ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2002
Seria: AUTOMATYKA z. 134 Nr kol. 1554

Stanisław GAWIEJNOWICZ, Wiesław KURC, Lidia PANKOWSKA*
Uniwersytet im. Adama Mickiewicza, Poznań

ALGORYTM ZACHŁANNY DLA PEWNEGO PROBLEMU SZEREGOWANIA ZADAŃ CZASOWO-ZALEŻNYCH

Streszczenie. W pracy jest rozważany jednomaszynowy problem szeregowania zadań czasowo-zależnych. Czas wykonywania p_j zadania j jest funkcją czasu t rozpoczęcia tego zadania, $p_j(t) = 1 + \alpha_j t$, gdzie $\alpha_j > 0$ dla j = 0, 1, ... ,n. Zadania są niepodzielne, niezależne, nie ma czasów gotowości ani linii krytycznych, a kryterium optymalności uszeregowania jest łączny czas zakończenia wykonywania zadań.
Dla danego ciągu współczynników α_j, j = 0, 1, ... ,n definiowana jest *sygnatura*, której własności stanowią podstawę do skonstruowania algorytmu zachłannego.

A GREEDY ALGORITHM FOR A TIME-DEPENDENT SCHEDULING PROBLEM

Summary. A single machine time-dependent scheduling problem is considered. The processing time p_j of job j is a function of the starting time t of the job, $p_j(t) = 1 + \alpha_j t$, where $\alpha_j > 0$ for j = 0, 1, ... ,n. Jobs are non-preemptable and independent, there are neither ready times nor deadlines, and the criterion of optimality is the total completion time.
On the ground of properties of a *signature* for a given sequence of coefficients α_j, j = 0,1, ... ,n a greedy algorithm for the problem is constructed.

1. Wprowadzenie

Klasyczna teoria szeregowania zadań opiera się na następujących założeniach [2],[5]:

(Z1) w danej chwili czasu każde zadanie może być wykonywane przez co najwyżej jeden procesor i każdy procesor wykonuje co najwyżej jedno zadanie,

(Z2) prędkość procesora w trakcie wykonywania zadania jest niezmienna,

(Z3) czasy wykonania zadań są wielkościami stałymi.

We współczesnej (nieklasycznej) teorii szeregowania bada się modele, w których jedno lub więcej z powyższych założeń nie jest spełnionych. Odrzucając założenie (Z3) mówimy o zadaniach ze zmiennymi czasami wykonania, w tym w szczególności o zadaniach czasowo-zależnych.

W pracy rozważany jest problem szeregowania zadań na jednej maszynie, przy założeniu że czas wykonania p_j zadania j jest niemalejącą funkcją liniową czasu rozpoczęcia zadania t, a przyjętym kryterium optymalizacji jest ΣC_j, czyli suma czasów zakończenia wykonania zadań. Stosując notację wprowadzoną przez Grahama i in. [9], tak opisany problem dalej oznaczy się symbolem:

$$1 \mid p_j(t) = 1 + \alpha_j t \mid \Sigma C_j \tag{1}$$

Problem ten może być modelem dla wielu zagadnień praktycznych, w których upływ czasu zmienia warunki wykonania zadania. Do zadań tego typu należą między innymi: zabezpieczanie techniczne i medyczne klęsk żywiołowych i katastrof, modelowanie procesów uczenia, modelowanie operacji finansowych np. związanych ze spłatą wielu kredytów oraz wiele podobnych.

Problem szeregowania zadań czasowo-zależnych ma już bogatą literaturę [1],[5]. Większość znanych wyników dotyczy przypadku jednego procesora oraz dwóch podstawowych kryteriów jakości uszeregowania, a mianowicie długości uszeregowania C_{max} i łącznego czasu zakończenia ΣC_j. Dla kryterium C_{max} w przypadku jednego procesora i czasów wykonania postaci $p_j(t) = \beta_j + \alpha_j t$ problem jest rozwiązywalny w czasie O(n log n), a optymalne uszeregowanie wyznaczone jest przez nierosnący porządek ułamków α_j/β_j [3],[8]. W przypadku wielu procesorów zarówno problem $P2 \mid p_j(t) = \alpha_j t \mid C_{max}$, jak i problem $P2 \mid p_j(t) = \beta_j + \alpha_j t \mid C_{max}$ są NP-trudne [12]. Dla kryterium ΣC_j rozwiązanie wielomianowe znane jest tylko dla przypadku jednego procesora oraz czasów wykonania zadania postaci $p_j(t) = \alpha_j t$ [15], natomiast przypadek wieloprocesorowy jest już NP-trudny [4],[12]. Problem minimalizacji ΣC_j dla przypadku jednego procesora z czasami wykonania $p_j(t) = \beta_j + \alpha_j t$ jest otwarty, nawet jeśli $\beta_j = 1$ [6], [7], [14]. Znane są dla tego problemu jedynie pewne warunki, które musi spełniać uszeregowanie optymalne (V-kształtność, własność symetrii, wybór pierwszego elementu), co pozwala obniżyć złożoność z $O(n^n)$ do $O(2^n)$.

Otwartość problemu $1 \mid p_j(t) = 1 + \alpha_j t \mid \Sigma C_j$ była powodem podjęcia w niniejszej pracy dalszych badań, których wynikiem jest algorytm zachłanny o złożoności wielomianowej

oparty na badaniu *sygnatur* dla ciągu $a = (a_1, a_2, \ldots, a_n)$, gdzie $a_j = 1 + \alpha_j$ dla $j = 1, 2, \ldots, n$, z $a_0 = 1+\alpha_0$ ustalonym.

2. Sygnatury

Rozważany w pracy problem szeregowania zadań ma następującą charakterystykę. Zadania wykonywane są na jednej maszynie, czasy wykonania zadania opisane są niemalejącą funkcją liniową $p_j(t) = 1 + \alpha_j t$, gdzie $\alpha_j > 0$ dla $j = 0, 1, \ldots, n$, a t jest czasem rozpoczęcia zadania. Szukamy takiego uszeregowania zadań, aby zminimalizować sumę czasów zakończenia $\sum_{j=0}^{n} C_j$, gdzie

$$C_0 = 1, \quad C_j = C_{j-1} + p_j(C_{j-1}) = 1 + (1 + \alpha_j)C_{j-1}, \qquad j = 1, 2, \ldots, n. \tag{2}$$

Wzór (2) można zapisać jako złożenie funkcji $f_j(t) = t + p_j(t) = 1 + a_j t$, gdzie $a_j = 1 + \alpha_j$ a $t = 0$. Mamy wówczas dla $j = 0, 1, \ldots, n$

$$C_j = f_j(f_{j-1}(\ldots (f_0(t)) \ldots)) \tag{3}$$

Zadania są w pełni scharakteryzowane przez ciąg współczynników $\hat{a} = (a_0, a_1, \ldots, a_n)$, a problem optymalności uszeregowania jest równoważny znalezieniu takiej permutacji ciągu $\hat{a}$, aby ΣC_j była minimalna. Ze wzoru (3) wynika, że jeżeli rozpoczynamy w czasie $t = 0$, to wartość współczynnika pierwszego zadania w rozważanym uporządkowaniu, $a_{[0]}$, nie ma wpływu na wartości współrzędnych wektora $C = [C_0, C_1, \ldots, C_n]$. Zatem dla dowolnego ciągu $\hat{a} = (a_0, a_1, \ldots, a_n)$ poszukiwana optymalna permutacja musi spełniać warunek $a_{[0]} = \max_{0 \le i \le n} \{a_i\}$ [14]. Niech $a = (a_1, \ldots, a_n)$ jest podciągiem ciągu $\hat{a} = (a_{[0]}, a_1, \ldots, a_n)$ z $a_{[0]}$ maksymalnym, a $\bar{a} = (a_n, a_{n-1}, \ldots, a_1)$ oznacza odwrotną permutację ciągu a. Stosując wzór (3) przy $t = 0$ otrzymujemy następujące wzory na składowe wektora $C(a) = [C_0, C_1, \ldots, C_n]$:

$$C_0 = 1, \quad C_1 = a_1 C_0 + 1, \ldots, \quad C_n = a_n C_{n-1} + 1,$$

przy czym oznaczenie $C(a)$ podkreśla zależność wartości składowych wektora C od porządku elementów ciągu a. Powyższy wzór rekurencyjny można zapisać w postaci macierzowej

$$\begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ -a_1 & 1 & \dots & 0 & 0 \\ 0 & -a_2 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -a_n & 1 \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ \dots \\ C_n \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \dots \\ 1 \end{bmatrix}, \tag{4}$$

lub krótko $A(a)C(a) = d(1)$, gdzie $A(a)$ jest M-macierzą [16] daną wzorem (4), $C(a) = [C_0, C_1, \dots, C_n]^T$, $d(1) = [1,1,\dots,1]^T \in \Re^{n+1}$. Macierz $A(a)$ można rozłożyć na iloczyn n macierzy elementarnego przekształcenia nieunitarnego [11] $A(a) = L_n(a)L_{n-1}(a)\dots L_1(a)$, gdzie

$$L_i(a) = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & -a_i & 1 & \dots & 0 \\ 0 & 0 & \dots & -a_i a_{i+1} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -a_i \dots a_n & 0 & \dots & 1 \end{bmatrix}, \; i = 1, 2, \dots, n.$$

Macierz odwrotna $L_i^{-1}(a)$ różni się od macierzy $L_i(a)$ tylko znakiem elementów leżących pod główną przekątną w i-tej kolumnie. Zatem [11]

$$A^{-1}(a) = L_1^{-1}(a)L_2^{-1}(a)\dots L_n^{-1}(a) = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ a_1 & 1 & 0 & \dots & 0 & 0 \\ a_1 a_2 & a_2 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 a_2 \dots a_n & a_2 \dots a_n & a_3 \dots a_n & \dots & a_n & 1 \end{bmatrix}$$

Współrzędne wektora $C(a) = A^{-1}(a)d(1)$ i wektora $C(\bar{a}) = A^{-1}(\bar{a})d(1)$, dla $\bar{a} = (a_n, a_{n-1}, \dots, a_1)$ wyrażają się odpowiednio wzorami:

$$C_j(a) = 1 + \sum_{i=1}^{j} a_i a_{i+1} \dots a_j, \quad C_j(\bar{a}) = 1 + \sum_{i=n-j+1}^{n} a_{n-j+1} a_{n-j+2} \dots a_i, \quad j = 0, 1, \dots, n. \tag{5}$$

Bezpośrednio z definicji $\|\cdot\|_1$ [10],[11] i wzoru (5) wynika następujący wniosek.

Wniosek 1. Kryterium optymalności uszeregowania ΣC_j jest równoważne kryterium $\|C(a)\|_1$.

Ponadto, ponieważ $A^{-1}(a)d(\beta) = A^{-1}(a)d(1)\beta$, gdzie $d(\beta) = [\beta, \beta, \dots, \beta]^T \in \Re^{n+1}$ oraz $\|\beta C(a)\|_1 = \beta\|C(a)\|_1$, można sformułować wniosek:

Wniosek 2. Problemy $1 \mid p_j(t) = 1 + \alpha_j t \mid \Sigma C_j$ i $1 \mid p_j(t) = \beta + \alpha_j t \mid \Sigma C_j$, gdzie $\beta > 0$, są równoważne.

Dwie nowe miary, $F(a) = \|C(a)\|_1 - (n+1)$ i $M(a) = \|C(a)\|_\infty$, dla wektora $C(a)$ zdefiniowano na podstawie norm l_p Höldera w przestrzeni $\Re^{n+1}$ [10]. Ponieważ $\|C(a)\|_1 = \sum_{j=0}^{n} C_j$, a $\|C(a)\|_\infty = \max_{0 \le j \le n} \{C_j\} \equiv C_{max}$, zatem miary te wyrażają się wzorami:

$$F(a) = \sum_{j=1}^{n} \sum_{i=1}^{j} a_i \ldots a_j \quad \text{i} \quad M(a) = 1 + \sum_{i=1}^{n} a_i \ldots a_n \tag{6}$$

Niech Π_n oznacza zbiór permutacji ciągu $(1, 2, \ldots, n)$. Jeżeli $\pi \in \Pi_n$ jest pewną, ustaloną permutacją, wówczas przez a_π oznaczamy odpowiednią permutację ciągu $a = (a_1, a_2, \ldots, a_n)$. $F(a)$-problemem będziemy nazywać problem minimalizacji funkcjonału $F(a)$ po wszystkich permutacjach $\pi \in \Pi_n$. Niech $a = a_{\pi^*} = (a_1, \ldots, a_n)$ będzie optymalnym rozwiązaniem $F(a)$-problemu, wówczas $\bar{a} = (a_n, a_{n-1}, \ldots, a_1)$ jest również rozwiązaniem optymalnym (własność symetrii) oraz a_{π^*} jest V-ciągiem [14], tzn. jest on nierosnący dla $i = 1, \ldots, m$ oraz niemalejący dla $i = m, \ldots, n$, gdzie $1 < m < n$.

Dla danego ciągu $a = (a_1, a_2, \ldots, a_n)$ zdefiniujemy dwie *sygnatury*:

$$S^-(a) = M(\bar{a}) - M(a) \quad \text{i} \quad S^+(a) = M(\bar{a}) + M(a) \tag{7}$$

Dany jest ciąg $a = (a_1, a_2, \ldots, a_n)$ oraz dwie wielkości α, β. Niech $a^L = (\alpha|a|\beta)$ i $a^R = (\beta|a|\alpha)$ oznaczają konkatenację ciągu a z elementami α, β we wskazanym porządku.

Lemat 1. Dla danego ciągu a oraz elementów α i β, niech $a^L = (\alpha|a|\beta)$ oraz $a^R = (\beta|a|\alpha)$. Wówczas prawdziwe są równości:

$$F(a^L) = F(\alpha|a|\beta) = F(a) + \alpha M(\bar{a}) + \beta M(a) + \alpha \cdot a \cdot \beta,$$

$$F(a^R) = F(\beta|a|\alpha) = F(a) + \beta M(\bar{a}) + \alpha M(a) + \alpha \cdot a \cdot \beta,$$

gdzie $\alpha \cdot a \cdot \beta = \alpha \cdot (\prod_{j=1}^{n} a_j) \cdot \beta$.

Dowód. Pełny dowód można znaleźć w pracy [6].

Jako wniosek z lematu 1 oraz zgodnie z definicjami sygnatur $S^-(a)$ i $S^+(a)$ można otrzymać wzory na sumę i różnicę wartości funkcjonału $F(\cdot)$ na ciągach a^L i a^R.

Lemat 2. Zgodnie z notacją z lematu 1, prawdziwe są następujące tożsamości:

$$F(\alpha|a|\beta) - F(\beta|a|\alpha) = (\alpha - \beta) S^-(a),$$

$$\tfrac{1}{2}(F(\alpha|a|\beta) + F(\beta|a|\alpha)) = F(a) + \tfrac{1}{2}(\alpha + \beta)S^{+}(a) + \alpha \cdot a \cdot \beta.$$

Na podstawie powyższego lematu można określić strategię dołączania nowych elementów α, β do danego ciągu a, tak aby wartość funkcjonału $F(\cdot)$ malała. Następujące twierdzenie jest głównym wynikiem w tym rozdziale.

Twierdzenie 1. Niech $\alpha > 1$ i $\beta > 1$ będą danymi wielkościami oraz niech a będzie danym ciągiem $F(a)$-problemu. Wówczas równoważne są następujące pary nierówności:

(a) $F(\alpha|a|\beta) \leq F(\beta|a|\alpha)$ i $(\alpha - \beta)S^{-}(a) \leq 0$,

(b) $F(\alpha|a|\beta) \geq F(\beta|a|\alpha)$ i $(\alpha - \beta)S^{-}(a) \geq 0$.

Z lematu 2 i twierdzenia 1 wynika, że zachowanie sygnatury $S^{-}(\cdot)$ może być podstawą strategii zachłannej rozwiązania $F(a)$-problemu.

Twierdzenie 2. Niech $a = (a_1, a_2, \ldots, a_n)$ będzie uporządkowanym niemalejąco ciągiem dla $F(a)$-problemu, $u = (a_1, a_2, \ldots, a_{k-1})$, $\alpha = a_k$, $\beta = a_{k+1}$, gdzie $1 < k < n$ oraz $\alpha \leq \beta$. Wówczas:

(a) $S^{-}(u) \geq 0 \Rightarrow F(\alpha|a|\beta) \leq F(\beta|a|\alpha)$,

(b) $S^{-}(u) \leq 0 \Rightarrow F(\alpha|a|\beta) \geq F(\beta|a|\alpha)$.

Dowód. Implikacja wynika z twierdzenia 1 oraz z faktu, że $\alpha - \beta \leq 0$.

3. Algorytm zachłanny

Niech $a = (a_1, a_2, \ldots, a_n)$ będzie uporządkowanym niemalejąco ciągiem dla $F(a)$-problemu, a u oznacza V-ciąg złożony z pierwszych $k \geq 1$ elementów ciągu a, dla którego aktualna wartość funkcjonału $F(\cdot)$ wynosi $F(u)$. Niech $\alpha = a_{k+1}$ i $\beta = a_{k+2}$, $\alpha \leq \beta$, będą dwoma kolejnymi elementami ciągu a. Przyjmując $u^L = (\alpha|u|\beta)$ i $u^R = (\beta|u|\alpha)$ otrzymujemy dwie możliwości rozszerzenia jednocześnie prawej i lewej gałęzi V-ciągu. Proponowana strategia zachłanna polega na takim wyborze rozszerzenia, aby wartość funkcjonału $F(\cdot)$ była najmniejsza. Niech ponadto $a_{[0]} = \max_{0 \leq i \leq n} \{a_i\}$.

Algorytm zachłanny dla problemu $1 \mid p_j(t) = 1 + \alpha_j t \mid \Sigma C_j$

Wejście: ciąg $\hat{a} = (a_0, a_1, \ldots, a_n)$, gdzie $a_j = 1 + \alpha_j$ dla $j = 0, 1, \ldots, n$

Wyjście: suboptymalny V-ciąg $u = (a_{[0]}, u_1, u_2, \ldots, u_n)$ otrzymany z ciągu $\hat{a}$

Krok 1: {przygotowawczy} sortowanie ciągu $\hat{a}$ w porządku niemalejącym, $a_{[1]} \leq \ldots \leq a_{[n]} \leq a_{[0]}$;

```
Krok 2: {pętla główna}
    If n nieparzyste then
     begin   u := (a[1]);
      for i := 2 to n-1 step 2 do
        if S⁻(u) ≤ 0 then u := (a[i+1]|u| a[i]) else u := (a[i]|u| a[i+1]);
     end
    else {n parzyste}
     begin   u := (a[1], a[2]);
      for i := 3 to n-1 step 2 do
        if S⁻(u) ≤ 0 then u := (a[i+1]|u| a[i]) else u := (a[i]|u| a[i+1]);
     end
Krok 3: {końcowy}   u := (a[0]|u).
```

Przykład 1. Niech a = (2, 3, 4, 6, 8, 16, 21). Optymalny V-ciąg ma postać a^* = (21, 8, 6, 3, 2, 4, 16) z wartością kryterium $\Sigma C_j(a^*)$ = 23226 ($F(a^*)$ = 23219). Proponowany w pracy algorytm generuje ciąg u = (21, 8, 6, 2, 3, 4, 16), dla którego $\Sigma C_j(u)$ = 23240. Inne znane heurystyki np. H1, H2 z pracy [14] dają odpowiednio V-ciągi: a_{H1} = (21, 8, 4, 2, 3, 6, 16) i a_{H2} = (21, 6, 3, 2, 4, 8, 16) z wartościami kryterium uporządkowania odpowiednio $\Sigma C_j(a_{H1})$ = 23418 i $\Sigma C_j(a_{H2})$ = 24890.

Przykład pokazuje, że chociaż w ogólnym przypadku algorytm nie daje rozwiązania optymalnego, to często zachowuje się lepiej niż inne znane heurystyki. Można ponadto przypuszczać, co potwierdzają eksperymenty, że w przypadku ciągów arytmetycznych algorytm generuje rozwiązanie optymalne.

Przykład 2. Niech a = (2, 3, 4, 5, 6, 7, 8). Algorytm generuje optymalny V-ciąg a^* = (8, 6, 5, 2, 3, 4, 7) z minimalną wartością kryterium $\Sigma C_j(a^*)$ = 27494. Rozwiązania otrzymane za pomocą heurystyk H1, H2 [14] mają postać a_{H1} = (8, 6, 4, 2, 3, 5, 7) i a_{H2} = (8, 5, 4, 2, 3, 6, 7) z wartościami kryterium uszeregowania odpowiednio: $\Sigma C_j(a_{H1})$ = 27519 i $\Sigma C_j(a_{H2})$ = 28131.

4. Przypadek współczynników naturalnych

Zaproponowany algorytm zachłanny znacznie się upraszcza, gdy elementy ciągu a są kolejnymi liczbami naturalnymi. Wynika to z faktu, że w pętli głównej algorytmu nie trzeba obliczać wartości sygnatury $S^-(u)$, ponieważ jej znak zmienia się cyklicznie. Koszt algorytmu wynosi zatem O(n log n).

Sygnaturę $S^-(a) = S_n^-$ dla n elementowego ciągu a zapiszemy teraz w postaci:

$$S_n^- = \sum_{i=1}^{m} a_1 \ldots a_i - \sum_{i=1}^{n-m} a_{n-i+1} \ldots a_n + \sum_{i=1}^{n-m} a_1 \ldots a_{m+i} - \sum_{i=1}^{n} a_i \ldots a_n, \quad 1 \le m \le n \qquad (8)$$

Z powyższego wzoru wyprowadzimy reprezentację sygnatury, oddzielnie dla parzystej i nieparzystej liczby elementów ciągu a.

Lemat 3. Niech $a = (a_1, \ldots, a_n)$. Jeżeli n = 2m, to

$$S_{2m}^- = \sum_{i=1}^{m} \eta_i(m)(a_1 \ldots a_{m-i+1} - a_{m+i} \ldots a_{2m}),$$

gdzie $\eta_1(m) = 1$, $\eta_i(m) = 1 + a_{m-i+2} \ldots a_{m+i-1}$ dla i = 2, 3, ..., m.

Jeśli n = 2m – 1, to

$$S_{2m-1}^- = \sum_{i=1}^{m-1} \omega_i(m)(a_1 \ldots a_{m-i} - a_{m+1} \ldots a_{2m-1}),$$

gdzie $\omega_i(m) = 1 + a_{m-i+1} \ldots a_{m+i-1}$ dla i = 1, 2, ..., m – 1.

Dowód. Pełny dowód można znaleźć w pracy [6].

Niech a= (1, 2, ..., n). Eksperyment obliczeniowy przeprowadzony dla n = 2, 3, ...,17 generuje optymalne rozwiązanie F(a)-problemu, odpowiednio dla n = 2m i n = 2m-1 postaci:

m = 1	1 2	3 1 2
m = 2	4 1 2 3	4 3 1 2 5
m = 3	5 4 1 2 3 6	7 4 3 1 2 5 6
	...	...

Takie samo rozwiązanie otrzymamy stosując algorytm zachłanny. Pokażemy, że nie wymaga to obliczania sygnatur, ponieważ ich znaki zmieniają się cyklicznie.

Niech ciąg a dla F(a)-problemu ma odpowiednio postać:

dla n = 2m, $\quad a = (r_m + (-1)^m, \ldots, r_2 + 1, r_1 - 1, r_1, r_2, \ldots, r_m)$,

dla n = 2m-1, $\quad a = (s_{m-1} + 2, \ldots, s_2 + 2, s_1 + 2, s_1, s_2, \ldots, s_m)$, $\qquad (9)$

gdzie

$$r_k = 2k - \tfrac{1}{2}((-1)^k + 3) + 1, \qquad k = 1, 2, \ldots, m \text{ dla } n = 2m,$$

$$s_k = 2k - \tfrac{1}{2}((-1)^k + 3), \qquad k = 1, 2, \ldots, m \text{ dla } n = 2m-1.$$

Stosując lemat 3 dla ciągu a danego wzorami (9) otrzymujemy:

Lemat 4. Niech n = 2m (przypadek parzysty). Wówczas dla m = 1, 2, ...

$$S^-_{2m} = \sum_{i=1}^{m} \eta_i(m)[(r_i + (-1)^i)\ldots(r_m + (-1)^m) - r_i \ldots r_m],$$

gdzie $\eta_1(m) = 1$, $\eta_i(m) = 1 + (r_1 \ldots r_{i-1})[(r_1 - 1) \ldots (r_{i-1} + (-1)^{i-1})]$ dla i = 2, ... , m.

Stosując reprezentację sygnatury podaną w lemacie 4 sformułujemy główne twierdzenie tej części pracy.

Twierdzenie 3. Niech n = 2m i niech a będzie ciągiem danym wzorami (9). Wówczas dla kolejnych sygnatur parzystych S^-_{2m+2} i S^-_{2m} prawdziwy jest następujący związek:

$$S^-_{2m+2} = r_{m+1} S^-_{2m} + (-1)^{m+1} \sum_{i=1}^{m+1} \eta_i(m)(r_i + (-1)^i)\ldots(r_m + (-1)^m),$$

gdzie sygnatura S^-_{2m} i współczynniki η_i są zdefiniowane jak w lemacie 4. Ponadto prawdziwa jest następująca tożsamość:

$$S^-_{2m+2} = R_m[(-1)^{m+1} + \Theta_m], \text{ gdzie } \Theta_m = \frac{S^-_{2m}}{R_m}(r_{m+1} + (-1)^{m+1}) \text{ i } R_m = \sum_{i=1}^{m+1} \eta_i(m) r_i \ldots r_m.$$

Analogiczne związki można sformułować dla przypadku nieparzystego.

Lemat 5. Niech n = 2m – 1 (przypadek nieparzysty). Wówczas dla m = 1, 2, ...

$$S^-_{2m-1} = \sum_{i=1}^{m-1} \omega_i(m)[(s_i + 2)\ldots(s_{m-1} + 2) - s_{i+1} \ldots s_m],$$

gdzie $\omega_i(m) = 1 + (s_1 \ldots s_i)[(s_1 + 2) \ldots (s_{i-1} + 2)]$ dla i = 1, 2, ... , m - 1.

Twierdzenie 4. Niech n = 2m + 1 i niech a będzie ciągiem danym wzorami (9). Wówczas dla kolejnych sygnatur nieparzystych S^-_{2m+1} i S^-_{2m-1} prawdziwy jest następujący związek:

$$S^-_{2m+1} = (s_m + 2) S^-_{2m-1} + (-1)^{m+1} \sum_{i=1}^{m} \omega_i(m)(s_{i+1} \ldots s_m),$$

gdzie sygnatura S^-_{2m-1} i współczynniki ω_i są zdefiniowane jak w lemacie 5. Ponadto prawdziwa jest następująca tożsamość:

$$S^-_{2m+1} = Q_m[(-1)^{m+1} + \Gamma_m], \text{ gdzie } \Gamma_m = \frac{S^-_{2m-1}}{Q_m}(s_m + 2) \text{ i } Q_m = \sum_{i=1}^{m} \omega_i(m) s_{i+1} \ldots s_m .$$

Zgodnie z twierdzeniem 2, w strategii zachłannej interesuje nas jedynie znak sygnatury. Przeprowadzony eksperyment obliczeniowy pokazał, że składnik Θ_m z twierdzenia 3 spełnia nierówność $|\Theta_m| \leq 0.8$ dla m = 1, 2, ..., 1000. Podobnie dla składnika Γ_m z twierdzenia 4 otrzymano oszacowanie $|\Gamma_m| < 0.73$ dla m = 1, 2, ..., 1000.

Hipoteza 1. Znaki sygnatur S^-_{2m} i S^-_{2m-1} zmieniają się zgodnie ze wzorem

$$\text{sign}(S^-_{2m}) = \text{sign}(S^-_{2m-1}) = (-1)^m, m \geq 1.$$

Hipoteza 2. Proponowany algorytm zachłanny generuje w przypadku ciągu $\alpha_j = j$, j = 1, ..., n rozwiązanie optymalne problemu $1 \mid p_j(t) = 1 + \alpha_j t \mid \Sigma C_j$.

5. Podsumowanie

Znalezienie takiego rozwiązania układu $A(a)C(a) = d(1)$, aby $\|C(a)\|_1$ była minimalna po wszystkich permutacjach ciągu $a = (a_1, a_2, \ldots, a_n)$, jest równoważne rozwiązaniu problemu $1 \mid p_j(t) = 1 + \alpha_j t \mid \Sigma C_j$. Na podstawie norm l_p Höldera zdefiniowano pojęcie *sygnatury* $S^-(a)$. Własności sygnatury odgrywają kluczową rolę w zachłannej strategii budowy ciągu suboptymalnego. Przeprowadzone eksperymenty obliczeniowe skłaniają do sformułowania hipotezy, że w przypadku ciągu $a = (a_1, a_2, \ldots, a_n)$, gdzie $a_i = i + 1$ dla i =1, 2, ...n, lub ogólniej ciągu arytmetycznego, algorytm generuje rozwiązanie optymalne w czasie O(n logn).

* Adres do korespondencji: Wydział Matematykl i Informatyki, Uniwersytet im. Adama Mickiewicza, ul. Umultowska 87, 61-614 Poznań, e-mail: lpankow@amu.edu.pl.

Praca została częściowo sfinansowana z grantu GN-05/2002 Wydziału Matematyki i Informatyki Uniwersytetu im. Adama Mickiewicza w Poznaniu.

LITERATURA

1. Alidaee B.,Womer N.K.: Scheduling with time dependent processing times: Review and extensions. Journal of the Operational Research Society, 50,1999, pp.711-720.

2. Błażewicz J., Ecker K.H., Pesch E., Schmidt G., Węglarz J.: Scheduling Computer and Manufacturing Processes, Springer, Berlin 2001.

3. Browne S., Yechiali U.: Scheduling deteriorating jobs on a single processor. Operations Research, 38, 1990, pp.495-498.

4. Chen Z-L.: Parallel machine scheduling with time dependent processing times. Discrete Applied Mathematics, 70, 1996, pp. 81-93.(Erratum Discr. Appl. Math., 75, 1997, p.103)

5. Gawiejnowicz S.: Szeregowanie zadań o zmiennych czasach wykonywania. Pro Dialog 13, Wydawnictwo Nakom, 2001, pp.75-90.

6. Gawiejnowicz S., Kurc W., Pankowska L.: A greedy approach for a time-dependent scheduling problem. Report 110/2001, Faculty of Mathematics and Computer Science, Adam Mickiewicz University, 2001.

7. Gawiejnowicz S., Kurc W., Pankowska L.: A greedy approach for a time-depended scheduling problem. Lecture Notes in Computer Science, accepted.

8. Gawiejnowicz S., Pankowska L.: Scheduling jobs with varying processing times. Information Processing Letters, 54, 1995, pp.175-178.

9. Graham R.L., Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G.: Optimisation and approximation in deterministic sequencing and scheduling theory. A survey, Annals of Discrete Mathematics 5, 1979, pp.287-326.

10. Hirsch F., Lacombe G.: Elements of Functional Analysis. Springer, Berlin 1999.

11. Dryja M., Jankowscy J. i M.: Przegląd metod i algorytmów numerycznych, WNT, Warszawa 1982.

12. Kononov A.: Scheduling problems with linear increasing processing times. In: Zimmermann U. et al. (eds), Operations Research 1996. Selected Papers of the Symposium on Operation Research. Springer, Berlin, 1997, pp.208-212.

13. Kononov A., Gawiejnowicz S.: NP-hard cases in scheduling deteriorating jobs on dedicated machines. Journal of the Operational Research Society, 52, pp.708-717.

14. Mosheiov G.: V-shaped policies for scheduling deteriorating jobs. Operations Research, 39, 1991, pp.979-991.

15. Mosheiov G.: Scheduling jobs under simple linear deterioration. Computers and Operations Research, 21, pp.653-659.

16. Varga R.S.: Matrix iterative analysis. Springer, Berlin 2000.

Recenzent: Dr hab. inż. Czesław Smutnicki

Abstract

In this paper we consider a single machine scheduling problem with time-dependent processing times of jobs and the total completion time criterion. The processing times are in the form of $p_j(t) = 1 + \alpha_j t$ where $\alpha_j > 0$, $j = 0, 1, \ldots, n$, and t is the starting time of a job. Jobs

are non-preemptable and independent, there are neither ready times nor deadlines. We wish to minimize the total completion time $\sum_{j=0}^{n} C_j$, where $C_0 = 1$, $C_j = C_{j-1} + p_j(C_{j-1}) = 1+(1+\alpha_j)C_{j-1}$ for $j = 1, 2, \ldots, n$.

We will apply two l_p-norm related measures [9] to the vector $C(a) = [C_0, C_1, \ldots, C_n]^T$, namely $F(a) = \sum_{j=1}^{n} \sum_{i=1}^{j} a_i \ldots a_j = \|C(a)\|_1 - (n+1)$ and $M(a) = 1 + \sum_{i=1}^{n} a_i \ldots a_n = \|C(a)\|_\infty$.

We will refer to minimizing of $F(a)$ as to $F(a)$-problem. According to [14], if $a = a_\pi$ is the optimal solution of the $F(a)$-problem, then $\bar{a} = (a_n, a_{n-1}, \ldots, a_1)$ is also optimal solution of the $F(a)$-problem. Moreover, such optimal sequence a_π must have a V-shape [14]. (This means that the sequence is non-increasing for $j = 1, \ldots, m$, and is non-decreasing for $j = m, \ldots, n$, where $1 < m < n$. We will call such sequences V-sequences).

For a given sequence $a = (a_1, \ldots, a_n)$, define two *signatures* of a:

$$S^-(a) = M(\bar{a}) - M(a) \quad \text{i} \quad S^+(a) = M(\bar{a}) + M(a)$$

We construct an algorithm for $F(a)$-problem using notion of *signatures* and the following lemma:

Lemma. For a given sequence a and two numbers α, β, let $a^L = (\alpha|a|\beta)$ oraz $a^R = (\beta|a|\alpha)$. Then there hold the equalities:

$$F(\alpha|a|\beta) - F(\beta|a|\alpha) = (\alpha - \beta)S^-(a),$$

$$\tfrac{1}{2}(F(\alpha|a|\beta) + F(\beta|a|\alpha)) = F(a) + \tfrac{1}{2}(\alpha + \beta)S^+(a) + \alpha \cdot a \cdot \beta.$$

Though the proposed algorithm is not optimal in general, we suppose that it is optimal for arithmetic sequences. On the ground of obtained results we proposed a number of formulae describing signatures for the case of the sequences when $a_j = j + 1$, $j = 1, 2, \ldots, n$. These formulae, together with conducted computational experiments, lead to another conjecture: in the case of arithmetic sequences one can construct an optimal schedule for the problem without calculating signatures $S^-(u)$.