

Zbigniew Buchalski\*

## **Heurystyczny algorytm planowania rozdziału zadań dla systemu z maszynami równoległymi o ograniczonych zasobach**

### **1. Wprowadzenie**

Intensywny rozwój równoległych systemów przetwarzania informacji pociągnął za sobą wzrost zainteresowania problematyką szeregowania zadań w systemach maszyn pracujących równoległe (niezależnie). Szczególnego znaczenia nabiera problem minimalizacji długości uszeregowania zadań na maszynach [1–5]. Prowadzone od wielu lat badania problematyki czasowo-optimalnego szeregowania zadań i rozdziału zasobów dowiodły, że 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*-trudnych [6–11].

Problem szeregowania zadań na maszynach równoległych z równoczesnym rozdziałem zasobów bardzo często spotykany jest w różnego rodzaju złożonych procesach produkcyjnych. Czasy wykonywania zadań mogą być z góry zadane lub też mogą zmieniać się w zależności od współdzielonych zasobów. Zadania mogą być pewnymi procesami dynamicznymi opisywanymi za pomocą skomplikowanych modeli matematycznych.

Z formalnego punktu widzenia omawiane problemy można w sposób ogólny scharakteryzować następująco: dany jest zbiór zadań, zbiór maszyn oraz ograniczone zasoby przeznaczony do wykonywania tych zadań. Należy znaleźć takie dopuszczalne uszeregowanie tych zadań na maszynach oraz taki rozdział ograniczonych zasobów pomiędzy maszyny, na których te zadania są wykonywane, aby przyjęte kryterium efektywności osiągnęło optimum przy zadanych ograniczeniach.

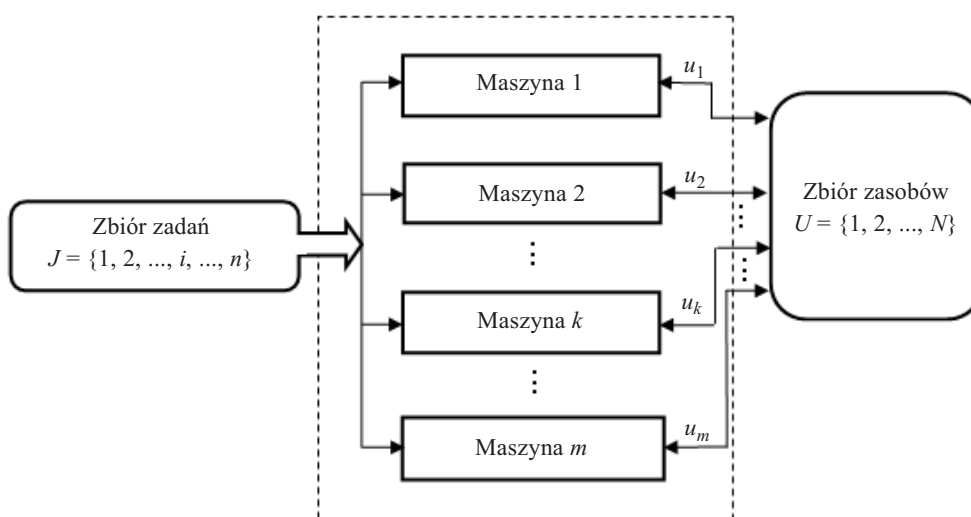
W niniejszym artykule przedstawiono pewien algorytm heurystyczny wyznaczający czasowo-optimalne uszeregowanie  $n$  zadań niezależnych niepodzielnych i zasobów nieodnawialnych podzielnych w sposób ciągły do  $m$  maszyn pracujących równoległe. Przedstawiono wyniki badań komputerowych przeprowadzonych na tym algorytmie dla losowo generowanych danych.

---

\* Instytut Informatyki, Automatyki i Robotyki, Politechnika Wroclawska

## 2. Opis problemu

Rozpatrzmy system maszyn równoległych przedstawiony na rysunku 1. Dany jest zbiór  $n$  zadań niezależnych ponumerowanych liczbami  $1, 2, \dots, n$  i oznaczony przez  $J$ . Zadania te należy wykonać bez przerw przy użyciu  $m$  maszyn równoległych ze zbioru  $M = \{1, 2, \dots, m\}$ . Każda maszyna, w dowolnej chwili czasu, może wykonywać co najwyżej jedno zadanie oraz wykonywanie zadania nie może być przerywane. Realizacja każdego z zadań na maszynach musi następować niezwłocznie po zakończeniu wykonywania poprzedniego zadania (lub nastąpić w chwili  $t = 0$ , gdy zadanie realizowane jest jako pierwsze na jednej z maszyn). Każde zadanie może być wykonywane na dowolnej maszynie, a liczba zadań do wykonania jest większa od liczby maszyn ( $n > m$ ).



Rys. 1. System maszyn równoległych

Zakładamy, że dysponujemy  $N$  jednakowymi jednostkami zasobu nieodnawialnego;  $N \geq m$ . Oznaczmy przez  $u_k$  tę ilość zasobu, która zostanie przydzielona  $k$ -tej maszynie. Zbiór zasobów oznaczmy przez  $U = \{1, 2, \dots, N\}$ .

Zachodzą następujące ograniczenia zasobowe:

$$\sum_{k=1}^m u_k \leq N, \quad k \in M, \quad u_k \in U, \quad u_k \geq 0.$$

Czas wykonywania  $i$ -tego zadania na  $k$ -tej maszynie, jeżeli przydzielono jej  $u_k$  jednostek zasobu, określony jest następującą funkcją:

$$T_i(u_k, k) = a_{ik} + \frac{b_{ik}}{u_k}, \quad u_k \in U, \quad k \in M, \quad i \in J \quad (1)$$

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

Postawiony do rozwiązania problem polega na znalezieniu takiego uszeregowania wszystkich  $n$  zadań na  $m$  maszynach i takiego przydziału zasobów do maszyn, aby minimalizować czas  $T_{opt}$  zakończenia wykonywania całego zbioru zadań  $J$ .

### 3. Model matematyczny problemu

Oznaczmy przez  $J_1, J_2, \dots, J_k, \dots, J_m$  podzbiory zadań realizowanych kolejno na 1, 2, ...,  $k$ , ...,  $m$ -tej maszynie. Należy rozwiązać następujący problem minimalizacji czasu zakończenia wykonywania zbioru zadań  $T_{opt}$

$$T_{opt} = \min_{\substack{J_1, J_2, \dots, J_m \\ u_1, u_2, \dots, u_m}} \max_{1 \leq k \leq m} \left\{ \sum_{i \in J_k} T_i(u_k, k) \right\} \quad (2)$$

przy następujących ograniczeniach:

- (i)  $J_w \cap J_z = \emptyset$ ;  $w, z = 1, 2, \dots, m$ ,  $w \neq z$ ,  $\bigcup_{k=1}^m J_k = J$ ;
- (ii)  $\sum_{k=1}^m u_k \leq N$ ;  $u_k \in U$ ,  $k = 1, 2, \dots, m$ ;
- (iii)  $u_1, u_2, \dots, u_m$  – całkowite dodatnie.

Postawione zadanie należy do klasy problemów  $NP$ -trudnych. Ograniczenie (iii) powoduje, że problem, który należy rozwiązać, jest dość skomplikowany. Aby go rozwiązać, dokonamy relaksacji warunku  $u_k \in U$ , zastępując go warunkiem  $0 \leq u_k \leq N$ ,  $k = 1, 2, \dots, m$ . Po rozwiązaniu odpowiedniego zadania minimalizacji dyskretno-ciągłej zaokrągla się otrzymane wartości zasobów  $u_k$  do liczb całkowitych dodatnich (patrz **krok 12** algorytmu heurystycznego).

Przy uwzględnieniu powyższej relaksacji, minimalny czas  $T_{opt}$  zakończenia wykonania zbioru zadań  $J$  przez system maszyn równoległych znajdziemy, rozwiązując następujące zadanie minimalizacji dyskretno-ciągłej

$$T_{opt} = \min_{\substack{J_1, J_2, \dots, J_m \\ u_1, u_2, \dots, u_m}} \max_{1 \leq k \leq m} \left\{ \sum_{i \in J_k} \tilde{T}_i(u_k, k) \right\} \quad (3)$$

przy następujących ograniczeniach:

- (i)  $J_w \cap J_z = \emptyset$ ;  $w, z = 1, 2, \dots, m$ ,  $w \neq z$ ,  $\bigcup_{k=1}^m J_k = J$ ;
- (ii)  $\sum_{k=1}^m u_k \leq N$ ;  $u_k \geq 0$ ,  $k = 1, 2, \dots, m$ ,

gdzie  $\tilde{T}_i : [0, N] \times \{1, 2, \dots, m\} \rightarrow R^+$  jest rozszerzeniem funkcji  $T_i : \{1, 2, \dots, N\} \times \{1, 2, \dots, m\} \rightarrow R^+$  i określone jest przez funkcję:

$$\tilde{T}_i(u_k, k) = a_{ik} + \frac{b_{ik}}{u_k}, \quad u_k \in [0, N], \quad a_{ik} > 0, \quad b_{ik} > 0, \quad k \in M, \quad i \in J \quad (4)$$

Rozwiązania optymalne zadania (3) oznaczmy przez  $u_k^*, J_k^*$ ,  $k = 1, 2, \dots, m$ . W celu znalezienia tych rozwiązań konieczne będzie wykorzystanie poniższego **Lematu**.

### Lemat

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

- (i)  $\sum_{k=1}^m u_k^* = N$ ;  $u_k^* > 0$ ,  $k : J_k^* \neq \emptyset$ ,  $k = 1, 2, \dots, m$ ;  
 $u_k^* = 0$ ,  $k : J_k^* = \emptyset$ ,  $k = 1, 2, \dots, m$ ;
- (ii)  $\sum_{i \in J_k^*} \tilde{T}_i(u_k^*, k) = \text{const}$ ;  $k : J_k^* \neq \emptyset$ ,  $k = 1, 2, \dots, m$ .

Warunek (i) w **Lemacie** 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  $F(J_1, J_2, \dots, J_m)$  określoną dla  $m$  zbiorów  $J_1, J_2, \dots, J_m$ , dla których zachodzi ograniczenie (i) zadania (3). Wartość tej funkcji jest rozwiązaniem następującego układu równań:

$$\begin{cases} \sum_{i \in J_k} a_{ik} + \frac{\sum_{i \in J_k} b_{ik}}{u_k} = F(J_1, J_2, \dots, J_m); & k : J_k \neq \emptyset, k = 1, 2, \dots, m \\ \sum_{k: J_k \neq \emptyset} u_k = N; \quad u_k > 0; \quad k : J_k \neq \emptyset, k = 1, 2, \dots, m. \end{cases} \quad (5)$$

Wykorzystując **Lemat** oraz (5), można zadanie minimalizacji (3) przedstawić w postaci:

$$T_{opt} = \min_{J_1, J_2, \dots, J_m} F(J_1, J_2, \dots, J_m) \quad (6)$$

przy następujących ograniczeniach:

- (i)  $J_w \cap J_z = \emptyset$ ;  $w, z = 1, 2, \dots, m$ ,  $w \neq z$ ;
- (ii)  $\bigcup_{k=1}^m J_k = J$ ;  $k = 1, 2, \dots, m$ .

Jeżeli  $J_1^*, J_2^*, \dots, J_m^*$  są rozwiązaniami zadania (6), to  $u_k^*, J_k^*$ ,  $k = 1, 2, \dots, m$ , gdzie:

$$u_k^* = \begin{cases} \frac{\sum_{i \in J_k^*} b_{ik}}{F(J_1^*, J_2^*, \dots, J_m^*) - \sum_{i \in J_k^*} a_{ik}}; & k : J_k^* \neq \emptyset, \quad 1 \leq k \leq m \\ 0 & k : J_k^* = \emptyset, \quad 1 \leq k \leq m \end{cases} \quad (7)$$

są rozwiązaniami zadania (3).

#### 4. Algorytm heurystyczny i wyniki badań komputerowych

Zakładamy, że maszyną najszybszą jest maszyna pierwsza ze zbioru  $M$ , a maszyną najwolniejszą jest maszyna  $m$ -ta. Miarą szybkości realizacji  $i$ -tego zadania przez  $k$ -tą maszynę jest tzw. współczynnik podziału zasobów  $\gamma$ ;  $\gamma > 1$ . Ostatniej  $m$ -tej maszynie przydzielamy  $u_m$  zasobów wg następującej zależności:

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

Pozostałym maszynom przydzielamy zasoby wg następującej zależności:

$$u_k = (m-k) \cdot \gamma \cdot u_m; \quad k = 1, 2, \dots, m-1 \quad (9)$$

Kolejne kroki algorytmu heurystycznego są następujące:

##### **Krok 1.**

Oblicz czasy wykonywania zadań  $T_i(u_k, k)$  na poszczególnych maszynach wg wzoru (1) dla zadanej wartości  $u_k = \frac{N}{m}$  i losowo generowanych parametrów  $a_{ik}, b_{ik}$ .

##### **Krok 2.**

Uzereguj malejąco czasy wykonywania poszczególnych zadań i utwórz listę  $L$  tych zadań.

##### **Krok 3.**

Uzereguj pierwszych  $m$  najdłuższych zadań z listy  $L$  do kolejnych  $m$  maszyn, poczynając od pierwszej maszyny, a na  $m$ -tej kończąc, a następnie usuń te zadania z listy  $L$ .

##### **Krok 4.**

Przydziel najkrótsze i najdłuższe zadanie z listy  $L$  do kolejnych maszyn, poczynając od pierwszej maszyny, a na  $m$ -tej kończąc, a następnie usuń te zadania z listy  $L$ . Jeżeli lista  $L$  się nie wyczerpała, to wykonaj następny krok, w przeciwnym wypadku przejdź do **kroku 6**.

##### **Krok 5.**

Uzereguj kolejne zadania z listy  $L$  na maszynach, poczynając od  $m$ -tej maszyny, a na pierwszej kończąc, a następnie usuń te zadania z listy  $L$ . Jeżeli lista  $L$  się nie wyczerpała, to wróć do **kroku 4**, w przeciwnym wypadku wykonaj **krok 6**.

**Krok 6.**

Oblicz sumaryczne czasy wykonywania zadań na poszczególnych maszynach dla uszeregowania zadań  $J_1, J_2, \dots, J_m$  utworzonego w **krokach 3÷5** i dla zadanej liczby zasobów  $u_k = \frac{N}{m}$  i podaj czas zakończenia  $T_{opt}$  wykonywania wszystkich zadań będący najdłuższym z tych czasów.

**Krok 7.**

Usuń najkrótsze zadanie z maszyny o najdłuższym czasie wykonywania zadań i przydziel je do maszyny o najkrótszym czasie wykonywania zadań.

**Krok 8.**

Oblicz czas zakończenia wykonywania wszystkich zadań  $T_{opt}$  po zamianie zadań w **kroku 7**. Jeżeli czas ten ulegnie skróceniu, to wróć do **kroku 7**. W przeciwnym przypadku cofnij ostatnio wykonaną czynność w **kroku 7** i zakończ szeregowanie zadań na maszynach.

**Krok 9.**

Dla danego współczynnika  $\gamma$  przydziel zasoby  $u_k, k \in M$  poszczególnym maszynom wyliczone z zależności (8) i (9).

**Krok 10.**

Dla uszeregowania zadań utworzonego w **krokach 3÷8** i dla liczby zasobów  $u_k, k \in M$  przydzielonych maszynom w **kroku 9**, oblicz czas zakończenia wykonywania wszystkich zadań  $T_{opt}$ .

**Krok 11.**

Powtórz **krok 9** i **krok 10** dla następnych dziewięciu zwiększających się kolejno wartości współczynnika  $\gamma$ . Porównaj wartości czasów zakończenia wykonywania zadań  $T_{opt}$  z kolejnych prób i wybierz najkrótszy z tych czasów.

**Krok 12.**

Wyznacz dyskretne liczby zasobów  $\hat{u}_k, k \in M$  według zależności:

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

gdzie  $\Delta = N - \sum_{j=1}^m \lfloor u_j \rfloor$  oraz  $\alpha$  jest permutacją elementów zbioru  $M = \{1, 2, \dots, m\}$  taką, że

$$u_{\alpha(1)} - \lfloor u_{\alpha(1)} \rfloor \geq u_{\alpha(2)} - \lfloor u_{\alpha(2)} \rfloor \geq \dots \geq u_{\alpha(m)} - \lfloor u_{\alpha(m)} \rfloor.$$

Na bazie przedstawionego w pracy algorytmu heurystycznego przeprowadzono badania komputerowe. Algorytm heurystyczny poddano ocenie dla dziesięciu zwiększających się kolejno wartości współczynnika podziału zasobów  $\gamma$  ze zbioru  $\{2, 4, \dots, 20\}$ . Dla zadanej liczby zadań  $n = 40, 80, 120, 160$ , liczby maszyn  $m = 3, 6, 9, 12$  i liczby zasobów  $N = 10\,000$  wylosowano parametry  $a_{ik}, b_{ik}$  ze zbioru  $\{1, 0, 1, 5, \dots, 19, 5, 20, 0\}$  przez generator

o jednostajnym rozkładzie prawdopodobieństwa. Dla każdej kombinacji  $n$  i  $m$  wygenerowano 30 instancji. Rezultaty analizy porównawczej algorytmu heurystycznego przedstawionego w niniejszej pracy i znanego z literatury algorytmu  $LPT$  przedstawione zostały w tabeli 1.

**Tabela 1**  
Wyniki analizy porównawczej algorytmu heurystycznego i algorytmu  $LPT$

$n/m$	Liczba instancji, dla których:			$\Delta^H$	$S^H$	$S^{LPT}$
	$T_{opt}^H < T_{opt}^{LPT}$	$T_{opt}^H = T_{opt}^{LPT}$	$T_{opt}^H > T_{opt}^{LPT}$	%	s	s
40/3	16	0	14	1,7	1,6	1,3
80/3	15	2	13	2,2	2,8	2,1
120/3	16	0	14	2,8	4,1	3,4
160/3	16	1	13	3,1	5,4	4,1
40/6	14	3	13	1,9	1,9	1,6
80/6	16	1	13	2,4	3,2	2,8
120/6	16	2	12	2,9	5,1	4,5
160/6	17	2	11	3,5	8,1	6,8
40/9	15	1	14	1,6	2,1	1,8
80/9	16	0	14	2,6	3,8	3,2
120/9	17	0	13	3,1	6,1	4,9
160/9	16	3	11	3,9	9,2	7,8
40/12	15	2	13	1,8	2,4	2,1
80/12	16	1	13	2,8	4,1	3,3
120/12	17	1	12	3,3	7,2	6,1
160/12	17	2	11	4,1	10,8	8,2

W tabeli 1 występują następujące wielkości:

$n$  – liczba zadań,  $m$  – liczba maszyn;

$T_{opt}^H$  – czas zakończenia wykonywania wszystkich zadań ze zbioru  $J$  przy wykorzystaniu algorytmu heurystycznego;

$T_{opt}^{LPT}$  – czas zakończenia wykonywania wszystkich zadań ze zbioru  $J$  przy wykorzystaniu algorytmu  $LPT$ ;

$\Delta^H$  – średnia procentowa poprawa czasu  $T_{opt}^H$  w stosunku do czasu  $T_{opt}^{LPT}$  wyrażona

$$\text{następującym wzorem: } \Delta^H = \frac{T_{opt}^{LPT} - T_{opt}^H}{T_{opt}^H} \cdot 100\%;$$

$S^H$  – średni czas obliczeń dla algorytmu heurystycznego;

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

## 5. Podsumowanie

Przedstawione w poprzednim rozdziale wyniki badań komputerowych wykazały, że jakość szeregowania zadań na równoległych maszynach na bazie zaproponowanego w pracy algorytmu heurystycznego uległa poprawie w stosunku do szeregowania za pomocą znanego z literatury algorytmu *LPT*. Kilkuprocentowa poprawa czasu  $T_{opt}^H$  w stosunku do  $T_{opt}^{LPT}$  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  $\Delta^H$  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 [12–15].

## Literatura

- [1] Bachman A., Janiak A.: *Jednomaszynowy problem szeregowania zadań czasowo i zasobowo zależnych przy kryterium minimalizacji czasu zakończenia wykonywania zadań*. Zeszyty Naukowe Politechniki Śląskiej, nr 1474, seria – Automatyka, Gliwice, z. 129, 2000, 23–32
- [2] 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, 349–361
- [3] Buchalski Z.: *Application of heuristic algorithm for the tasks scheduling on parallel machines to minimise the total processing time*. Proc. of the 15th International Conference on Systems Science, Vol. II, Oficyna Wydawnicza Politechniki Wrocławskiej, 2004, 235–242
- [4] Buchalski Z.: *Minimising the Total Processing Time for the Tasks Scheduling on the Parallel Machines System*. Proc of the 12th IEEE International Conference on Methods and Models in Automation and Robotics, 2006, 1081–1084
- [5] Janiak A.: *Single machine scheduling problem with a common deadline and resource dependent release dates*. European Journal of Operational Research, vol. 53, 1991, 317–325
- [6] Błażewicz J., Ecker E., Schmidt G., Węglarz J.: *Scheduling in Computer and Manufacturing Systems*. Springer-Verlag, 1993
- [7] Ishii H., Martel C., Masuda T., Nishida T.: *A generalized uniform processor system*. Oper. Res. Vol. 33, 1985, 346–362
- [8] Janiak A., Kovalyov M.: *Single machine scheduling subject to deadlines and resources dependent processing times*. European Journal of Operational Research, , vol. 94, 1996, 284–291
- [9] Józefowska J., Węglarz J.: *On a methodology for discrete-continuous scheduling*. European Journal of Operational Research, vol. 107, 1998, 338–353
- [10] 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, z. 129, 2000, 221–229
- [11] Nowicki E., Smutnicki C.: *The flow shop with parallel machines. A Tabu search approach*. European Journal of Operational Research 106, 1998, 226–253
- [12] Bianco L., Błażewicz J., Dell’olmo P., Drozdowski M.: *Preemptive multiprocessors task scheduling with release times and times windows*. Annals of Operations Research 70, 1997, 43–55



- 
- [13] Błażewicz J., Dell'olmo P., Drozdowski M., Speranza M.G.: *Scheduling multiprocessor tasks on three dedicated processors*. Information Processing Letters 41, 1992, 275–280
  - [14] Błażewicz J., Drabowski M., Węglarz J.: *Scheduling multiprocessor tasks to minimize schedule length*. IEEE Transactions on Computers C-35, 1986, 389–393
  - [15] 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, 26–29

