max}	OPT C_{max}	$\frac{ALGC_{max}}{OPTC_{max}}$
1a100, 111.txt	141	117	1,2
1a100, 112.txt	110	87	1,3
1a100, 113.txt	121	98	1,2
1a100, 114.txt	126	115	1,1
1a100, 115.txt	116	104	1,1
1a100, 116.txt	107	101	1,1
1a100, 117.txt	130	114	1,1
1a100, 118.txt	119	103	1,2
1a100, 119.txt	138	114	1,2
1a100, 120.txt	115	98	1,2
1a100, 121.txt	41	30	1,4
1a100, 122.txt	46	36	1,3
1a100, 123.txt	46	32	1,4
1a100, 124.txt	52	33	1,6
1a100, 125.txt	41	31	1,3
1a100, 126.txt	40	29	1,4
1a100, 127.txt	45	35	1,3
1a100, 128.txt	38	30	1,3
1a100, 129.txt	43	30	1,4
1a100, 130.txt	42	29	1,4
1a100, 131.txt	24	19	1,3
1a100, 132.txt	25	23	1,1
1a100, 133.txt	27	16	1,7
1a100, 134.txt	25	18	1,4
1a100, 135.txt	22	16	1,4
1a100, 136.txt	30	24	1,3
1a100, 137.txt	22	17	1,3
1a100, 138.txt	25	19	1,3

1a100, 139.txt	24	16	1,5
1a100, 140.txt	23	17	1,4
1a100, 141.txt	18	10	1,8
1a100, 142.txt	17	12	1,4
1a100, 143.txt	16	12	1,3
1a100, 144.txt	14	10	1,4
1a100, 145.txt	18	11	1,6
1a100, 146.txt	16	12	1,3
1a100, 147.txt	20	14	1,4
1a100, 148.txt	20	13	1,5
1a100, 149.txt	26	23	1,1
1a100, 150.txt	17	14	1,2
1a100, 151.txt	13	8	1,6
1a100, 152.txt	18	13	1,4
1a100, 153.txt	16	10	1,6
1a100, 154.txt	15	11	1,4
1a100, 155.txt	11	9	1,2
1a100, 156.txt	15	10	1,5
1a100, 157.txt	11	8	1,4
1a100, 158.txt	15	11	1,4
1a100, 159.txt	12	9	1,3
1a100, 160.txt	14	9	1,6
1a100, 211.txt	224	204	1,1
1a100, 212.txt	222	204	1,1
10a100, 111.txt	209	198	1,1
10a100, 112.txt	185	173	1,1

5 Podsumowanie wyników i wnioski

Jak widać, dla przykładowych danych rozpatrywany algorytm daje dobre rozwiązania, bliskie optymalnym. Eksperymenty potwierdziły wysoką jakość algorytmu. W żadnym przypadku iloraz nie przekroczył 2, co jest współczynnikiem aproksymacji algorytmu. Zatem najprawdopodobniej ćwiczenie wykonano poprawnie.