METODY OPTYMALIZACJI – LABORATORIUM 3

Adrian Mucha, Politechnika Wrocławska 31/05/2020

Wstęp

W niniejszej pracy zaimplementowano algorytm aproksymacyjny oparty na programowaniu liniowym dla zagadnienia *Generalized Assignment Problem* z użyciem pakietu `JuMP`.

Zestaw danych testowych pochodzi z biblioteki *OR-Library*[1], natomiast szczegóły dotyczące algorytmu znajdują się w książce *Iterative Methods in Combinatorial Optimization*[2].

Uogólnione zagadnienie przydziału

Opis problemu

Zadaniem jest aproksymacja optymalnego przydziału zbioru maszyn (potrafiących pracować równolegle) do zbioru zadań w taki sposób by minimalizować łączny czas wykonywania przy jednoczesnym uwzględnieniu ograniczeń jakie są nałożone na maszyny.

Model

- M - zbiór maszyn
- J - zbiór zadań
- T_i - i-ta maszyna może pracować łącznie T_i jednostek czasu, $i \in M$
- c_{ij} - koszt wykonania j-tego zadania na i-tej maszynie
- p_{ij} - czas pracy nad zadaniem j-tym (żużycie zasobów) przez i-tą maszynę

Problem sprowadza się do stworzenia grafu dwudzielnego którego wierzchołki dzielą się na zbiory M oraz J. Początkowo tworzony jest graf pełny $G_{M,J}$, gdzie krawędź między maszyną $i \in M$ a zadaniem $j \in J$ posiada koszt c_{ij}.

Celem jest znalezienie podgrafu $F \subset G$, takiego że $(\forall j \in J)\ d_F(j) = 1$. Krawędzie incydentne do zadania j reprezentują do której maszyny zostało ono przypisane.

Zmienne decyzyjne

- Macierz $\mathbf{X}$, której elementy $\mathbf{x_{ij}}$ mówią o tym czy j-te zadanie zostało przypisane do i-tej maszyny.

[1] http://people.brunel.ac.uk/ mastjjb/jeb/orlib/gapinfo.html

[2] http://www.contrib.andrew.cmu.edu/ ravi/book.pdf

Ograniczenia

Niech $M' \subseteq M$ - zbiór maszyn

- Każde zadanie przyporządkowano jedynie raz

$$(\forall j \in J) \sum_{e=(i,j)\in\delta(j)} x_e = 1$$

- Czas wykonywania zadań na maszynie nie przekracza jego czasu dostępności

$$(\forall i \in M) \sum_{e\in\delta(i)} p_e x_e \leq T_i$$

- zmienne decyzyjne są nieujemne

$$(\forall e \in E)\ x_e \geq 0$$

Funkcja kosztu

Całkowity koszt po krawędziach grafu G

$$\min \sum_{e\in E} c_e x_e$$

Algorytm iteracyjny

1. $E(F) \longleftarrow \emptyset,\ M' \longleftarrow M$
2. Dopóki $J \neq \emptyset$
   (a) Znajdź optymany punkt ekstremalny x zagadnienia LP_{ga} i usuń każdą zmienną decyzyjną x_{ij}
   (b) Jeżeli istnieje $x_i j = 1$
      i. $F \longleftarrow F \cup \{ij\}$
      ii. $J \longleftarrow J \setminus \{j\}$
      iii. $T_i \longleftarrow T_i - p_{ij}$
   (c) (**Relaksacja**) Jeżeli istnieje maszyna i o stopniu $d(i) = 1$ lub $d(i) = 2$ posiadająca $\sum_{j\in J} x_{ij} \geq 1$, to aktualizuj $M' \longleftarrow M' \setminus \{i\}$

Powyższy algorytm zapewnia, że maszyna i jest używana nie więcej niż $2T_i$ jednostek czasu swojej dostępności.

Wyniki

Poniższe tabele prezentują otrzymane wyniki. $T_{\max}(F)$ jest maksynalnym odchyleniem czasu przypisanego dla maszyny, oraz kolejno $T_{\max}$ jest odpowiadającemu mu ograniczeniu.

- Pierwszy wpis w tabeli posiada zaburzony czas, gdyż algorytm dopiero się "rozgrzewał"

- Czas wykonywania zależy od rozmiaru problemu i mieści się w granicach 5-40ms
- Algorytm potrzebował od 5 do 9 iteracji aby rozwiązać dany egzemplarz
- Maksymalne przeciążenie maszyny nigdy nie przekroczyło $2T_i$ i w zaobserwowanych wynikach średnio były przeciążone o 25%, z maksymalnym odchyleniem 65%

Plik	Problem	Iteracje	$T_{\max}(F)$	$T_{\max}$	$\frac{T_{\max}(F)}{T_{\max}}$	czas
gap1.txt	c515-1	6	52	38	1.368421052631579	669ms
	c515-2	6	50	37	1.3513513513513513	5ms
	c515-3	5	54	37	1.4594594594594594	5ms
	c515-4	5	46	36	1.2777777777777777	5ms
	c515-5	5	56	34	1.6470588235294117	4ms
gap2.txt	c520-1	7	49	42	1.1666666666666667	8ms
	c520-2	5	57	46	1.2391304347826086	6ms
	c520-3	6	63	48	1.3125	6ms
	c520-4	6	66	52	1.2692307692307692	6ms
	c520-5	5	59	45	1.3111111111111111	7ms
gap3.txt	c525-1	5	74	64	1.15625	6ms
	c525-2	7	58	48	1.2083333333333333	8ms
	c525-3	5	67	56	1.1964285714285714	8ms
	c525-4	5	75	64	1.171875	9ms
	c525-5	6	77	56	1.375	10ms
gap4.txt	c530-1	6	88	72	1.2222222222222223	7ms
	c530-2	6	75	62	1.2096774193548387	8ms
	c530-3	5	95	78	1.2179487179487178	9ms
	c530-4	7	88	78	1.1282051282051282	11ms
	c530-5	8	78	65	1.2	13ms
gap5.txt	c824-1	6	52	35	1.4857142857142858	12ms
	c824-2	5	53	40	1.325	9ms
	c824-3	6	46	38	1.2105263157894737	35ms
	c824-4	6	47	32	1.46875	9ms
	c824-5	5	46	33	1.393939393939394	7ms
gap6.txt	c832-1	5	61	45	1.3555555555555556	9ms
	c832-2	5	61	49	1.2448979591836735	11ms
	c832-3	5	65	45	1.4444444444444444	11ms
	c832-4	6	62	41	1.5121951219512195	12ms
	c832-5	6	73	50	1.46	12ms

Plik	Problem	Iteracje	$T_{\max}(F)$	$T_{\max}$	$\frac{T_{\max}(F)}{T_{\max}}$	czas
gap7.txt	c840-1	5	75	63	1.1904761904761905	13ms
	c840-2	6	68	56	1.2142857142857142	19ms
	c840-3	6	74	57	1.2982456140350878	13ms
	c840-4	6	75	58	1.293103448275862	18ms
	c840-5	6	69	56	1.2321428571428572	19ms
gap8.txt	c848-1	5	62	52	1.1923076923076923	15ms
	c848-2	6	55	48	1.1458333333333333	18ms
	c848-3	6	59	49	1.2040816326530612	17ms
	c848-4	7	55	47	1.1702127659574468	25ms
	c848-5	8	69	57	1.2105263157894737	19ms
gap9.txt	c1030-1	6	50	40	1.25	14ms
	c1030-2	8	47	38	1.236842105263158	17ms
	c1030-3	5	50	31	1.6129032258064515	10ms
	c1030-4	7	52	38	1.368421052631579	16ms
	c1030-5	7	50	34	1.4705882352941178	16ms
gap10.txt	c1040-1	6	64	49	1.3061224489795917	18ms
	c1040-2	5	68	48	1.4166666666666667	15ms
	c1040-3	6	60	44	1.3636363636363635	19ms
	c1040-4	6	52	44	1.1818181818181819	18ms
	c1040-5	7	59	51	1.1568627450980393	20ms
gap11.txt	c1050-1	7	78	61	1.278688524590164	22ms
	c1050-2	9	82	61	1.3442622950819672	31ms
	c1050-3	6	92	74	1.2432432432432432	23ms
	c1050-4	7	92	70	1.3142857142857143	41ms
	c1050-5	7	74	61	1.2131147540983607	20ms
gap12.txt	c1060-1	7	92	78	1.1794871794871795	29ms
	c1060-2	6	85	72	1.1805555555555556	24ms
	c1060-3	5	82	66	1.2424242424242424	21ms
	c1060-4	8	92	74	1.2432432432432432	35ms
	c1060-5	6	92	72	1.2777777777777777	24ms