Sprawozdanie z laboratorium:
Praktyka i Teoria Szeregowania Zadań

Część I: Prosta heurystyka konstruująca pojedyncze rozwiązanie problemu
$1 \mid d_j = d \mid \sum Ew + \sum Tw$

14 listopada 2018

Autor: **Adam Czyżewski** inf127198 I2 adam.czyzewski@student.put.poznan.pl

Zajęcia laboratoryjne: środa, godz. 9:45

1 Wstęp

Zadania szeregowania zadań na pojedynczej maszynie przy minilizowaniu sumarycznego, ważonego kosztu przekroczenia/przyspieszeniu jest problemem zaliczane do problemów $NP-trudnych$[1]. Skonstruowany algorytm jest prostą heurystyką, która generuje pojedyncze rozwiązanie bazując na wyliczonej wartości ważonych współczynników. Algorytm testowany był na zbiorze instancji $OR-Library$.

2 Algorytm

2.1 Nomenklatura

- d Czas docelowy zakończenia zadania (wspólny dla wszystkich zadań)
- h Wpółczynnik definujacy docelowy czas zadania na podstawie sumarycznego czasu wykonania p_i wszystkich zadań.
- p_i Czas wykonywania zadania i
- a_i Waga kosztu zakończenia zadania i przed terminie.
- b_i Waga kosztu zakończenia zadania i po terminie
- A_p Sumaryczny czas wykonywania zadań (p_i) ze zboru A
- B_p Sumaryczny czas wykonywania zadań (p_i) ze zboru B

2.2 Pseudokod

1. Dzielimy wszystkie dostępne zadania na 2 zbiory:
   (a) Zbiór $A \leftarrow$ zadania, gdzie $a_i < b_i$
   (b) Zbiór $B \leftarrow$ zadania, gdzie $a_i \geqslant b_i$
2. Obliczamy współczynniki, które będą stanowiły klucz sortowania zbioru A:
   (a) β_c - określa jaki ułamek czasu przetwarzania zadań ze zbioru A odbędzie się po wyznaczonym terminie
   $$\beta_c = max\{0, \frac{A_p - d}{A_p}\}$$
   (b) h_{sigm} - wskazuje jak bardzo powinniśmy brać pod uwagę termin docelowy zakończenia wszystkich zadań.
   $$h_{sigm} = \frac{1}{1 + e^{-3\dot{(}h-0.5)}}$$

(c) WPT_i^a - współczynnik ważonego czasu wykonania obliczany dla każdego zadania i na podstawie wagi a_i.

$$WPT_i^a = \frac{a_i}{p_i}$$

(d) WPT_i^b - współczynnik ważonego czasu wykonania obliczany dla każdego zadania i na podstawie wagi b_i.

$$WPT_i^b = \frac{b_i}{p_i}$$

(e) WET_i - heurystyczny współczynnik stosunku wagi b_i do wagi a_i zadania i

$$WET_i = \frac{b_i}{a_i}$$

3. Sortujemy zadania w zbiorze A nierosnąco przy użyciu klucza opisanego wzorem:

$$t_{key} = (1 + (1 - h_{sigm})) * WET_i + (1 - \beta_c + h_{sigm}) * WPT_i^{a_i} - (\beta_c - h_{sigm}) * WPT_i^b$$

4. Obliczamy czas, w którym rozpoczniemy wykonywanie pierwszego zadania, za pomocą wzoru: $max\{0, d - A_p\}$
5. Z posortowanego zbioru zadań A pobieramy ostatnie zadanie z listy i dodajemy je jako kolejne do ostatecznego zbioru rozwiązań, aż:
   (a) Zbiór A będzie pusty, lub
   (b) Kolejne zadanie ze zbioru A spowoduje, że sumaryczny czas wykonania (p_i) zadan dodanych do ostatecznego zbioru rozwiązań przekroczny docelowy czas wykonania każdego z zadań (d).
6. Jeżeli zbiór A nie jest pusty, to przerzucamy pozostałe zadania do zbioru B
7. Sortujemy zadania w zbiorze B niemalejąco przy użyciu klucza opisanego wzorem:

$$t_{key} = WPT_i^b$$

8. Z posortowanego zbioru zadań B pobieramy ostatnie zadanie z listy i dodajemy je jako kolejne do ostatecznego zbioru rozwiązań, aż zbiór B będzie pusty.
9. Zadania w ostaecznym zbiorze rozwiązań są ostatecznym wynikiem działania heurystyki.

2.3 Oszacowanie złożoności

Do implementacji wykorzystano język Python, który do celów sortowania używa algorytmu "Timsort". Algorytm wykonuje łącznie 2 sortowania, 2 podzbiorów zadań. Złożoność opisanego algorytmu w średnim, jak i najgorszym przypadku wynosi $O(nlogn)$.

3 Wyniki

($*$ - wartość optymalna)

Instancja	K	R	K	Błąd wzgl. wartości R [%]	Czas obliczeń [ms]
10	1	1025	1031	0.585	0.491
	6	908*	922	1.542	0.174
20	1	3066	3071	0.163	0.332
	6	3601	3656	1.222	0.247
50	1	24868	25371	2.023	0.959
	6	21497	24141	12.229	0.836
100	1	89588	94566	5.557	0.509
	6	86724	91543	5.557	0.462
200	1	301449	326273	8.235	1.055
	6	292453	314201	7.436	0.833
500	1	1839902	2010879	9.293	2.626
	6	1658411	1796465	8.324	3.165
1000	1	8570154	9292446	8.428	3.601
	6	7144491	7904956	10.644	3.909

Tabela 1: Wyniki dla współczynnika h=0.4

Instancja	K	R	K	Błąd wzgl. wartości R [%]	Czas obliczeń [ms]
10	1	841*	867	3.092	0.232
	6	755*	775	2.649	0.173
20	1	2986	3077	3.048	0.248
	6	3016	3415	13.229	0.231
50	1	17990	20469	13.780	0.477
	6	14251	15695	10.133	0.744
100	1	72019	74629	3.624	0.405
	6	62519	67283	7.620	1.007
200	1	254268	264715	4.109	0.875
	6	236160	249453	5.629	0.837
500	1	1581233	1663732	5.217	2.677
	6	1413345	1472589	4.192	1.648
1000	1	6411581	6850331	6.843	4.879
	6	6082142	6373019	4.782	7.789

Tabela 2: Wyniki dla współczynnika h=0.6

Błąd min.	Błąd maks.	Średni błąd	Odchylenie stand.	Średni czas obliczeń [ms]
0.163	13.780	6.045	3.796	1.479

Tabela 3: Wartości wyliczone na podstawie wszystkich danych

Literatura

[1] Martin Feldmann, Dirk Biskup *Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches*. Computers & Industrial Engineering, Volume 44, Issue 2, February 2003, Pages 307-323