HEURYSTYCZNA PROCEDURA SZEREGOWANIA ZADAŃ I ROZDZIAŁU OGRANICZONYCH ZASOBÓW W SYSTEMIE MASZYN RÓWNOLEGŁYCH-OCENA EFEKTYWNOŚCI ALGORYTMU HEURYSTYCZNEGO

Zbigniew BUCHALSKI

Streszczenie: Artykuł dotyczy zagadnienia czasowo-optymalnego przydziału zasobu podzielnego w sposób ciągły i *n* zadań do *m* maszyn równoległych. Założono, że zadania są niezależne i niepodzielne. Dla zadanej funkcji czasu realizacji zadań sformułowano model formalny zagadnienia i zaproponowano pewien algorytm heurystyczny wyznaczający czasowo-optymalne szeregowanie zadań i przydział zasobów do maszyn równoległych. Przedstawiono wyniki badań numerycznych przeprowadzonych na tym algorytmie.

Słowa kluczowe: systemy maszyn równoległych, szeregowanie zadań, rozdział zasobów, algorytmy heurystyczne.

1. Wstęp

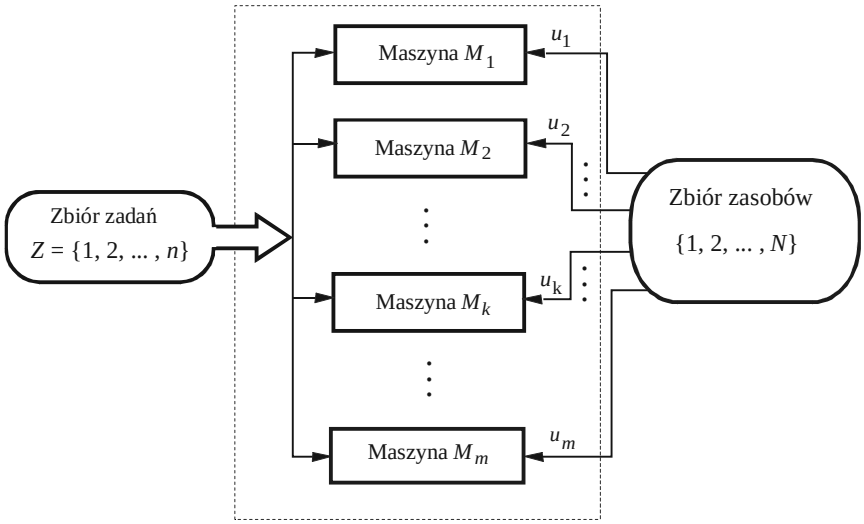
Odczuwalny od wielu lat gwałtowny rozwój równoległych systemów przetwarzania informacji pociągnął za sobą potrzebę rozwiązywania problemów czasowo-optymalnego szeregowania zadań i rozdziału zasobów [1, 2, 3, 4, 5]. Znaczenie praktyczne rozwiązywania tych problemów jest bezsporne w wielu dziedzinach życia. Problemy te mogą być związane przykładowo ze sterowaniem procesem produkcyjnym, wyznaczaniem kolejnych etapów montażu urządzeń, harmonogramowaniem zadań transportowych, itp.

Zadania optymalizacji zarówno dyskretnej, jak i ciągłej należą do klasy problemów bardzo trudnych zarówno z teoretycznego, jak i obliczeniowego punktu widzenia i najczęściej należą do klasy problemów *NP*- zupełnych [6, 7, 8]. W systemach maszyn równoległych spotykamy się z szeregowaniem zadań na maszynach oraz przydziałem zasobów do maszyn. Przy rozwiązywaniu tych problemów występują istotne trudności natury obliczeniowej w związku z czym eliminuje się z rozważań algorytmy dokładne, pozostawiając do zastosowania praktycznego jedynie algorytmy heurystyczne umożliwiające rozwiązanie postawionych problemów w krótkim czasie z zadowalającą dokładnością [9, 10, 11].

W niniejszym artykule przedstawiono pewien algorytm heurystyczny wyznaczający czasowo-optymalne harmonogramowanie rozdziału *n* zadań niezależnych niepodzielnych i jednostek zasobu nieodnawialnego podzielnego w sposób ciągły do *m* maszyn pracujących równolegle. Przedstawiono wyniki badań numerycznych przeprowadzonych na tym algorytmie dla losowo generowanych danych.

2. Sformułowanie problemu

Rozpatrzmy dyskretny system produkcyjny zawierający maszyny połączone równolegle przedstawiony na poniższym rysunku:



Rys. 1. System maszyn równoległych

Na system maszyn równoległych nakładamy następujące założenia:

(i) posiada m różnych maszyn $M = \{1, 2,..., k,... , m\}$, na których należy wykonać n niezależnych zadań $Z = \{1, 2,..., i,..., n\}$,

(ii) zadanie może być wykonywane na dowolnej maszynie i w trakcie jego wykonywania nie może być przerywane,

(iii) liczba zadań do wykonania jest większa od liczby maszyn $n > m$,

(iv) realizacja każdego z zadań na maszynach musi następować niezwłocznie po zakończeniu wykonywania poprzedniego zadania lub nastąpić w chwili zerowej, gdy zadanie realizowane jest jako pierwsze na jednej z maszyn.

Niech N oznacza globalną ilość zasobów nieodnawialnych, a przez u_k oznaczmy tą część zasobów, które zostaną przydzielone k-tej maszynie 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 k-tej maszynie określony jest przez następującą funkcję $T_i(u_k, k)$:

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

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

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

3. Model formalny problemu

Jeżeli oznaczymy przez $Z_k \subset Z$ zbiór zadań uszeregowanych na k-tej maszynie, to T_{zak} znajdziemy rozwiązując następujący problem minimalizacyjny:

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

Ograniczenia nałożone na rozwiązanie tego problemu są następujące:

- $Z_r \cap Z_s = \phi; \quad r,\ s = 1,2,\ldots,m, \quad r \neq s, \ \bigcup_{k=1}^{m} Z_k = Z,$
- $\sum_{k=1}^{m} u_k \le N,$
- $u_1, u_2, \ldots, u_m$ - 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.
Czas T_{zak} znajdziemy rozwiązując następujący problem minimalizacji dyskretno-ciągłej:

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

przy następujących ograniczeniach:

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

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

gdzie: $T_i' : [0, N] \times \{1,2,\ldots,m\} \to R^+$ jest rozszerzeniem następującej funkcji $T_i : \{1,2,\ldots,N\} \times \{1,2,\ldots,m\} \to R^+$ 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 $u_k^*, Z_k^*, k = 1,2,..., m$ są rozwiązaniami zadania (3), to:

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

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

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

Warunek (i) w **LEMACIE 1** oznacza, że w przydziale czasowo-optymalnym zasobów i zadań do maszyn wykorzystuje się wszystkie jednostki zasobów, a warunek (ii), że czasy pracy tych maszyn, na których wykonywane są jakieś zadania, są identyczne.

Zdefiniujmy funkcję $F(Z_1, Z_2, ..., Z_m)$ określoną dla m zbiorów $Z_1, Z_2, ..., Z_m$, dla których zachodzi ograniczenie (i) dla wzoru (3). Wartość tej funkcji jest rozwiązaniem następującego układu równań:

$$\begin{cases} \sum_{i \in Z_k} a_{ik} + \dfrac{\sum_{i \in Z_k} b_{ik}}{u_k} = F(Z_1, Z_2, \dots, Z_m); \quad k : Z_k \neq \varnothing, \quad k = 1, 2, \dots, m \\ \sum_{k=1}^{m} u_k = N; \quad u_k > 0, \quad k : Z_k \neq \varnothing, \quad k = 1, 2, \dots, m. \end{cases} \tag{5}$$

Wykorzystując **LEMAT 1** oraz (5) zadanie minimalizacji (3) można przedstawić w następującej postaci:

$$T_{zak} = \min_{Z_1, Z_2, ..., Z_m} F(Z_1, Z_2, ..., Z_m), \tag{6}$$

przy następujących ograniczeniach:

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

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

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

$$u_k^* = \begin{cases} \dfrac{\sum_{i \in Z_k^*} b_{ik}}{F\left(Z_1^*, Z_2^*, ..., Z_m^*\right) - \sum_{i \in Z_k^*} a_{ik}}; & k : Z_k^* \neq \phi, \quad 1 \leq k \leq m, \\ 0 \qquad\qquad\qquad\qquad\quad ; & k : Z_k^* = \phi, \quad 1 \leq k \leq m \end{cases} \tag{7}$$

jest rozwiązaniem zadania (3).

4. Algorytm heurystyczny

Maszyny wchodzące w skład systemu maszyn równoległych różnią się pod względem szybkości wykonywanych zadań. Na szybkość tą wpływ ma ilość zasobów przydzielonych poszczególnym maszynom. Im więcej zasobów zostanie przydzielonych *k*-tej maszynie, tym będzie ona szybsza.

Zasoby przydzielone zostają do maszyn w następujący sposób:

- miarą szybkości realizacji *i*-tego zadania przez *k*-tą maszynę jest tzw. współczynnik podziału zasobów □; □ > 1,
- zakładamy, że maszyną najszybszą jest maszyna pierwsza, a maszyną najwolniejszą jest maszyna *m*-ta,
- maszynie *m*-tej przydzielamy u_m zasobów wg następującej zależności:

$$u_m = \frac{N}{1 + \sum_{k=1}^{m-1} [(m-k) \cdot \beta]} \tag{8}$$

- pozostałym maszynom przydzielamy zasoby wg następującej zależności:

$$u_k = (m-k) \cdot \beta \cdot u_m; \qquad k = 1, 2, \ldots, m-1. \tag{9}$$

Przedstawiony powyżej sposób przydziału zasobów do maszyn wykorzystany zostanie w zaproponowanym heurystycznym algorytmie szeregowania zadań na równoległych maszynach. Algorytm ten skonstruowany został w taki sposób, że najpierw szereguje on zadania na jednakowych maszynach, tj. takich, do których przydzielona została jednakowa liczba dostępnych zasobów, czyli $u_k = \frac{N}{m}$, k = 1, 2, …, m. Po tym uszeregowaniu następuje zróżnicowanie maszyn pod względem liczby przydzielanych im zasobów i sprawdzenie czy skrócony został czas zakończenia wykonywania wszystkich zadań T_{zak}.

Kolejne kroki algorytmu heurystycznego są następujące:

Krok 1. Oblicz czasy wykonywania zadań na poszczególnych maszynach

$$T_i(u_k, k) = a_{ik} + \frac{b_{ik}}{u_k}, \quad i = 1, 2, \ldots, n, \; k = 1, 2, \ldots, m$$ dla zadanej wartości $u_k = \frac{N}{m}$ i losowo generowanych parametrów a_{ik}, b_{ik}..

Krok 2. Uszereguj malejąco czasy wykonywania poszczególnych zadań i utwórz listę L tych zadań.

Krok 3. Oblicz średni czas $T_{śr}$ wykonywania zadań przez każdą z maszyn wg wzoru:

$$T_{śr} = \frac{\sum_{i=1}^{n} T_i(u_k, k)}{m}; \qquad i \in Z, k \in M, u_k = \frac{N}{m}.$$

Krok 4. Przydzielaj kolejno najdłuższe i najkrótsze zadania z listy L do pierwszej wolnej maszyny aż do momentu, gdy suma czasów wykonywania zadań przydzielonych tej maszynie nie przekroczy czasu $T_{śr}$. Przydzielone zadania usuń z listy L.

Krok 5. Jeżeli są jeszcze maszyny na których nie uszeregowano żadnych zadań to wróć do **Kroku 4**. W przeciwnym wypadku przejdź do **Kroku 6**.

Krok 6. Przydzielaj kolejno najkrótsze zadania z listy L do kolejnych maszyn od pierwszej poczynając aż do momentu, gdy suma czasów realizacji zadań przez kolejne maszyny nie przekroczy czasu $T_{śr}$. Przydzielone zadania usuń z listy L.

Krok 7. Jeżeli lista L nie została jeszcze wyczerpana to wróć do **Kroku 6**. W przeciwnym wypadku przejdź do **Kroku 8.**

Krok 8. Oblicz czas zakończenia wykonywania wszystkich zadań T_{zak} dla uszeregowania zadań na maszynach utworzonego w **Krokach 4÷7** i dla $u_k = \frac{N}{m}$.

Krok 9. Dla zadanego współczynnika □ przydziel zasoby u_k, $k = 1, 2, \ldots, m$ poszczególnym maszynom wyliczone z zależności (8) i (9).

Krok 10. Dla uszeregowania zadań na maszynach utworzonego w **Krokach 4÷7** i dla licz-by zasobów u_k przydzielonych maszynom w **Kroku 9** oblicz czas zakończenia wykonywania wszystkich zadań T_{zak}.

Krok 11. Powtórz **Krok 9** i **Krok 10** dla następnych dziewięciu zwiększających się kolej-no wartości współczynnika □. Po zakończeniu tych prób przejdź do **Kroku 12**.

Krok 12. Porównaj wartości czasów zakończenia wykonywania zadań T_{zak} z kolejnych prób i wybierz najkrótszy z tych czasów.

Krok 13. Wyznacz dyskretne ilości zasobów $\hat{u}_k, k = 1, 2, \ldots, m$ według zależności:

$$\hat{u}_{\alpha(k)} = \begin{cases} \lfloor u_{\alpha(k)} \rfloor + 1; & k = 1, 2, \ldots, \Delta, \\ \lfloor u_{\alpha(k)} \rfloor \quad ; & k = \Delta + 1, \Delta + 2, \ldots, m, \end{cases}$$

gdzie $\Delta = N - \sum_{j=1}^{m} \lfloor u_j \rfloor$ oraz □ jest permutacją elementów zbioru $M=\{1,2,..., m\}$ taką, że $u_{\alpha(1)} - \lfloor u_{\alpha(1)} \rfloor \geq u_{\alpha(2)} - \lfloor u_{\alpha(2)} \rfloor \geq \ldots \geq u_{\alpha(m)} - \lfloor u_{\alpha(m)} \rfloor$.

Jeżeli istnieją takie maszyny, którym przydzielono zerowe ilości zasobów, to przydziel każdej z tych maszyn po jednej jednostce zasobu pobierając je z kolejnych maszyn poczynając od maszyny, której przydzielono największą ilość zasobów.

5. Wyniki badań numerycznych

Przeprowadzono badania numeryczne na bazie przedstawionego algorytmu dla dziesięciu zwiększających się kolejno wartości współczynnika podziału zasobów □ z przedziału [2, 4, ... , 20]. Parametry charakteryzujące *i*-te zadanie i *k*-tą maszynę a_{ik} , b_{ik} wylosowane zostały ze zbioru {4.0, 8.0, ... , 80.0} przez generator o jednostajnym rozkładzie prawdopo-dobieństwa. Dla każdej kombinacji *n* i *m* wygenerowano 25 instancji. Rezultaty analizy porównawczej algorytmu heurystycznego skonstruowanego dla potrzeb niniejszej pracy i znanego z literatury algorytm *LPT* przedstawione zostały w Tab.1.

Tab. 1.Wyniki analizy porównawczej algorytmu heurystycznego i algorytmu LPT

n/m	Liczba instancji, dla których:			Δ^H	S^H	S^{LPT}
	$T_{zak}^H < T_{zak}^{LPT}$	$T_{zak}^H = T_{zak}^{LPT}$	$T_{zak}^H > T_{zak}^{LPT}$	%	sek	sek
30/3	13	1	11	2,5	2,4	1,9
30/6	13	2	10	2,7	2,7	2,4
30/9	12	2	11	3,4	3,9	3,4
30/12	13	3	9	3,6	4,6	4,1
30/15	14	1	10	3,9	5,8	5,5
60/3	13	1	11	1,9	2,7	2,3
60/6	14	0	11	2,7	3,8	3,2
60/9	13	0	12	3,6	4,9	3,9
60/12	14	2	9	4,8	6,5	5,2
60/15	15	2	8	5,1	7,8	5,9
90/3	12	2	11	2,8	3,9	3,2
90/6	13	1	11	2,9	6,3	5,9
90/9	14	1	10	3,7	7,3	6,2
90/12	15	0	10	4,8	8,6	7,4
90/15	13	1	11	5,2	9,7	8,6
120/3	12	1	12	2,9	5,9	5,2
120/6	13	0	12	3,0	6,7	6,2
120/9	14	1	10	3,5	8,7	7,8
120/12	15	2	8	4,9	9,9	8,6
120/15	16	1	8	6,8	12,1	10,4

W Tab. 1 występują następujące wielkości:

n – liczba zadań,

m – liczba maszyn,

T_{zak}^{H} – czas zakończenia wykonywania wszystkich zadań ze zbioru Z przy wykorzystaniu algorytmu heurystycznego,

T_{zak}^{LPT} – czas zakończenia wykonywania wszystkich zadań ze zbioru Z przy wykorzystaniu algorytmu *LPT*,

Δ^{H} – średnia procentowa poprawa czasu T_{zak}^{H} w stosunku do T_{zak}^{LPT}:

$$\Delta^{H} = \frac{T_{zak}^{LPT} - T_{zak}^{H}}{T_{zak}^{H}} \cdot 100\%,$$

S^{H} – średni czas obliczeń dla algorytmu heurystycznego,

S^{LPT} – średni czas obliczeń dla algorytmu *LPT*.

6. Uwagi końcowe

Przedstawione w poprzednim rozdziale eksperymenty obliczeniowe wykazały, że ja-kość szeregowania zadań na równoległych maszynach na bazie zaproponowanego w pracy algorytmu heurystycznego uległa poprawie w stosunku do szeregowania za pomocą znane-go z literatury algorytmu *LPT*. Kilkuprocentowa poprawa czasu T_{zak}^{H} w stosunku do T_{zak}^{LPT} może być zachętą do dalszych prac nad efektywnymi algorytmami heurystycznymi.

Zastosowanie podanego w pracy algorytmu heurystycznego jest wskazane przede wszystkim dla systemów produkcyjnych o dużej liczbie zadań, gdyż wówczas średnia procentowa poprawa □ □ □jest największa. Zaproponowany algorytm może służyć zarówno do rozdziału operacji na stanowiska produkcyjne wyposażone w odpowiednie maszyny w dyskretnych systemach produkcyjnych, jak i do szeregowania programów w wieloprocesorowych systemach komputerowych.

Literatura

1. Błażewicz J., Dell'Olmo P., Drozdowski M., Speranza M. G. : Scheduling multiprocessor tasks on three dedicated processors. Information Processing Letters 41, 1992, pp. 275-280.
2. Janiak A.: Single machine scheduling problem with a common deadline and resource dependent release dates. European Journal of Operational Research, Vol. 53, 1991, pp. 317-325.
3. Janiak A., Kovalyov M.: Single machine scheduling subject to deadlines and resources dependent processing times. European Journal of Operational Research, 1996, Vol. 94, pp. 284-291.
4. Nowicki E., Smutnicki C.: The flow shop with parallel machines. A Tabu search approach. European Journal of Operational Research 106, 1998, pp. 226-253.
5. Buchalski Z.: A Program Scheduling Heuristic Algorithm in Multiprocessing Computer System with Limited Memory Pages. Polish Journal of Environmental Studies, Vol. 15, No. 4C, 2006, pp. 26-29.

6. Józefowska J., Węglarz J.: On a methodology for discrete-continuous scheduling. European Journal of Operational Research, Vol. 107, 1998, pp. 338-353.
7. Józefowska J., Mika M., Różycki R., Waligóra G., Węglarz J.: Rozwiązywanie dyskretno-ciągłych problemów rozdziału zasobów przez dyskretyzację zasobu ciągłego. Zeszyty Naukowe Politechniki Śląskiej Nr 1474, seria Automatyka, Gliwice, 2000, z. 129, s. 221-229.
8. Kubale M., Giaro K.: Złożoność zwartego szeregowania zadań jednostkowych w systemie otwartym, przepływowym i mieszanym. Uczelniane Wydawnictwo Naukowo-Dydaktyczne AGH, seria-Automatyka, półrocznik, tom 5, zeszyt ½, Kraków, 2001, s. 329-334.
9. Boctor F. F.: A new and efficient heuristic for scheduling projects with resources restrictions and multiple execution models. European Journal of Operational Research, Vol. 90, 1996, pp. 349-361.
10. Buchalski Z.: An heuristic solution procedure to minimize the total processing time of programs in multiprocessing computer system. Information Systems Architecture and Technology ISAT 2005, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2005, pp. 20-26.
11. Buchalski Z.: An Heuristic Algorithm for Solving the Scheduling Problem in Multiprocessing Computer System. Polish Journal of Environmental Studies, Vol. 16, No. 4A, 2007, pp. 44-48.

Dr inż. Zbigniew BUCHALSKI
Instytut Informatyki, Automatyki i Robotyki
Politechnika Wrocławska
50-372 Wrocław, Janiszewskiego 11/17
tel.: (0 71) 320 32 92
e-mail: zbigniew.buchalski@pwr.wroc.pl