ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1980
Seria: AUTOMATYKA z. 54 Nr kol. 650

Jacek Błażewicz
Politechnika Poznańska

NOWE ALGORYTMY SZEREGOWANIA ZADAŃ NA DEDYKOWANYCH MASZYNACH

Streszczenie. Praca poświęcona jest analizie złożoności obliczeniowej deterministycznych problemów szeregowania zadań na maszynach dedykowanych, przy założeniu, że zadania mogą potrzebować do swego wykonania także dodatkowych ograniczonych zasobów.

1. Wstęp

Problematyka szeregowania dotyczy, najogólniej mówiąc, zagadnień określania kolejności wykonania zadań (operacji obróbki detali, operacji montażowych, transportowych czy obliczeniowych w systemach komputerowych) przez dany zbiór maszyn (obrabiarek, maszyn budowlanych, środków transportowych lub procesorów systemów cyfrowych). Zagadnienia te, jeśli chodzi o problemy deterministyczne, są przedmiotem badań od przeszło dwudziestu lat, jednak istotny wzrost zainteresowania tymi problemami nastąpił w ostatnich kilku latach. Fakt ten wynika zarówno z coraz pilniejszej potrzeby optymalnego wykorzystania czasu pracy maszyn i urządzeń, jak i z rozwoju teorii złożoności obliczeniowej (czasowej) problemów decyzyjnych (którymi w odpowiednim sformułowaniu są także problemy szeregowania). W teorii tej wyróżnia się, ogólnie mówiąc, klasy problemów "łatwych", dla których istnieją algorytmy optymalne o czasie wykonywania ograniczonym od góry przez wielomian od rozmiaru problemu, i "trudnych" ("NP-zupełnych"), dla których algorytmy takie najprawdopodobniej nie istnieją. Stąd udowodnienie przynależności danego problemu do klasy problemów "trudnych" zwalnia nas z konieczności poszukiwania algorytmu wielomianowego optymalnego, usprawiedliwiając poszukiwanie algorytmów wielomianowych przybliżonych (suboptymalnych), bądź też nieefektywnych (wykładniczych) algorytmów optymalnych.

Analizie złożoności obliczeniowej problemów szeregowania zadań na maszynach dedykowanych, to jest różniących się spełnianymi funkcjami, poświęcona będzie niniejsza praca. Rozpatrzymy problemy szeregowania przy ograniczeniach ze strony dodatkowych zasobów (którymi mogą być np. dodatkowe narzędzia, pomocniczy sprzęt, obsługa ludzka o określonych kwalifikacjach, a w systemach komputerowych - pamięć operacyjna, urządzenia we/wy, pamięci zewnętrzne czy niektóre środki programowe). Problemy szeregowania z ograniczeniami zasobowymi są ostatnio coraz częściej rozpatrywane ze względu

na ich oczywiste znaczenie praktyczne [2,3,5,6,8,9,10].

2. Podstawowe definicje

Będziemy rozpatrywać zbiór n zadań $Z=\{Z_1,Z_2,\ldots,Z_n\}$, zbiór m maszyn $M=\{M_1,M_2,\ldots,M_m\}$ oraz zbiór s rodzajów dodatkowych zasobów $R=\{R_1, R_2,\ldots,R_s\}$, przy czym zasób R_l jest dostępny w liczbie m_l jednostek. W danej chwili każda maszyna może wykonywać co najwyżej jedno zadanie. Maszyny są dedykowane, to znaczy różnią się spełnianymi funkcjami i pracują w ciągu technologicznym. W tym przypadku wyróżnia się trzy systemy obsługi zadań: przepływowy, otwarty i ogólny. W systemie przepływowym każde zadanie ze zbioru Z musi być wykonywane w tej samej kolejności przez wszystkie maszyny ze zbioru M. W systemie otwartym każde zadanie także musi być wykonywane przez wszystkie maszyny, lecz kolejność wykonywania nie jest narzucona. Natomiast w systemie ogólnym, podzbiór maszyn mających wykonać dane zadanie, jak i kolejność wykonywania każdego zadania są dowolne, choć określone. Określimy teraz parametry charakteryzujące zadanie.

Każde zadanie dzieli się na operacje $O_{j1},O_{j2},\ldots,O_{jk_j}$, z których każda może żądać innej maszyny. Przy tym dla systemu przepływowego liczba operacji przypadających na zadanie jest równa liczbie maszyn m, a kolejność ich wykonywania jest taka, że operacja O_{j1} jest wykonywana na maszynie M_1, operacja O_{j2} - na maszynie M_2 itd., oraz wykonanie operacji $O_{j,i-1}$ musi poprzedzać wykonanie operacji O_{ji}, $i=2,3,\ldots,m$. W przypadku systemu otwartego na wykonanie jednego zadania składa się także m operacji, identyczne jest także ich przyporządkowanie do maszyn. Kolejność wykonania operacji jest natomiast dowolna i nieokreślona. W przypadku systemu ogólnego zarówno liczba operacji przypadających na zadanie, jak i przyporządkowanie ich do maszyn i kolejność wykonywania są dowolne, lecz muszą być a priori określone.

Poza tym w ogólności zadanie $Z_j \in Z$ jest scharakteryzowane przez następujące dane:

1. Wektor żądań zasobowych - $R(Z_j)=[R_1(Z_j), R_2(Z_j),\ldots,R_s(Z_j)]$, gdzie $0 \leq R_l(Z_j) \leq m_l$, $l=1,2,\ldots,s$, oznacza liczbę jednostek zasobu R_l potrzebną do wykonania zadania Z_j. Jeśli żądania zasobowe operacji składających się na jedno zadanie różnią się między sobą, żądania te dla każdej operacji O_{ji} muszą być przedstawione w postaci wektora $R(O_{ji})=[R_1(O_{ji}), R_2(O_{ji}),\ldots,R_s(O_{ji})]$.
2. Wektor czasów wykonywania - $p_j=p_{j1},p_{j2},\ldots,p_{jm}$, gdzie p_{ji} jest czasem wykonywania zadania Z_j na maszynie M_i.

Zadania w rozpatrywanych problemach są niepodzielne i niezależne.

Przejdźmy teraz do definicji uszeregowania, uszeregowania optymalnego, problemu szeregowania i algorytmu szeregowania.

Uszeregowaniem będziemy nazywali takie przyporządkowanie w czasie ma-

szyn ze zbioru M i dodatkowych zasobów ze zbioru R do zadań ze zbioru Z, dla którego spełnione są następujące warunki:

- w każdej chwili każda maszyna wykonuje co najwyżej jedno zadanie;
- wszystkie zadania zostaną wykonane;
- dla każdego $t \geq 0$, $\sum_{Z_j \in A(t)} R_l(Z_j) \leq m_l$, gdzie $A(t)$ jest zbiorem zadań wykonywanych w chwili t; $l=1,2,\ldots,s$;

Uszeregowaniem optymalnym nazywać będziemy uszeregowanie minimalizujące długość uszeregowania $C_{max}=\max_j \{C_j\}$.

Przez problem szeregowania Π rozumieć będziemy uporządkowany ciąg parametrów podzielonych na trzy podciągi $\alpha|\beta|\gamma$ /por. [13]/, przy czym parametry nie muszą mieć nadanych wartości.

Ustalając wartości wszystkich parametrów danego problemu szeregowania, otrzymujemy konkretny problem szeregowania, który oznaczymy przez $I \in D_\Pi$.

Algorytmem szeregowania dla problemu Π nazywać będziemy dowolną procedurę, która dla każdego konkretnego problemu $I \in D_\Pi$ znajduje uszeregowanie, jeśli ono istnieje.

Optymalnym algorytmem szeregowania dla problemu Π nazywać będziemy algorytm szeregowania, który zapewnia minimalizację kryterium γ dla każdego znalezionego uszeregowania.

Jak wiadomo, udowadniając "trudność" /NP-zupełność/problemu szeregowania należy sformułować w postaci problemu decyzyjnego, to znaczy w postaci pytania, na które odpowiedź brzmi "tak" lub "nie". Łatwo zauważyć, że jeśli problem szeregowania można rozwiązać w wielomianowym czasie, to można rozwiązać w wielomianowym czasie również odpowiadający mu problem decyzyjny. Z drugiej strony, jeśli problem decyzyjny jest obliczeniowo "trudny" /NP-zupełny/, to "trudny" jest także odpowiadający mu problem szeregowania. Wynika z tego, że w celu wykazania "łatwości" problemu decyzyjnego wystarczy wykazać "łatwość" odpowiedniego problemu szeregowania, natomiast dla wykazania "trudności" problemu szeregowania wystarczy wykazać "trudność" odpowiadającego mu problemu decyzyjnego. W ten sposób będziemy postępować w dalszej części pracy, jakkolwiek wprowadzone poniżej definicje klas złożonościowych będą dotyczyły problemów decyzyjnych.

Zdefiniujemy teraz klasy P i NP problemów decyzyjnych [1,7,11,12].

Klasę P tworzą wszystkie problemy decyzyjne, które w co najwyżej wielomianowym czasie rozwiązuje /w sensie znalezienia rozwiązania konkretnego problemu/ deterministyczna maszyna Turinga, natomiast klasę NP - wszystkie problemy decyzyjne, które w co najwyżej wielomianowym czasie rozwiązuje niedeterministyczna maszyna Turinga.

W celu zdefiniowania najbardziej interesującej klasy problemów decyzyjnych, tzw. NP-zupełnych, wprowadza się definicję transformacji wielomianowej.

Transformacją wielomianową problemu Π_2 do problemu Π_1 /zapisujemy

$\Pi_2 \propto \Pi_1$ / nazywa się funkcję $f: D_{\Pi_2} \rightarrow D_{\Pi_1}$, która spełnia następujące warunki:

1) czas obliczenia funkcji f przez DTM dla każdego konkretnego problemu $I_2 \in D_{\Pi_2}$ jest ograniczony od góry przez wielomian od rozmiaru konkretnego problemu I_2, $N(I_2)$;
2) dla każdego konkretnego problemu $I_2 \in D_{\Pi_2}$ odpowiedź brzmi "tak" wtedy i tylko wtedy, gdy dla konkretnego problemu $f(I_2)$ odpowiedź brzmi również "tak".

Problem decyzyjny Π_1 jest NP-zupełny jeśli $\Pi_1 \in NP$ i dla każdego innego problemu decyzyjnego $\Pi_2 \in NP$, $\Pi_2 \propto \Pi_1$.

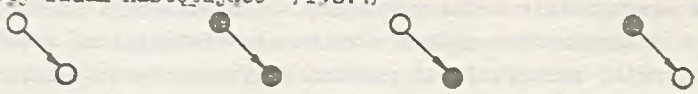
W następnych rozdziałach rozpatrzymy złożoność obliczeniową problemów szeregowania zadań dla trzech systemów obsługi: przepływowego, otwartego, i ogólnego.

3. System przepływowy

Dla problemu $P\,2|res111, \tau_j=1|C_{max}$ optymalny jest następujący algorytm [5] .

Algorytm 1

1. Podziel wszystkie zadania na cztery typy w zależności od żądań zasobowych operacji, składających się na poszczególne zadania. Oznacz te cztery typy zadań następująco /rys.1/

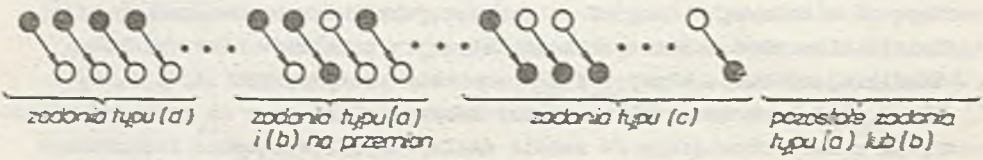


(a) (b) (c) (d)

Rys.1

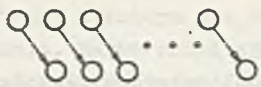
(a) - $R_1(O_{j1})=0$, $R_1(O_{j2})=0$, (b) - $R_1(O_{j1})=1$, $R_1(O_{j2})=1$.
(c) - $R_1(O_{j1})=0$, $R_1(O_{j2})=1$, (d) - $R_1(O_{j1})=1$, $R_1(O_{j2})=0$.

2. Optymalne uszeregowanie można skonstruować w następujący sposób /por. rys.2/.



Rys.2

- Przydziel do maszyn wszystkie zadania typu (d).
- Przydzielaj na przemian zadania typu (a) i (b) .
- Przydziel wszystkie zadania typu (c)
- Pozostałe zadania typu (a) lub (b) przydzielane są następująco:
  d) jeśli pozostały tylko zadania typu (a) , przydzielaj je kolejno do maszyn w sposób pokazany na rys.3.



Rys.3

Rys.4

β) jeśli pozostały tylko zadania typu (b), przydzielaj je do maszyn kolejno w sposób pokazany na rysunku 4.

Łatwo można sprawdzić, że złożoność obliczeniowa algorytmu 1 jest $O(n)$.

Złożoność obliczeniowa problemów $F2|res\cdot 11,\tau_j=1|C_{max}$ oraz $F|res1\cdot 1,\tau_j=1|C_{max}$ jest sprawą otwartą, natomiast problem $F2|res111|C_{max}$ jest NP-zupełny, co wykazano w [4]

4. System otwarty

Problem $O2|res,\ldots,\tau_j=1|C_{max}$ rozwiązuje optymalnie poniższy algorytm 2

Algorytm 2

1. Utwórz nieskierowany graf G, o 2n wierzchołkach odpowiadających operacjom. Połącz łukami wierzchołki odpowiadające operacjom wykonywanym na pierwszej maszynie z wierzchołkami odpowiadającymu operacjom wykonywanym na drugiej maszynie wtedy i tylko wtedy, gdy żądania zasobowe operacji połączonych łukiem nie przekraczają ograniczeń zasobowych, a ponadto operacje te nie wchodzą w skład tego samego zadania.
2. Dla tak określonego grafu G znajdź maksymalne skojarzenie, to znaczy zbiór K złożony z maksymalnej liczby łuków tego grafu, z których żadne dwa nie mają wspólnego wierzchołka.
3. Podstaw minimalną długość uszeregowania $C^*_{max} = 2n-|K|$, Skonstruuj optymalne uszeregowanie, w którym równolegle wykonywane są operacje połączone łukami należącymi do zbioru K, a pojedynczo pozostałe operacje.

Złożoność obliczeniowa algorytmu 2 jest $O(n^3)$. Złożoność obliczeniowa pozostałych problemów szeregowania zadań dla tego systemu jest sprawą otwartą.

5. System ogólny

W przypadku systemu ogólnego można udowodnić, że wszystkie problemy szeregowania zadań są NP-zupełne, gdyż NP-zupełny jest najprostszy z tych problemów [5].

Twierdzenie 1.

Problem $J2|res111,\tau_j=1|C_{max}$ jest NP-zupełny.

Dowód

Oznaczmy przez Π_1 badany problem, a przez Π_2 problem TRÓJPODZIAŁU

/por. [1] /.

Mając dany konkretny problem $I_2 \in D_{\pi_2}$, określony przez wartości elementów $s(a_1), s(a_2), \ldots, s(a_{3q})$ i wartość ograniczenia B, dane odpowiadającego mu konkretnego problemu $I_1 \in D_{\pi_1}$ tworzy się w następujący sposób.

- Określ 3q+1 zadań o jednostkowych czasach wykonywania, przy czym 3q zadań tworzących zbiór E_j ma postać

$$O_2', \ (O_1, O_2)^{s(a_j)}, \ O_1', \qquad j=1,2,\ldots,3q,$$

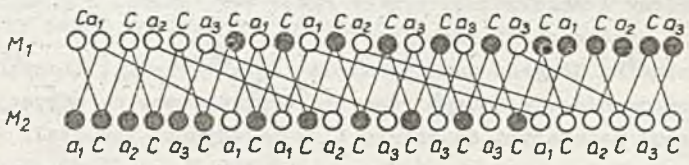
gdzie O_1, O_1' - operacje wykonywane na maszynie M_1,
O_2, O_2' - operacje wykonywane na maszynie M_2,
$R_1(O_1)=R_1(O_2)=0$,
$R_1(O_1')=R_1(O_2')=1$,
$(O_1,O_2)^{s(a_j)}$ - oznacza wykonanie operacji (O_1,O_2), $s(a_j)$ razy;

ostatnie zadanie C składa się z q równych, szeregowo połączonych łańcuchów operacji postaci

$$[(O_1,O_2')^3, \ (O_1',O_2')^{B-3}, \ (O_1',O_2)^3] \ ;$$

- Liczba maszyn m=2;
- Wartość funkcji kryterialnej $y_1=2q(B+3)$.

Jeśli istnieje rozwiązanie konkretnego problemu, to znaczy istnieje podział danego zbioru na trójelementowe podzbiory o żądanej właściwości, to zadania w odpowiadającym mu konkretnym problemie $I_1 \in D_{\pi_1}$, należy przydzielać tak, jak pokazano na rysunku 5, na którym przyjęto B=9. Dłu -



Rys 5

gość łańcucha operacji składających się na wykonanie zadania C jest równa 2q(B+3), czyli żądanej długości uszeregowania. Zatem w każdej jednostce czasu któraś z operacji tego łańcucha przydzielona jest do maszyny. Ponieważ istnieje podział zbioru elementów na podzbiory trójelementowe i suma wartości elementów w każdym podzbiorze równa jest B, więc zadania odpowiadające elementom wchodzącym w skład jednego podzbioru można przydzielić do maszyn w przedziale czasu o długości 2(B+3), w którym przydzielony jest jeden z łańcuchów tworzących zadanie C. Ponieważ łańcuchów tych jest q, zatem długość uszeregowania jest równa 2q(B+3).

Jeśli teraz istnieje uszeregowanie o długości równej 2q(B+3) dla konkretnego problemu $I_1 \in D_{\pi_1}$, to w uszeregowaniu tym zadanie C musi być wykonywane tak, jak pokazano na rysunku 5 (jedna operacja po drugiej). Zadania ze zbioru E_j zostały tak dobrane, że jeśli $C_{max}=2q(B+3)$, to

równolegle z każdym z q łańcuchów zadania C muszą być wykonywane dokładnie 3 zadania. Łatwo można sprawdzić, że zadania te odpowiadają elementom, których suma wartości jest równa B. Istnieje zatem rozwiązanie o żądanej właściwości dla konkretnego problemu $I_2 \in D_{\Pi_2}$.

Kończy to dowód, ponieważ problem Π_2 jest NP-zupełny [10].

□

LITERATURA

[1] Aho,A.V., J.E.Hopcroft, J.D.Ullman : The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass, 1974.

[2] Błażewicz,J., : Deadline scheduling of tasks with ready times and resource constraints, Information Processing Letters 8, No2, 1979, 60-63.

[3] Błażewicz,J., : Solving the resource constrained deadline scheduling problem via reduction to the network flow problem, European J. of Operational Research, 4, No2, 1980.

[4] Błażewicz,J. : Złożoność obliczeniowa algorytmów i problemów szeregowania zadań. Wyd. Politechniki Poznańskiej, Seria Rozprawy, 1980.

[5] Błażewicz,J., J.K.Lenstra, A.H.G.Rinnooy Kan : Scheduling subject to resource constraints : classification and complexity, Discrete Applied Mathematics, to appear.

[6] Błażewicz,J., J.Węglarz : Scheduling under resource constraints - achievements and prospects, in M.Arato, A.Butrimenko and E.Gelenbe /eds./ Performance of Computer Systems, North Holland, Amsterdam, 1979, pp.181-199.

[7] Cook, S.A., : The complexity of theorem proving procedures, Proc. 3rd ACM Symposium on Theory of Computing, 1971, pp.151-158.

[8] Garey,M.R., R.L.Graham : Bounds for multiprocessor scheduling with resource constraints SIAM J. on Computing 4, 1975, pp.187-200.

[9] Garey,M.R., R.L.Graham, D.S.Johnson, A.C.-C. Yao, Resource constrained scheduling as generalized bin packing, J.Combinatorial Theory Ser.A 21, 1976, pp.257-298.

[10] Garey,M.R., D.S.Johnson : Complexity results for multiprocessor scheduling under resource constraints, SIAM J. on Computing 4, 1975, pp. 397-411.

[11] Garey,M.R., D.S.Johnson : Computers and Intractability : A Guide to the Theory of NP-Completeness, W.H.Freeman, San Francisco, 1979.

[12] Karp,R.M. : Reducibility among combinatorial problems, in R.E.Miller and J.W.Thatcher /eds./ Complexity of Computer Computation, Plenum Press, New York, 1972, pp.85-104.

[13] Węglarz,J. : Złożoność obliczeniowa problemów szeregowania zadań na równoległych maszynach z uwzględnieniem dodatkowych zasobów, Prace II Krajowej Konferencji Automatyzacji Dyskretnych Procesów Przemysłowych, Kozubnik, 1980.

НОВЫЕ АЛГОРИТМЫ СОСТАВЛЕНИЯ РАСПИСАНИЯ ЗАДАЧ НА МАШИНАХ КОНВЕЕРНОГО ТИПА

Р е з ю м е

В работе рассмотрено вычислительную сложимость детерминистических проблем составления расписания задач на машинах конвеерного типа с учётом дополнительных ресурсов.

THE NEW ALGORITHMS OF TASKS'SCHEDULING ON DEDICATED MACHINES

S u m m a r y

The paper is devoted to the analysis of the computational complexity of deterministic problems of tasks'scheduling on dedicated machines, in case of the tasks demanding for their processing some of additional limited resources.