

Sterowanie procesami dyskretnymi - sprawozdanie 1

Aleksandra Marecka, Artur Kowalczyk

22.03.2019

1 Wstęp

Na pierwszych zajęciach należało wykonać dwa algorytmy, których zadaniem było obliczenie minimalnego czasu oraz ustaleniu kolejności zadań na poszczególnych maszynach. Pierwszym z nich był algorytm przeglądu zupełnego, który liczy czas każdej z permutacji zadań, a następnie wybiera najkrótszy z nich. Kolejnym sposobem do rozwiązania zadania problemu przepływowego jest algorytm Johnsona, który w określony sposób wybiera kolejność wykonywania zadań bez konieczności liczenia każdej permutacji. Kolejnym algorytmem z jakim się zapoznaliśmy jest algorytm NEH, który w inny sposób rozwiązuje problem przepływowy.

2 Algorytm NEH, a inne algorytmy

Algorytm NEH służy do obliczenia problemu przepływowego poprzez najpierw zsumowanie czasów poszczególnego zadania na wszystkich maszynach, a następnie posortowaniu ich nierosnąco. Tak uporządkowana lista wybiera pierwszy element z niej oraz liczy dany c_{max} , następnie ustawia kolejne zadanie z listy we wszystkich dostępnych konfiguracjach oraz również liczy c_{max} , następnie wybiera jego najkrótszą wersję i do tej konfiguracji dobiera kolejny argument z listy, aż do ułożenia wszystkich zadań w kolejności, która będzie trwała najkrócej. Algorytm NEH w porównaniu do przeglądu zupełnego nie musi liczyć wszystkich możliwych permutacji danych zadań, przez co jest dużo szybszy i skuteczniejszy od przeglądu zupełnego. Algorytm Johnsona również liczy c_{max} szybciej niż przegląd zupełny, jednakże nie jest stosowany do wielomaszynowości. Jest on skuteczny jedynie dla dwóch lub trzech maszyn, gdzie w przypadku trzech maszyn musi z nich stworzyć 2 maszyny wirtualne sumując czas pierwszej i drugiej maszyny, traktując ją jako maszynę wirtualną 1 oraz czas drugiej i trzeciej maszyny traktując ją jako maszynę wirtualną 2.

Plik	Przegląd zupełny		Algorytm Johnsona		Algorytm NEH	
	cmax	Kolejność	cmax	Kolejność	cmax	Kolejność
dwie.txt	27	1, 5, 4, 3, 2	27	5, 1, 4, 3, 2	27	5, 1, 4, 3, 2
ta000.txt	32	1, 4, 3, 2	39	1, 4, 3, 2	32	1, 4, 3, 2
pdf.txt	18	1, 3, 2, 4	20	3, 1, 2, 4	18	3, 1, 2, 4
2_03.txt	188	[10, 12, 8, 15, 7, 11, 14, 6, 13, 1, 5, 9, 2, 3, 4]	188	[10, 12, 8, 15, 7, 11, 14, 6, 13, 1, 5, 9, 2, 3, 4]	188	[12, 10, 8, 15, 11, 7, 14, 6, 13, 1, 5, 9, 2, 4, 3]
2_04.txt	142	[8, 3, 5, 7, 12, 13, 14, 1, 6, 10, 2, 11, 9, 4]	142	[8, 3, 5, 7, 12, 13, 14, 1, 6, 10, 2, 11, 9, 4]	142	[8, 3, 7, 5, 12, 14, 13, 6, 1, 10, 11, 2, 9, 4]
2_05.txt	48	[1, 2, 3, 4, 7, 5, 8, 6]	48	[8, 7, 5, 4, 1, 3, 2, 6]	48	[8, 3, 1, 7, 5, 4, 2, 6]
2_06.txt	107	[3, 1, 4, 6, 7, 8, 2, 5]	107	[8, 7, 3, 1, 6, 4, 2, 5]	107	[8, 7, 3, 6, 1, 4, 5, 2]
2_07.txt	32	[1, 5, 4, 3, 2]	32	[5, 4, 1, 3, 2]	32	[5, 1, 4, 3, 2]
2_08.txt	79	[1, 2, 3, 4, 5, 7, 8, 9, 10, 6]	79	[8, 7, 3, 9, 10, 5, 4, 1, 6, 2]	79	[8, 2, 7, 1, 3, 9, 10, 4, 5, 6]
2_09.txt	59	[1, 2, 3, 4, 5]	59	[2, 3, 4, 5, 1]	59	[1, 4, 2, 3, 5]
2_10.txt	120	[1, 2, 3, 4, 5, 6, 7, 9, 10, 8]	120	[1, 3, 7, 6, 4, 9, 5, 2, 10, 8]	120	[1, 9, 10, 4, 2, 5, 3, 7, 6, 8]
3_04.txt	98	[3, 6, 1, 2, 4, 7, 5]	136	[3, 6, 1, 4, 7, 5, 2]	98	[3, 6, 2, 1, 4, 7, 5]
3_05.txt	98	[6, 2, 4, 9, 3, 1, 5, 7, 10, 8]	110	[6, 2, 9, 5, 3, 7, 10, 8, 4, 1]	98	[6, 4, 2, 9, 7, 10, 5, 3, 8, 1]
3_06.txt	71	[1, 4, 3, 5, 6, 7, 2]	80	[4, 5, 6, 3, 7, 2, 1]	71	[7, 3, 4, 5, 1, 6, 2]
3_07.txt	109	[3, 2, 5, 6, 1, 4, 7, 8]	127	[5, 3, 6, 4, 7, 8, 1, 2]	109	[5, 3, 6, 4, 2, 7, 8, 1]
3_08.txt	101	[6, 1, 2, 8, 3, 4, 9, 10, 5, 7]	114	[6, 9, 1, 8, 4, 10, 3, 5, 2, 7]	103	[9, 1, 4, 6, 10, 2, 8, 3, 5, 7]
3_09.txt	88	[7, 1, 3, 4, 5, 6, 8, 2, 9]	111	[7, 6, 3, 5, 1, 2, 8, 9, 4]	88	[7, 9, 6, 5, 1, 3, 2, 4, 8]
3_10.txt	56	[1, 2, 3, 4, 5, 6]	62	[3, 4, 2, 1, 6, 5]	56	[6, 1, 3, 2, 4, 5]

Aby móc porównać wszystkie algorytmy można sprawdzać jedynie wartości cmax na 2 lub 3 maszynach.

3 Podsumowanie

1. Przegląd zupełny jest najwolniejszym algorytmem, jednakże znajduje on zawsze najkrótszy możliwy czas, gdyż przeszukuje on wszystkie możliwe permutacje danych zadań. Już w przypadku powyżej 10 zadań algorytm ten działał wolno.
2. Algorytm Johnsona jest szybszym algorytmem od przeglądu zupełnego, aczkolwiek nie jest stworzony dla więcej niż 3 maszyn
3. Algorytm Johnsona w przypadku 3 maszyn znajduje większy cmax od pozostałych. Najprawdo-

podobniej jest to spowodowane tworzeniem maszyn wirtualnych, a nie faktycznym przeszukiwaniu wszystkich zadań na 3 maszynach.

4. Algorytm NEH znajduje taki sam czas jak w przypadku przeglądu zupełnego, jest od niego szybszy i może być stosowany do wielu maszyn oraz wielu zadań.
5. Przeprowadzenie wszystkich instancji ta_{max} potwierdza poprawność algorytmu NEH, gdyż c_{max} oraz kolejność jest zgodna z tą podaną w NEH.demo.