ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1984
Seria: AUTOMATYKA z. 74 Nr kol. 810

Eugeniusz Nowicki, Stanisław Zdrzałka
Politechnika Wrocławska
Instytut Cybernetyki Technicznej

SZEREGOWANIE ZADAŃ NA MASZYNIE PRZY ZMIENNYCH CZASACH WYKONYWANIA ZADAŃ

Streszczenie. Sformułowano klasę zagadnień szeregowania n zadań na maszynie o przepustowości n z równoczesnym rozdziałem zasobów odnawialnych. Prędkość wykonywania zadań jest ciągłą monotoniczną funkcją ilości dostarczonych zasobów. Należy znaleźć rozdział zasobów pomiędzy zadania oraz ich uszeregowanie minimalizujące jedną z następujących funkcji celu: średni ważony czas przebywania zadań na maszynie, maksymalny czas przebywania zadań na maszynie przy zadanych najwcześniejszych terminach rozpoczęcia wykonywania zadań, maksymalną nieterminowość przy zadanych pożądanych czasach zakończenia wykonywania zadań z zerowymi i niezerowymi najwcześniejszymi terminami rozpoczęcia wykonywania zadań. W pracy przedstawiono ogólną metodę rozwiązania tych zagadnień. Pokazano również, że dla pewnej klasy zadań powyższe zagadnienia sprowadzają się do odpowiednich klasycznych zadań szeregowania na jednej maszynie o przepustowości 1, dla których istnieją bądź algorytmy wielomianowe, bądź w przypadku gdy problemy są NP-zupełne, dobre algorytmy heurystyczne.

1. Wstęp

W większości badanych zagadnień szeregowania zadań na maszynach zakłada się, że czasy wykonywania zadań są ustalone, w sensie deterministycznym bądź stochastycznym. W rzeczywistości występuje wiele takich sytuacji, w których czas wykonywania zadania na danej maszynie może być zmieniany poprzez odpowiedni dopływ zasobów. Zagadnienie tego typu rozpatrywano w pracach [6,7,10,13,14,15]. Problemy rozpatrywane przez Vicksona [13,14] są najbardziej zbliżone do klasycznych problemów szeregowania. Zakłada się w nich, że czas wykonywania zadania można skrócić ponosząc pewien dodatkowy koszt, który jest liniową funkcją czasu skrócenia. Problem polega na znalezieniu uszeregowania zadań oraz czasów skrócenia minimalizujących koszt przebywania zadań na maszynie plus koszt skrócenia zadań. Nash i Gittins [6,7] dyskutują problem, w którym maszyna może wykonywać równocześnie n zadań, przy czym intensywność wykonywania zadań zmieniana jest w sposób ciągły a rozmiary zadań są zmiennymi losowymi o znanych rozkładach. Węglarz w [15] analizuje problem, w którym prędkość wykonywania zadania jest ciągłą funkcją ilości zasobu i w każdej chwili nie można wykonywać więcej niż m zadań (m identycznych równoległych maszyn). Należy znaleźć rozdział zasobów, który minimalizuje maksymalny czas wykonania zadań.

Niniejsza praca jest rozszerzeniem pracy autorów [10]. Rozpatrujemy klasę zagadnień szeregowania n zadań na maszynie o przepustowości n z równoczesnym rozdziałem zasobów odnawialnych podzielnych w sposób ciągły.

Prędkość wykonywania zadań jest ciągłą niemalejącą funkcją ilości dostarczonych zasobów [8,9,15]. Należy znaleźć rozdział zasobów w czasie pomiędzy zadania oraz ich uszeregowanie minimalizujące jedną z następujących funkcji celu: średni ważony czas przebywania zadań na maszynie, maksymalny czas wykonywania zadań na maszynie przy zadanych najwcześniejszych terminach wykonywania zadań, maksymalną nieterminowość przy zadanych pożądanych czasach zakończenia wykonywania zadań z zerowymi i niezerowymi najwcześniejszymi terminami rozpoczęcia wykonywania zadań. Rozważania przeprowadza się dla dwóch sytuacji: gdy występują ograniczenia kolejnościowe między zadaniami oraz gdy ich brak.

2. Sformułowanie problemu i rezultaty pomocnicze

Zbiór n zadań $J = \{1,2,\ldots,n\}$ ma być wykonany na maszynie, która może w danej chwili czasu wykonywać n zadań. Prędkość wykonywania i-tego zadania, $\dot{x}_i(t)$, w chwili t jest funkcją ilości zasobu $u_i(t)$

$$\begin{aligned} \dot{x}_i(t) &= f_i(u_i(t)), \quad t \geqslant 0, \\ x_i(t_{io}) &= 0 \end{aligned} \tag{1}$$

gdzie $x_i(t)$ określa stan zadania w chwili t, f_i - ciągła niemalejąca funkcja spełniająca $f_i(0) = 0$. Dla każdego zadania określony jest jego stan końcowy $s_i > 0$. Przez t_{io} oznaczamy czas rozpoczęcia wykonywania zadania a przez t_i czas jego zakończenia, tj. najwcześniejszy czas dla którego zachodzi

$$\int_{t_{io}}^{t_i} f_i(u_i(t))dt = s_i. \tag{2}$$

Ilość zasobów dostępna w każdej chwili czasu jest równa jeden, stąd

$$\sum_{i=1}^{n} u_i(t) \leqslant 1, \quad t \geqslant 0. \tag{3}$$

Niech $\prec$ będzie relacją częściowego porządku na zbiorze J, gdzie $i \prec j$ oznacza, że zadanie j może być wykonywane tylko po zakończeniu zadania i, tzn. $t_i \leqslant t_{jo}$.

Zagadnienie 1

Załóżmy, że w chwili $t = 0$ wszystkie zadania są gotowe do wykonywania. Należy znaleźć rozdział zasobów pomiędzy zadania w czasie, tzn. sterowanie w układzie otwartym $u(t) = (u_1(t),\ldots,u_n(t))$, $0 \leqslant t \leqslant T = \max_{1 \leqslant i \leqslant n} t_i$, odcinkami ciągłe spełniające (2), (3) oraz

$$\begin{aligned} &u_i(t) = 0 \quad t < t_{io}, \quad t > t_i, \quad i \in J \\ &i \prec j \Rightarrow t_i \leqslant t_{jo}, \quad i,j \in J \end{aligned} \tag{4}$$

i minimalizujące

$$\sum_{i=1}^{n} w_i t_i \tag{5}$$

gdzie $w_i > 0$ – waga i-tego zadania.

Zagadnienie 2

Przez r_i oznaczmy odpowiednio najwcześniejszy możliwy termin rozpoczęcia wykonywania zadania i (czas gotowości) a przez d_i pożądany termin zakończenia wykonywania tego zadania. Należy znaleźć rozdział zasobów pomiędzy zadania w czasie, tzn. sterowanie w układzie otwartym $u(t) = (u_1(t),\dots,u_n(t))$, $0 \leqslant t \leqslant T = \max_{1 \leqslant i \leqslant n} t_i$, odcinkami ciągłe spełniające (2),(3),(4),(5) oraz

$$t_{io} \geqslant r_i, \quad i \in J \tag{6}$$

i minimalizujące

$$\max_{1 \leqslant i \leqslant n} \{t_i - d_i\}. \tag{7}$$

Charakterystyczną cechą powyższych zagadnień jest fakt, że istnieją przedziały, w których zadania nie mogą być wykonywane, tzn. $u_i(t) = 0$. Wykorzystując to, wprowadzimy teraz szereg pojęć i wykażemy odpowiednie własności, które umożliwią zastosowanie jednolitego podejścia do sformułowanych zagadnień.

Przez $F_k \subset J$ oznaczmy zbiór wszystkich zadań, które mogą być wykonywane w przedziale $(\tau_{k-1},\tau_k]$, $k = 1,\dots,m$. Określmy przez U_k zbiór wektorów $u = (u_1,\dots,u_n) \in R^n$ spełniających $\sum_{i=1}^n u_i \leqslant 1$, $u_i = 0$ jeżeli $i \in J \setminus F_k$, $u_i \geqslant 0$ w przeciwnym wypadku. Niech $\mathcal{U}(\tau_{k-1},\tau_k,F_k)$ oznacza zbiór wszystkich funkcji odcinkami ciągłych $u(\cdot)$ z $(\tau_{k-1},\tau_k]$ w U_k.

Rozbijmy przedział $[0,T]$ na zbiór m rozłącznych przedziałów $[\tau_0,\tau_1],(\tau_1,\tau_2],\dots,(\tau_{m-1},\tau_m]$, $\tau_0 \triangleq 0$, $\tau_m \triangleq T$ o długości $\Delta_k \triangleq \tau_k - \tau_{k-1}$. Dla każdej pary $(\tau_{k-1},\tau_k]$, F_k takiej, że $\Delta_k > 0$ zdefiniujmy zbiór

$$P(\tau_{k-1},\tau_k,F_k) = \left\{v \in R^n : v_i = \frac{1}{\Delta_k}\int_{\tau_{k-1}}^{\tau_k} f_i(u_i(t))dt,\ i=1,\dots,n,\ u(\cdot) \in \mathcal{U}(\tau_{k-1},\tau_k,F_k)\right\}, \tag{8}$$

Niech $X(\tau_{k-1})$ oznacza zbiór wszystkich stanów początkowych w chwili τ_{k-1}. Wtedy zbiór wszystkich stanów osiągalnych w chwili $\tau_k > \tau_{k-1}$ ma postać $X(\tau_k) = X(\tau_{k-1}) + \Delta_k P(\tau_{k-1},\tau_k,F_k)$. Ogólnie, dla danego zestawu par $(\tau_{k-1},\tau_k],F_k$, $k=1,\dots,m$ zbiór wszystkich stanów osiągalnych w chwili T jest sumą arytmetyczną zbiorów $\Delta_k P(\tau_{k-1},\tau_k,F_k)$ dla tych k, dla których $\Delta_k > 0$. Zdefiniujmy na koniec zbiór

$$V_k = \{v \in R^n : v_i = f_i(u_i),\ i=1,\dots,n,\ u \in U_k\} \tag{9}$$

Poniższy lemat pozwala określić postać zbioru $P(\tau_{k-1},\tau_k,F_k)$.

Lemat 1

$$P(\tau_{k-1},\tau_k,F_k) = \text{co } V_k \text{ dla każdego } \Delta_k > 0$$

Dowód lematu można przeprowadzić podobnie jak w pracy [8].

Pewne szczególne własności rozpatrywanych zagadnień otrzymuje się przy założeniu, że

$$f_i(u_i) \leqslant u_i f_i(1) \quad u_i \in [0,1] \tag{10}$$

Warto zauważyć, że funkcje wypukłe spełniają warunek (10).

Lemat 2

Niech $\bar{v}^j = (0,\dots,0,f_j(1),0,\dots,0)$, $j=1,\dots,n$, $\bar{v}^0 = (0,\dots,0)$. Jeżeli funkcje f_i, $i=1,\dots,n$ spełniają (10), to co V_k jest zbiorem wszystkich kombinacji wypukłych punktów $\bar{v}^0, \bar{v}^j$, $j \in F_k$.

Dowód lematu wynika bezpośrednio z własności (10) oraz definicji powłoki wypukłej.

Wykorzystując powyżej wprowadzone pojęcia oraz własności, w następnych rozdziałach omówimy sformułowane zagadnienia.

3. Minimalizacja średniego ważonego czasu przebywania zadań na maszynie

Niech π będzie pewną permutacją ze zbioru J spełniającą $\prec$ a $\pi(i)$ zadaniem ze zbioru J znajdującym się na i-tej pozycji. Permutacja π spełnia $\prec$ wtedy i tylko wtedy, gdy z tego, że $l \prec s$ wynika, że $i < j$ gdzie $l = \pi(i)$, $s = \pi(j)$. Załóżmy teraz, że zadania kończą się wykonywać w kolejności określonej przez permutacje π, tzn.

$$t_{\pi(1)} \leqslant t_{\pi(2)} \leqslant \dots \leqslant t_{\pi(n)}. \tag{11}$$

Dla ustalonego π zagadnienie 1 z dodatkowym warunkiem (11) będziemy oznaczać przez $\mathcal{P}(\pi)$. W celu rozwiązania zagadnienia 1 należy rozwiązać zadanie $\mathcal{P}(\pi)$ dla wszystkich π spełniających $\prec$ i wybrać to rozwiązanie, które ma najmniejszą wartość funkcji celu (5). W związku z tym przeanalizujemy teraz dokładnie zadanie $\mathcal{P}(\pi)$ dla wybranej arbitralnie permutacji π spełniającej $\prec$. W tym celu połóżmy $m = n$, $\tau_k = t_{\pi(k)}$, $k=1,\dots,r$ i niech

$$F_k = \{i \in \{\pi(k),\dots,\pi(n)\} : j \prec i,\ j \in \{\pi(k),\dots,\pi(n)\}\}$$

Z lematu 1 wynika, że dla ustalonych $t_{\pi(k)}$, $k=1,\dots,n$ zbiór stanów osiągalnych w chwili $t_{\pi(k)}$ ma postać

$$\Delta_1 \operatorname{co} V_1 + \Delta_2 \operatorname{co} V_2 + \dots + \Delta_k \operatorname{co} V_n,$$

gdzie $\Delta_k = t_{\pi(k)} - t_{\pi(k-1)}$, $t_{\pi(0)} \triangleq 0$ i V_k są zdefiniowane przez (9). Uwzględniając to, zadanie $\mathcal{P}(\pi)$ może być przedstawione następująco:

$$\min_{\{\Delta_k, v^k\}} \Delta_1 \sum_{k=1}^{n} w_{\pi(k)} + \Delta_2 \sum_{k=2}^{n} w_{\pi(k)} + \dots + \Delta_n w_{(n)} \tag{12}$$

przy ograniczeniach

(i) $$\sum_{k=1}^{n} \Delta_k v_i^k = s_i,\ i \in J, \tag{13}$$

(ii) $$v^k \in \operatorname{co} V^k,\ \Delta_k \geqslant 0,\ k=1,\dots,n. \tag{14}$$

W ogólnym wypadku powyższe zadanie jest trudne ze względu na ograniczenie (ii) ponieważ nie można w analitycznej postaci przedstawić powłoki wypukłej zbioru V^k. Istnieje jednak wiele interesujących przypadków, w których zadanie to można rozwiązać i rozwiązanie podać w analitycznej postaci. Do jednego z nich należy przypadek, gdy funkcje f_i spełniają (10).

Z lematu 2 wynika, że dla każdego $v^k \in coV^k$ istnieją λ_0^k, λ_j^k, $j \in F_k$ nieujemne i takie, że $\lambda_0^k + \sum_{j \in F_k} \lambda_j^k = 1$, $v^k = \lambda_0^k \bar{v}^0 + \sum_{j \in F_k} \lambda_j^k \bar{v}^j$. Korzystając z tego faktu i oznaczając przez $Q_i = \{\alpha_i, \alpha_i+1, \ldots, \beta_i\}$ zbiór przedziałów, w których może być wykonywane zadanie i-te, $i \in J$, oraz kładąc $\Delta_j^k = \lambda_j^k \Delta_k$ zadanie (12),(13),(14) można przedstawić w postaci

$$\min_{\{\Delta_i^k\}} \sum_{i=1}^n \sum_{k \in Q_i} \left(\Delta_i^k \sum_{s=k}^n w_{\pi(s)}\right) + \sum_{k=1}^n \Delta_0^k \sum_{s=k}^n w_{\pi(s)} \tag{15}$$

przy ograniczeniach

$$\text{(i)} \qquad \sum_{k \in Q_i} \Delta_i^k = s_i/f_i(1), \; i \in J, \tag{16}$$

$$\text{(ii)} \qquad \Delta_i^k \geqslant 0, \; k \in Q_i, \; i \in J, \; \Delta_0^k \geqslant 0, \; k \in J. \tag{17}$$

Rozwiązaniem powyższego zadania programowania liniowego jest $\Delta_i^{*k} = 0$ dla $k = \alpha_i, \alpha_i+1, \ldots, \beta_i-1$, $\Delta_i^{*k} = s_i/f_i(1)$ dla $k = \beta_i$, $i \in J$, ponadto, $\Delta_0^{*k} = 0$ dla $k \in J$. Minimalna wartość funkcji celu dla problemu $\mathcal{P}(\pi)$ ma zatem postać

$$Q(\pi) = \sum_{k=1}^n \frac{s_{\pi(k)}}{f_{\pi(k)}(1)} \sum_{s=k}^n w_{\pi(s)} = \sum_{k=1}^n p_{\pi(k)} \sum_{s=k}^n w_{\pi(s)}, \tag{18}$$

gdzie $p_{\pi(k)} = s_{\pi(k)}/f_{\pi(k)}(1)$ jest czasem wykonywania zadania $\pi(k)$. Z powyższych rozważań wynika, że jeżeli wszystkie zadania spełniają warunek (10), to dla ustalonej kolejności kończenia zadań π optymalne sterowanie polega na wykonywaniu zadań szeregowo bez przerwań w kolejności zgodnej z π z maksymalną intensywnością. Teraz należy znaleźć permutację spełniającą $\prec$, dla której (18) przyjmuje wartość minimalną. Jest to klasyczne zadanie szeregowania na jednej maszynie z ograniczeniami kolejnościowymi zapisywane symbolicznie jako $n|1|\prec|\sum w_i C_i$. W pracy [5] pokazano, że zagadnienie to jest NP-zupełne, chociaż dla szczególnej postaci ograniczeń kolejnościowych, np. ograniczeń typu drzewa, ograniczeń szeregowo-równoległych, istnieją algorytmy o wielomianowej złożoności obliczeniowej [2], [11]. W przypadku, gdy $\prec = \emptyset$ minimum (18) osiągane jest dla permutacji σ, dla której zachodzi $p_{\sigma(1)}/w_{\sigma(1)} \leqslant \cdots \leqslant p_{\sigma(n)}/w_{\sigma(n)}$, [12]. Wynika stąd szczególna postać rozwiązania rozważanego zagadnienia 1.

Własność 1

Jeżeli funkcje f_i spełniają warunek (10) oraz $\prec = \emptyset$, to sterowanie $u^*(t)$, $0 \leqslant t \leqslant T^*$, określone następująco

$$u^*_{\sigma(k)}(t) = \begin{cases} 1, & t \in (t^*_{k-1}, t^*_k] \\ 0, & \text{w przeciwnym przypadku} \end{cases}, \quad k = 1,\dots,n$$

gdzie $t^*_k = \sum_{j=1}^{k} p_{\sigma(j)}$, $T^* = t^*_n$, σ - permutacja spełniająca $p_{\sigma(1)}/w_{\sigma(1)} \leq \dots \leq p_{\sigma(n)}/w_{\sigma(n)}$, jest sterowaniem optymalnym.

4. Minimalizacja czasu zakończenia wykonywania przy zadanych czasach gotowości

Rozpatrzymy teraz zagadnienie 2 przy założeniu $d_i = 0$, $i \in J$. Załóżmy, że zadania ponumerowane są wg kolejności ich pojawiania się, tzn. $r_1 \leq r_2 \leq \dots \leq r_n$. Podobnie jak poprzednio ustalamy najpierw permutacje π spełniającą $\prec$ i określającą kolejność wykonywania poszczególnych zadań (11). Ponadto w każdym z przedziałów $[r_1, r_2), \dots, [r_{n-1}, r_n), [r_n, \infty)$ umiejscawiamy poszczególne terminy wykonania zadań w ten sposób, żeby spełniony był warunek (11) oraz $r_{\pi(i)} \leq t_{\pi(i)}$. Należy zauważyć, że podobnie jak w przypadku kolejności π, jest wiele różnych możliwości dokonania takiej alokacji terminów wykonania zadań. Oznaczmy dalej przez R_k zbiór tych zadań, które kończą się w przedziale $[r_k, r_{k+1})$, $k = 1,\dots,n$, $r_{n+1} \triangleq \infty$. Mamy zatem $r_k \leq t_{\pi(\alpha_k)} \leq t_{\pi(\alpha_{k+1})} \leq \dots \leq t_{\pi(\beta_k)} < r_{k+1}$, gdzie $\{\pi(\alpha_k), \pi(\alpha_k+1), \dots, \pi(\beta_k)\} = R_k$. Oznaczmy dalej przez $F_{k,j}$, $j = 1,\dots,l_k+1$, $l_k \triangleq |R_k|$, zbiór tych zadań, które mogą być wykonywane równolegle odpowiednio w przedziałach $[r_k, t_{\pi(\alpha_k)}), \dots, [t_{\pi(\beta_k-1)}, t_{\pi(\beta_k)}), [t_{\pi(\beta_k)}, r_{k+1})$. Jeżeli $l_k = 0$, to $F_{k,1}$ określone jest w oparciu o przedział $[r_k, r_{k+1})$. Zauważmy dalej, że w zaproponowanym rozbiciu osi czasowej ustalone są tylko punkty r_k. Teraz możemy już skorzystać z lematu 1 i sformułować problem optymalizacji dla ustalonych π oraz R_k, $k = 1,\dots,n$. Problem ten oznaczymy przez $\mathcal{P}(\pi, R_k)$. Oznaczamy $\Delta_{k,1} = t_{\pi(\alpha_k)} - r_k$, $\Delta_{k,2} = t_{\pi(\alpha_k+1)} - t_{\pi(\alpha_k)}, \dots \Delta_{k,l_k} = t_{\pi(\beta_k)} - t_{\pi(\beta_k-1)}$, $\Delta_{k,l_k+1} = r_{k+1} - t_{\pi(\beta_k)}$ przy $\Delta_{n,l_n+1} \triangleq 0$. W przypadku, gdy $l_k = 0$, $\Delta_{k,1} = r_{k+1} - r_k$.

Otrzymujemy następujący problem optymalizacyjny

$$\min_{\{\Delta_{k,j}\}} \sum_{j=1}^{l_n} \Delta_{n,j}$$

przy ograniczeniach

(i) $$\sum_{k=1}^{n-1} \sum_{j=1}^{l_k+1} \Delta_{k,j} v_i^{k,j} + \sum_{j=1}^{l_n} \Delta_{n,j} v_i^{k,j} = s_i, \quad i \in J$$

(ii) $$v^{k,j} \in \operatorname{co} V^{k,j}, \ \Delta_{k,j} \geq 0, \quad j=1,\dots,l_k+1, \ k=1,\dots,n-1, \quad j=1,\dots,l_k, \ k=n$$

(iii) $$\sum_{j=1}^{l_k+1} \Delta_{k,j} = r_{k+1} - r_k, \quad k=1,\dots,n-1.$$

gdzie zbiór $V^{k,j}$ zdefiniowany jest tak samo jak V^k w poprzednim rozdziale, z tym, że w oparciu o zbiory $F_{k,j}$. Rozwiązanie przedstawionego zagad-

nienia otrzymamy rozwiązując każdy problem $\mathcal{P}(\pi, R_k)$ (tzn. dla wszystkich permutacji π spełniających $\prec$, a przy ustalonej π dla wszystkich możliwych zestawów R_k, $k=1,\ldots,n$). Oczywiście, nie jest to metoda efektywna. Okazuje się jednak, że w przypadku, gdy funkcje f_i spełniają warunek (10), wtedy wykorzystując lemat 2, zagadnienie to można sprowadzić, w podobny sposób jak to zrobiono w poprzednim rozdziale, do klasycznego zagadnienia szeregowania na jednej maszynie z przerwaniami oraz ograniczeniami kolejnościowymi, w którym czasy wykonywania zadań dane są jako $p_i = s_i/f_i(1)$. Zagadnienie to oznaczane jest w literaturze symbolem $n|1|r_i \geqslant 0$, przer., $\prec | C_{max}$. Dla jego rozwiązania można zastosować algorytm o złożoności obliczeniowej $O(n^2)$ sformułowany dla ogólniejszego zagadnienia $n|1|r_i \geqslant 0$, przer., $\prec | f_{max}$ w pracy [1]. Algorytm ten przebiega w następujący sposób. Najpierw, należy zmodyfikować r_i w ten sposób, żeby $r_k \geqslant r_j + p_j$ jeżeli $j \prec k$. Następnie szeregujemy zadanie zgodnie z niemalejącymi r_i. Uporządkowanie takie tworzy pewną ilość bloków, tzn. zestawów zadań wykonywanych kolejno bez przestojów. W bloku określamy zadanie k-te, które nie ma w nim następników, i które postawione na ostatniej pozycji bloku daje minimalną wartość funkcji celu. Jeżeli zadanie k-te nie jest ostatnim zadaniem w bloku, to wyciągamy je z bloku, przeszeregowujemy pozostałe zadanie z bloku wg niemalejących r_i i w otrzymane przedziały czasowe, w których maszyna ma przestoje umieszczamy zadanie k-te. Powtarzając tę procedurę dla każdego bloku i dla każdego z podbloków otrzymanych wskutek kolejnych przeszeregowań uzyskujemy uszeregowanie optymalne, z którego możemy ponadto wyznaczyć sterowanie optymalne $u^*(t)$, $0 \leqslant t \leqslant T^*$.

W przypadku, gdy zbiór $\prec$ jest pusty, wtedy zgodnie z rozwiązaniem problemu $n|1|r_i \geqslant 0|C_{max}$ [3], który wtedy jest równoważny problemowi $n|1|r_i \geqslant 0$, przer.$|C_{max}$ otrzymujemy następujące rozwiązanie optymalne.

Własność 2

Jeżeli funkcje f_i spełniają warunek (10), $\prec = \emptyset$, $d_i = 0$, $i=1,\ldots,n$, to sterowanie $u^*(t)$, $0 \leqslant t \leqslant T^*$ określone następująco:

$$u^*_{\sigma(k)}(t) = \begin{cases} 1, & t \in (t^*_k, t^*_k + p_{\sigma(k)}] \\ 0, & \text{w przeciwnym przypadku} \end{cases}$$

gdzie $t^*_1 = r_{\sigma(1)}$, $t^*_k = \max\{r_{\sigma(k)}, t^*_{k-1} + p_{\sigma(k-1)}$, $k=2,\ldots,n$, $T^* = t^*_n + p_{\sigma(k)}$, σ - dowolna permutacja spełniająca $r_{\sigma(1)} \leqslant \cdots \leqslant r_{\sigma(k)}$, jest sterowaniem optymalnym w zagadnieniu 2.

Własność 2 mówi, że sterowanie optymalne polega na wykonywaniu zadań szeregowo z maksymalną intensywnością wg kolejności wyznaczonej przez niemalejące r_i.

Podobny wniosek można otrzymać dla zagadnienia 2, w którym r_i = const oraz $d_i \geqslant 0$. W tym przypadku, jeżeli $\prec = \emptyset$ oraz f_i spełniają (10) zagadnienie sprowadza się do problemu $n|1||L_{max}$ a optymalne sterowanie polega na wykonywaniu zadań szeregowo, z maksymalną intensywnością w kolejności

określonej przez niemalejące d_i.

W przypadku $r_i \geq 0$, $d_i \geq 0$ spełnienie warunku (10) prowadzi do problemu $n|1|r_i \geq 0$, przer., $< | L_{max}$, który można rozwiązać wykorzystując podany poprzednio algorytm z pracy [1].

5. Uwagi końcowe

Sformułowane zagadnienia szeregowania n zadań na maszynie o przepustowości n ze sterowalnymi czasami wykonywania zadań charakteryzują się w swych ogólnych postaciach dużą złożonością obliczeniową i autorzy nie potrafili znaleźć dla nich efektywnych metod obliczeniowych. Wskazano tylko ogólną metodę podejścia do ich rozwiązywania polegającą na sformułowaniu odpowiednich zadań będących połączeniem zadań programowania nieliniowego z zadaniami kombinatorycznymi. Okazało się jednak, że w przypadku spełnienia pewnych założeń odnośnie funkcji f_i (warunek (10)) zagadnienia te sprowadzają się do odpowiednich zagadnień szeregowania zadań na jednej maszynie o przepustowości 1 z przerwaniami lub bez przerwań i ograniczeniami kolejnościowymi, dla których istnieją w literaturze efektywne, o wielomianowej złożoności obliczeniowej, algorytmy.

LITERATURA

[1] Baker R.R., Lowler E.L., Lenstra J.K., Rinnooy Kan A.M.G., Preemptive scheduling of a simple machine to minimize maximum cost subject to release dates and precedence constraints, Econometric Institute, Erasmus University, Rotterdam Report 8028/0, 1980.

[2] Horn W.A., Single-machine job sequencing with treelike precedence ordering and linear delay penalties, SIAM J.Appl.Math., 23 (1972), 189-202.

[3] Jackson J.R., Scheduling a production line to minimize maximum tardiness, Research Report 43, Management Science Research Project, University of California, Los Angeles, 1955.

[4] Lawler E.L., Sequencing jobs to minimize total weighted completion time subject to precedence constraints, Ann. Discrete Math. 2 (1978), 75-90.

[5] Lenstra J.K., Sequencing by enumerative methods, North-Holland, American Elsevier, Amsterdam 1978.

[6] Nash P., Gittins J.C., A Hamiltonian approach to stochastic resource allocation, Adv. Appl. Prob. 9 (1977), 55-68.

[7] Nash P., Controlled jump process models for stochastic scheduling, Int. J. Control, 29 (1979), 1011-1025.

[8] Nowicki E., Zdrzałka S., Optimal Control of a Complex of Independent Operations, Int. J. Systems Sci. 12 (1981), 77-93.

[9] Nowicki E., Zdrzałka S., Optimal control policies for resource allocation in an activity network, European J. Oper. Res. 16, No 2 (1984).

[10] Nowicki E., Zdrzałka S., Scheduling jobs with controllable processing times as an optimal control problem, Int. J. Control, w druku.

[11] Sidney J.B., Decomposition algorithms for single-machine sequencing with precedence relations and deferral costs, Oper. Res., 23 (1975), 283-299.

[12] Smith W.E., Various optimizers for single-stage production, Naval Res Logist. Quart. 3 (1956), 59-66.

[13] Vickson R.G., Choosing the job sequence and processing times to minimize total processing plus flow cost on a single machine, Oper. Res., 28 (1980), 1155-1167.

[14] Vickson R.G., Two single machine sequencing problems involving controllable job processing times, AIIE Transactions, 12 (1980), 258--262.

[15] Węglarz J., Project Scheduling with Discrete and Continuous Resources, IEEE Trans. Systems Man., Cybernetics, SMC-9 (1979), 644-650.

Recenzent: Doc.dr hab.inż.Konrad Wala
Wpłynęło do Redakcji do 30.03.1984r.

РАСПИСАНИЕ ЗАДАЧ НА МАШИНЕ ПРИ ИЗМЕНЯЮЩИМСЯ ВРЕМЕНИ ИСПОЛНЕНИЯ ЗАДАЧ

Р е з ю м е

Сформулирован класс задач расписания n заданий на машине с пропускной способностью n, с одновременным распределением восстанавливаемых ресурсов. Скорость выполнения заданий является непрерывной монотонной функцией количества ресурсов. Необходимо найти распределение ресурсов между заданиями а также расписание заданий, минимизирующее один из следующих критериев: среднее взвешенное время пребывания заданий в системе, максимальное время пребывания и максимальную несрочность. В статье представлен общий метод решения этих задач. Показано также, что для некоторого класса задач такая проблема сводится к классическим проблемам расписания задач на одной машине, для которых существуют полиномиальные алгоритмы, или-же в случае NP - трудных проблем, хорошие эвристические алгоритмы.

SCHEDULING JOBS WITH VARYING PROCESSING TIMES

S u m m a r y

A problem of scheduling n jobs on a machine with capacity n under resource constraints is formulated. A job processing intensity is a monotone, continuous function of an instantaneous resource usage. The problem is to find resource allocation and job schedule minimizing one of the following cost functions: average flow time, makespan under given release dates, and maximum lateness under zero or nonzero release dates.