ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1998
Seria: AUTOMATYKA z. 123 Nr kol. 1389

Zbigniew BUCHALSKI
Politechnika Wrocławska

SZEREGOWANIE ZADAŃ NA RÓŻNYCH MASZYNACH RÓWNOLEGŁYCH Z ROZDZIAŁEM OGRANICZONYCH ZASOBÓW

Streszczenie. W pracy rozważany jest problem czasowo-optymalnego szeregowania zadań i rozdziału zasobów na różnych maszynach równoległych. Założono, że zadania są niezależne i niepodzielne. Liczba zadań do wykonania jest większa od liczby maszyn. Sformułowano model matematyczny problemu i podano algorytm heurystyczny. Przedstawiono wyniki eksperymentów obliczeniowych.

TASK SCHEDULING ON DIFFERENT PARALLEL MACHINES WITH ALLOCATION OF LIMITED RESOURCES

Summary. In the paper we describe and solve time-optimal tasks scheduling and resources allocation problem on different, parallel machines. We assume that all tasks are independent nonpreemtive and number of tasks is greater than number of machines. The mathematical model of the problem is formulated and heuristic algorithm is presented. Some results of executed numerical experiments are presented.

1. Wprowadzenie

Problem szeregowania zadań na maszynach równoległych z równoczesnym rozdziałem zasobów bardzo często spotykany jest w różnego rodzaju złożonych procesach produkcyjnych. Zasobami podzielnymi w sposób ciągły najczęściej są gaz, energia elektryczna, paliwo. W wieloprocesorowych systemach komputerowych spotykamy się z szeregowaniem programów na procesorach oraz przydziałem zasobów w postaci stron pamięci operacyjnej do procesorów.

Problematyka ta rozpatrywana była między innymi w [1,2,7,8,9,10]. Niniejsza praca jest kontynuacją wcześniejszych prac autora [3,4,5,6] i dotyczy zagadnienia czasowo-optymalnego przydziału zasobów nieodnawialnych podzielnych w sposób ciągły do m różnych maszyn równoległych.

W rozdziale drugim formułuje się problem oraz podany jest jego model matematyczny. Ponieważ zagadnienie to należy do klasy problemów NP-trudnych, w rozdziale trzecim przedstawiony został algorytm heurystyczny rozwiązujący postawiony problem oraz podane są wyniki eksperymentów obliczeniowych przeprowadzonych na tym algorytmie.

2. Sformułowanie problemu, model matematyczny

Dany jest system maszyn równoległych, o którym zakładamy, że:

- posiada m różnych maszyn. Zbiór maszyn oznaczmy przez M:

$$M = \{M_1, M_2, \ldots, M_k, \ldots, M_m\}, \quad k = 1,2,\ldots,m;$$

- maszyny te mają wykonać n niezależnych, niepodzielnych zadań. Zbiór zadań oznaczmy przez Z:

$$Z = \{1,2,\ldots,n\},$$

- liczba zadań do wykonania jest większa od liczby maszyn $n>m$;
- wszystkie zadania są gotowe do realizacji w chwili zerowej.

Niech N oznacza globalną ilość zasobów nieodnawialnych, a przez u_k oznaczmy tę część zasobów, które zostaną przydzielone maszynie M_k w trakcie wykonywania zadań uszeregowanych na tej maszynie. Ograniczenie dotyczące zasobów jest następujące:

$$\sum_{k=1}^{m} u_k \le N, \quad u_k \ge 0, \quad 1 \le k \le m.$$

Czas wykonywania i-tego zadania na maszynie M_k określony jest przez funkcję $T_i(u_k, k)$:

$$T_i(u_k, k) = a_{ik} + \frac{b_{ik}}{u_k}, \quad u_k \in \{1,2,\ldots,N\}, \quad 1 \le k \le m, \quad 1 \le i \le n. \tag{1}$$

Parametry $a_{ik} > 0$ i $b_{ik} > 0$ charakteryzują i-te zadanie i maszynę M_k.

Należy znaleźć takie uszeregowanie zadań na maszynach i taki przydział ograniczonych zasobów do maszyn równoległych, aby minimalizować czas wykonania całego zbioru zadań.

Jeżeli oznaczymy przez $Z_k \subset Z$ zbiór zadań uszeregowanych na maszynie M_k, to kryterium optymalności Q będzie miało następującą postać:

$$Q = \min_{\substack{Z_1, Z_2, \ldots, Z_m \\ u_1, u_2, \ldots, u_m}} \max_{1 \le k \le m} \left\{ \sum_{i \in Z_k} T_i(u_k, k) \right\} \tag{2}$$

przy ograniczeniach:

(i) $\quad Z\ Z_r \cap Z_s = \phi, \quad r,s = 1,2,\ldots,m, \quad r \ne s, \quad \bigcup_{k=1}^{m} Z_k = Z$

(ii) $\quad \sum_{k=1}^{m} u_k \le N,$

(iii) $\quad u_1, u_2, \ldots, u_m$ - całkowite dodatnie.

Dla uproszczenia problemu przyjmiemy najpierw, że zasoby nieodnawialne $u_1, u_2, \ldots, u_m$ są typu ciągłego. Przy tym założeniu wyznaczymy rozwiązanie optymalne, a następnie zaokrąglimy otrzymane wartości zasobów do najbliższych liczb naturalnych. Tak więc kryterium optymalności Q będzie miało następującą postać optymalizacji dyskretno-ciągłej:

$$Q = \min_{\substack{Z_1, Z_2, \ldots, Z_m \\ u_1, u_2, \ldots, u_m}} \max_{1 \le k \le m} \left\{ \sum_{i \in Z_k} T_i'(u_k, k) \right\} \tag{3}$$

Przy ograniczeniach:

(i) $Z_r \cap Z_s = \phi, \quad r, s = 1,2,\ldots,m, \quad r \neq s, \quad \bigcup_{k=1}^{m} Z_k = Z$

(ii) $\sum_{k=1}^{m} u_k \le N, \quad u_k \ge 0, \quad k = 1,2,\ldots,m,$

gdzie: $T_i' : [0, N] \times \{1,2,..,m\} \to R^+$ jest rozszerzeniem funkcji $T_i : \{1,2,\ldots,N\} \times \{1,2,..,m\} \to R^+$ i określone jest przez funkcję:

$$T_i'(u_k, k) = a_{ik} + \frac{b_{ik}}{u_k}, \quad u_k \in [0, N], \quad 1 \le k \le m, \quad 1 \le i \le n. \tag{4}$$

Do rozwiązania postawionego problemu pomocny będzie następujący lemat:

LEMAT 1

Jeżeli $u_k^*, Z_k^*, k = 1,2,\ldots,m$ są rozwiązaniami zadania (3), to:

(i) $\sum_{k=1}^{m} u_k^* = N; \quad u_k^* > 0, \quad k : Z_k^* \neq \phi, \quad k = 1,2,\ldots,m;$

$u_k^* = 0, \quad k : Z_k^* = \phi, \quad k = 1,2,\ldots,m;$

(ii) $\sum_{k \in Z_k^*}^{m} T_k^*(u_k^*, k) = const, \quad k : Z_k^* \neq \phi, \quad k = 1,2,\ldots,m.$

Dla ustalonych zbiorów $Z_1, Z_2, \ldots, Z_m$ spełniających ograniczenie 3(i) niech:

$$F(Z_1, Z_2, \ldots, Z_m) \triangleq \sum_{i \in Z_1} a_{i1} + \frac{\sum_{i \in Z_1} b_{i1}}{\overline{u}_1} \tag{5}$$

gdzie $\overline{u}_1$ jest rozwiązaniem następującego układu równań:

$$\begin{cases} \sum_{i \in Z_k} a_{ik} + \dfrac{\sum_{i \in Z_k} b_{ik}}{\overline{u}_k} = \sum_{i \in Z_{k+1}} a_{i,k+1} + \dfrac{\sum_{i \in Z_{k+1}} b_{i,k+1}}{\overline{u}_{k+1}}, k = 1,2,\ldots,m-1 \\ \sum_{k=1}^{m} \overline{u}_k = N, \overline{u}_k > 0, k = 1,2,\ldots,m; \end{cases}$$

w układzie tym zmiennymi są $\overline{u}_1, \overline{u}_2, \ldots, \overline{u}_m$.

Z def. (5) oraz z LEMATU 1 wynika, że zadanie (2) można przedstawić w nastepującej postaci:

$$\min_{Z_1, Z_2, \ldots, Z_m} F(Z_1, Z_2, \ldots, Z_m), \tag{6}$$

przy ograniczeniach

(i) $Z_r \cap Z_s = \phi, \quad r, s = 1,2,\ldots,m, \quad r \neq s,$

(ii) $\bigcup_{k=1}^{m} Z_k = Z$

Jeżeli $Z_1^*, Z_2^*, \ldots, Z_m^*$ jest rozwiązaniem zadania (6), to $u_k^*, Z_k^*, k = 1,2,\ldots,m,$ gdzie

$$u_k^* = \begin{cases} \dfrac{\sum_{i \in Z_k^*} b_{ik}}{F(Z_1^*, Z_2^*, \ldots, Z_m^*) - \sum_{i \in Z_k^*} a_{ik}}; & k : Z_k^* \neq \phi, \quad 1 \leq k \leq m, \\ 0 & ; \ k : Z_k^* = \phi, \quad 1 \leq k \leq m \end{cases}$$

jest rozwiązaniem zadania (2).

3. Algorytm heurystyczny

Przed realizacją wstępnego uszeregowania zadań na maszynach należy ustalić ilość zasobów u_k, $k = 1,2,\ldots,m$ dostępnych dla każdej z maszyn M_k, $k = 1,2,\ldots,m$ (wartości te wybieramy z ogólnej puli N zasobów nieodnawialnych, jaką dysponujemy).

Formuła przydziału zasobów wygląda następująco:

- zakładamy, że maszyną najszybszą jest maszyna pierwsza M_1, a maszyną najwolniejszą maszyna ostatnia, czyli M_m,
- przyjmujemy istnienie tzw. współczynnika podziału zasobów (oznaczmy go przez α), różnicującego maszyny pod względem szybkości realizacji zadań.

Jeżeli maszynie najwolniejszej przydzielimy u_m jednostek zasobów, to dla pozostałych maszyn przydział zasobów będzie wyglądał następująco:

$$\begin{aligned} u_1 &= (m-1) \cdot \alpha \cdot u_m \\ u_2 &= (m-2) \cdot \alpha \cdot u_m \\ &\vdots \\ u_k &= (m-k) \cdot \alpha \cdot u_m \\ &\vdots \\ u_{m-1} &= [m-(m-1)] \cdot \alpha \cdot u_m = \alpha \cdot u_m \end{aligned} \tag{7}$$

Jak wiadomo:

$$\sum_{k=1}^{m} u_k = N. \tag{8}$$

Rozwijając sumę (8) oraz wprowadzając do niej parametr α, otrzymamy:

$$(m-1)\cdot\alpha\cdot u_m + (m-2)\cdot\alpha\cdot u_m + \ldots + (m-k)\cdot\alpha\cdot u_m + \ldots + 2\cdot\alpha\cdot u_m + \alpha\cdot u_m + u_m = N \tag{9}$$

Z zależności (9) wyliczamy wartość u_m dla maszyny "bazowej", najwolniejszej M_m :

$$u_m + \sum_{k=1}^{m-1}\left[(m-k)\cdot\alpha\cdot u_m\right] = N,$$

a zatem

$$u_m = \frac{N}{1+\sum_{k=1}^{m-1}\left[(m-k)\cdot\alpha\right]} \tag{10}$$

Pozostałe maszyny otrzymają liczbę zasobów określoną zależnością:

$$u_k = (m-k)\cdot\alpha\cdot u_m, \quad k = 1,2,\ldots,m-1. \tag{11}$$

Po zakończeniu procesu przydziału zasobów do kolejnych maszyn dokonujemy podziału zbioru zadań Z na grupy (o łącznej ich liczbie równej liczbie maszyn) w taki sposób, aby każda z nich skupiała w sobie jednakową liczbę zadań do wykonania na pojedynczej maszynie. Oznaczmy tę liczbę zadań przez L. Pozostałe zadania nie przydzielone do tych równolicznych grup, przydzielone zostaną maszynom w końcowej fazie szeregowania, począwszy od najszybszej maszyny w kierunku maszyn wolniejszych.

Poszczególne kroki algorytmu heurystycznego są następujące:

Krok 1. Oblicz czasy wykonania zadań na poszczególnych maszynach dla $u_i = N$, $i = 1,2,\ldots,n$. Czas realizacji i-tego zadania na maszynie M_k będzie wynosił $T_i'(N,k) = a_{ik} + \frac{b_{ik}}{N}$, $i = 1,2,\ldots,n, k = 1,2,\ldots m$. Podaj liczbę L zadań w każdej z m grup.

Krok 2. Dla określonego współczynnika podziału zasobów α przydziel zasoby poszczególnym maszynom wyliczone z zależności (10) i (11).

Krok 3. W zbiorze zadań znajdź zadanie, którego czas $T_i'(N,k)$, $i = 1,2,\ldots,n, k = 1,2,\ldots,m$ jest największy. Uszereguj to zadanie na bieżącej maszynie. Jeżeli lista zadań do wykonania została wyczerpana, przejdź do **kroku 8**, w przeciwnym wypadku realizuj **krok 4**.

Krok 4. Sprawdź, jaka liczba zadań została uszeregowana na bieżącej maszynie. Jeżeli liczba ta osiągnęła wartość L, przejdź do **kroku 5**, jeżeli nie, to uszereguj następne zadania na maszynie bieżącej, skacząc do **kroku 3**.

Krok 5. Jeżeli zbiór maszyn się nie wyczerpał, uszereguj zadania na kolejnej maszynie, przechodząc z powrotem do **kroku 3**. Jeżeli natomiast zbiór maszyn już się wyczerpał i nie ma już nie uszeregowanych zadań na liście zadań, przejdź do **kroku 8.** W przypadku gdy zbiór maszyn już się wyczerpał, ale zostały jeszcze zadania nie uszeregowane, przejdź do **kroku 6.**

Krok 6. Uszereguj najdłuższe, nie przydzielone jeszcze żadnej maszynie zadanie do maszyny M_1.

Krok 7. Jeżeli lista zadań się nie wyczerpała, to uszereguj najdłuższe nie przydzielone jeszcze zadanie kolejnej maszynie. Krok ten powtarzaj aż do momentu wyczerpania się listy zadań do wykonania.

Krok 8. Wylicz czas realizacji zbioru zadań w utworzonym uszeregowaniu, następnie zwiększ wartość współczynnika podziału zasobów α.

Krok 9. Powtórz **kroki 1-8** dla pięciu zwiększających się kolejno wartości współczynnika α. Po pięciu próbach realizuj następny krok.

Krok10. Porównaj wartości czasów realizacji zbioru zadań z kolejnych prób. Wybierz najkrótszy czas realizacji zbioru zadań.

Powyższy algorytm wykonuje za każdym razem pięć prób znalezienia najlepszego, z punktu widzenia czasu realizacji zbioru zadań, rozwiązania. Po wykonaniu serii prób z różnymi wartościami współczynnika α, porównywane są ze sobą czasy realizacji zbioru zadań i wybierany jest najkrótszy z nich.

Na bazie przedstawionego algorytmu przeprowadzono eksperymenty obliczeniowe dla współczynnika podziału zasobów α zmieniającego się w przedziale [1.5,...,5.5] ze skokiem co 1.0. Parametry charakteryzujące i-te zadanie i k-tą maszynę a_{ik}, b_{ik} wylosowane zostały ze zbioru {10,...,99} przez generator o jednostajnym rozkładzie prawdopodobieństwa. Wyniki przedstawione zostały w tablicy 1.

Tablica 1

Wyniki eksperymentów obliczeniowych algorytmu heurystycznego

Liczba zadań n	Liczba maszyn m	Średni czas obliczeń [sek]	Mediana czasów obliczeń [sek]
20	4	3.1	3
20	6	3.7	4
20	8	4.2	4
40	4	3.9	4
40	6	6.9	7
40	8	9.2	9
60	4	8.1	8
60	6	11.3	11
60	8	14.2	14

LITERATURA

1. Błażewicz J., Cellary W., Słowiński R., Węglarz J.: Algorytmy sterowania rozdziałem zadań i zasobów w kompleksie operacji. Wydawnictwo Politechniki Poznańskiej, Poznań 1979.
2. Błażewicz J., Lenstra J.K., Rinnooy Kan A.H.G.: Scheduling subject to resource constraints: classification and complexity. Discrete Appl. Math., Mathemtisch Centrum, Amsterdam 1980.
3. Buchalski Z.: Some problem of time-optimal allocation of memory and tasks in multiprocessor computer systems. Polish Cybernetical Society, Vol. 2, Cybernetics in control and computer systems engineering, Warsaw 1985, pp. 41-49.
4. Buchalski Z.: Zagadnienie czasowo-optymalnego szeregowania zadań i rozdziału zasobów w systemie wielomaszynowym. Prace Konferencji Naukowo-Technicznej "Problematyka budowy i eksploatacji maszyn i urządzeń w ujęciu systemowym", AGH Kraków 1986, str. 62-69.
5. Buchalski Z.: Algorytm dla problemu szeregowania zadań na maszynach dla pewnych funkcji czasu wykonywania zadań. Zeszyty Naukowe Politechniki Śląskiej Nr 970, Seria Automatyka, z. 94, Gliwice 1988, str. 61-68.
6. Buchalski Z.: Zagadnienie przydziału zadań i zasobów do maszyn równoległych dla pewnych funkcji czasu wykonywania zadań. Wydawnictwa AGH, Automatyka, Półrocznik tom 1, zeszyt 1, Kraków 1997, str. 61-70.
7. Ishii H., Martel C., Masuda T., Nishida T.: A generalized uniform processor system. Oper. Res. Vol. 33, nr 2, 1985, pp. 346-362
8. Węglarz J.: Project Scheduling with Continously-Divisible Doubly Constrained Resources, Mgt. Sci, vol. 27, nr 3. 1981.

9. Węglarz J.: Sterowanie w systemach typu kompleks operacji. PWN, Warszawa-Poznań 1981.

10. Węglarz J.: Synthesis problems in allocating continuous, doubly constrained resources among dynamic activities, Operation Research 1990, pp. 715-730.

Recenzent: Prof. zw.Tadeusz Puchałka

Abstract

In the paper the problem of time-optimal tasks scheduling and resources allocation on different, parallel malchines is considered. Results obtained here are for some processing time function $T_i(u_k,k) = a_{ik} + \frac{b_{ik}}{u_k}$, $u_k \in \{1,2,\ldots,N\}$, $k = 1,2,\ldots,m$, $i = 1,2,\ldots,n$, where $a_{ik} > 0$, $b_{ik} > 0$ are parameters determined i-th task and k-th machines, u_k is number of resources allocated to M_k machines. We assume that all tasks are independent, nonpreemtive and their ready times are equal to zero. Number of tasks is greater than number of different, parallel machines. We also assume that the amount of resources available at each moment is constant. The purpose of optimization is to find such a schedule of tasks on parallel machines and such an allocation of limited nonrenewable resources among machines that schedule length criterion is minimized. Because our problem belongs to the class of NP-hard problems we propose a heuristic algorithm which employs some problem properties. Some results of executed numerical experiments are presented.