ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2000
Seria: AUTOMATYKA z. 129 Nr kol. 1474

Adam JANIAK, Piotr SŁONIŃSKI
Politechnika Wrocławska

PROBLEM SZEREGOWANIA ZADAŃ WYKONYWANYCH NA JEDNEJ MASZYNIE Z DYNAMICZNYM MODELEM TERMINÓW ICH DOSTĘPNOŚCI

Streszczenie. W niniejszej pracy zaprezentowano rozwiązanie problemu minimalizacji czasu zakończenia wykonywania zbioru n zadań o dynamicznych modelach terminów dostępności na pojedynczej maszynie krytycznej. Dane jest ograniczenie na ilość zasobu dostępną do rozdysponowania w danej chwili. Wykazano szereg istotnych własności tego problemu, a na ich podstawie skonstruowano algorytm optymalnego rozdziału zasobu dla zadań w ustalonej permutacji oraz algorytm aproksymacyjny szeregowania zadań.

A SCHEDULING PROBLEM WITH JOBS PROCESSED ON A SINGLE MACHINE WITH DYNAMIC MODEL OF THEIR RELEASE DATES

Summary. The aim of this contribution is to present the solution of the problem of minimizing the time of processing a set of n jobs with dynamical (differential) models of job release dates on a single critical machine. The amount of resource available at each moment is known a priori. Many important properties of this problem have been proven. They are the base for construction of optimal resource allocation algorithm for jobs processed in a given permutation. There is also presented the approximation algorithm for the scheduling problem.

1. Wprowadzenie

Klasyczna teoria szeregowania zadań zakłada, że czasy wykonywania zadań oraz ich terminy dostępności (czyli czas, jaki musi upłynąć, aby można rozpocząć wykonywanie zadań) są stałe. Ponadto nie bierze się pod uwagę wpływu ilości zużytych zasobów na wartość rozpatrywanej funkcji kryterialnej.

Jednak w bardzo wielu przypadkach zachodzi potrzeba rozpatrywania znacznie ogólniejszego modelu, tj. takiego, w którym **tempo** zmiany stanu pewnego procesu produkcyjnego jest zależne dynamicznie od ilości dostarczonego zasobu. Z procesami tego typu spotykamy się bardzo często w zastosowaniach przemysłowych (np. podgrzewanie wlewków hutniczych w piecach wgłębnych) i informatycznych (np. szeregowanie zadań na równoległych procesorach). Model dynamiczny, w odniesieniu do czasów wykonywania zadań, został wprowadzony przez Burkova [1] oraz wykorzystywany między innymi w badaniach Janiaka [3], Węglarza [6], Józefowskiej [5] i Przysady [2].

W niniejszej pracy przedstawimy wyniki badań dotyczących dynamicznych modeli terminów dostępności. Zaprezentowana zostanie analiza procesów, w których wykonywanie właściwej części zadania może rozpocząć się dopiero po zakończeniu pewnego wstępnego procesu przygotowawczego. Tempo zmiany tego procesu przygotowawczego jest zależne dynamicznie od ilości przydzielonego zasobu ciągłego i może być ciągłą, nieujemną i ściśle rosnącą funkcją zasobu. Podamy rozwiązanie problemu minimalizacji czasu zakończenia wykonywania wszystkich zadań przy ograniczeniu na ilość zasobu dostępnego do rozdysponowania w danej chwili.

W rozdziale 2 sformułowano precyzyjnie rozpatrywany tutaj problem. W rozdziale 3 wykazano szereg własności tego problemu. Wykazano, między innymi, że problem ten nawet dla identycznych dynamicznych modeli terminów dostępności zadań jest NP-trudny. Wykazano także, że problem ten jest silnie NP-trudny dla różnych modeli tych terminów. W rozdziale 4, wykorzystując wykazane w rozdziale 3 własności, sprowadzono skomplikowany problem optymalnego rozdziału zasobu dla ustalonej kolejności wykonywania zadań (będący trudnym problemem optymalizacji dynamicznej) do znacznie łatwiejszego zagadnienia programowania wypukłego. Do rozwiązania rozpatrywanego w tej pracy problemu w rozdziale 5 zaproponowano kilka algorytmów aproksymacyjnych i przeprowadzono ich analizę eksperymentalną i najgorszego przypadku. W rozdziale 6 podano możliwości uogólnienia rozważanego problemu na modele terminów dostępności składające się z części stałej i dynamicznej.

2. Sformułowanie problemu

Rozpatrywany w niniejszej pracy problem można sformułować w następujący sposób. Dany jest zbiór niepodzielnych n zadań $J = \{1, 2, \ldots, n\}$ do wykonania na jednej maszynie (procesorze). Zadanie J_j (lub krótko: zadanie j) ze zbioru J wykonywane jest przez p_j, $j = 1, 2, \ldots, n$ jednostek czasu. Każde zadanie j przed zasadniczym wykonaniem na rozpatrywanej maszynie podlega pewnemu wstępnemu procesowi obróbki (np. na maszynach poprzedzających rozpatrywaną maszynę), który dalej będzie interpretowany jako termin dostępności tego zadania w celu zasadniczego wykonania na rozpatrywanej maszynie.

Model terminu dostępności r_j zadania j opisany jest następującym równaniem różniczkowym:

$$\dot{x}_j^r(t) = \frac{dx_j^r(t)}{dt} = f_j\left(u_j(t)\right), \quad x_j^r(t_0 = 0) = 0, \quad j = 1, 2, \ldots, n, \tag{1}$$

gdzie:

- $x_j^r(t)$ – stan wstępnego procesu determinującego termin dostępności zadania j w chwili t, $x_j(t_0 = 0) = 0$ jest stanem początkowym tego procesu,
- $u_j(t) \in [0, \hat{R}]$ – ilość zasobu przydzielona do terminu dostępności zadania j w chwili t, a $\hat{R}$ jest stałym w czasie ograniczeniem na ilość zasobu odnawialnego,
- $f_j(\cdot)$ – funkcja ciągła, niemalejąca, $f_j(0) = 0$.

Ponadto zadany jest stan końcowy $\hat{x}_j^r$ tego wstępnego procesu, który musi być osiągnięty, aby zadanie stało się dostępne do obróbki na rozpatrywanej maszynie. Innymi słowy zachodzi następująca zależność:

$$\int_0^{r_j} f_j(u_j(t))\, dt = \hat{x}_j^r, \quad j = 1, 2, \ldots, n. \tag{2}$$

Po doprowadzeniu terminu dostępności zadania j do stanu końcowego $\hat{x}_j^r$ można rozpocząć właściwe wykonywanie zadania j, na rozpatrywanej maszynie, trwające p_j jednostek czasu. Przez C_j będziemy oznaczać (nieznany a priori) moment zakończenia wykonywania zadania j, a przez r_j, jak już wspomnieliśmy (nieznany a priori), termin dostępności zadania j.

Dla ustalonej kolejności wykonywania zadań przez rozdział zasobu będziemy rozumieli odcinkami ciągłą, nieujemną funkcję wektorową

$$\mathrm{u} \triangleq \mathrm{u}(t) \triangleq [u_1(t), u_2(t), \ldots, u_n(t)].$$

Niech zbiór U^t będzie zbiorem dopuszczalnych rozdziałów zasobu $\mathbf{u}$. Rozdział zasobu $\mathbf{u}(t)$ jest dopuszczalny, gdy spełniony jest następujący warunek:

$$\sum_{j=1}^{n} u_j(t) \leqslant \hat{R}, \qquad t \in [0, C_{\max}].$$

Zadane mogą być również technologiczne ograniczenia kolejnościowe pomiędzy zadaniami. Zbiór wszystkich dopuszczalnych kolejności z wykonywania zadań będziemy oznaczali $\mathbf{Z}$.

Dla rozpatrywanego tutaj problemu (zapisanego w notacji trójpolowej [4])

$$1 \left| \prec, \dot{x}_j^r(t) = f_j(u_j(t)), \sum u_j(t) \leqslant \hat{R} \right| C_{\max} \tag{3}$$

należy znaleźć optymalne rozwiązanie $(\mathbf{z}^*, \mathbf{u}^*(t)), \mathbf{z}^* \in \mathbf{Z}, \mathbf{u}^*(t) \in \mathbf{U}^t$, tzn. taką dopuszczalną kolejność wykonywania zadań $\mathbf{z}^* \in \mathbf{Z}$ oraz taki dopuszczalny rozdział zasobu $\mathbf{u}^*(t) \in \mathbf{U}^t$, aby czas zakończenia wykonywania wszystkich zadań $C_{\max}(\mathbf{z}^*, \mathbf{u}^*(t))$ był minimalny (tzn. $\min_{\mathbf{z}\in\mathbf{Z}} \min_{\mathbf{u}(t)\in\mathbf{U}^t} C_{\max}(\mathbf{z}, \mathbf{u}(t)) = C_{\max}(\mathbf{z}^*, \mathbf{u}^*(t))$).

3. Własności problemu

Niniejszy rozdział rozpoczniemy od określenia złożoności obliczeniowej rozpatrywanego problemu (3).

Twierdzenie 1. *Problem* $1 \left| \prec, \dot{x}_j^r(t) = f(u_j(t)), \sum u_j(t) \leqslant \hat{R} \right| C_{\max}$ *jest NP-trudny, a problem* $1 \left| \dot{x}_j^r(t) = f_j(u_j(t)), \sum u_j(t) \leqslant \hat{R} \right| C_{\max}$ *jest silnie NP-trudny.*

Ze względu na ograniczenia nałożone na objętość pracy dowód pomijamy.

Łatwo można pokazać, że dla ustalonego uszeregowania $\mathbf{z} \in \mathbf{Z}$ zadania mogą być wykonywane na procesorze jedno po drugim, bez żadnych przerw między nimi. Zauważmy bowiem, że jeśli mamy do czynienia z sytuacją, w której $C_j = r_j + p_j < r_{j+1}$ (tzn. istnieje przerwa między zakończeniem wykonywania zadania j a rozpoczęciem wykonywania zadania $j+1$), wówczas wystarczy wykonywanie zadania j rozpocząć o $r_{j+1} - C_j$ jednostek czasu później. Oznaczmy dodatkowo przez $t_0 \triangleq r_0 \triangleq 0$ chwilę rozpoczęcia wstępnego procesu wyznaczającego termin dostępności zadań.

Wówczas dla ustalonego uszeregowania zadań $\mathbf{z} \in \mathbf{Z}$ oraz dowolnego rozdziału zasobu $\mathbf{u}(t) \in \mathbf{U}^t$ czas zakończenia wykonania wszystkich zadań wyraża się następującym wzorem:

$$C_{\max}(\mathbf{z}, \mathbf{u}(t)) = \Delta r_0 + \sum_{j=1}^{n} p_j,$$

gdzie $\Delta r_0 \triangleq \Delta r_0(\mathbf{z}, \mathbf{u}(t)) \triangleq r_1 - t_0$. Zauważmy, że składnik $\sum_{j=1}^{n} p_j$ powyższej sumy jest wielkością stałą. Zatem minimalizacja kryterium $C_{\max}(\mathbf{z}, \mathbf{u}(t))$ sprowadza się do minimalizacji składnika $\Delta r_0(\mathbf{z}, \mathbf{u}(t))$.

Łatwo sprawdzić, że zbiór dopuszczalnych rozdziałów zasobu U_k w przedziale $[r_k, r_{k+1})$, dla $k = 0, 1, \ldots, n-1$, zdefiniowany następująco:

$$\begin{aligned} U_k \triangleq\ & \{\mathbf{u}(t) : \mathbf{u}(t) = [u_1(t), u_2(t), \ldots, u_n(t)] \wedge \\ & \wedge \forall_{t \in [r_k, r_{k+1})} \sum_{j=1}^{n} u_j(t) \leqslant N, \quad u_j(t) \geqslant 0\}, \end{aligned}$$

jest zbiorem wypukłym.

Dla ustalonego uszeregowania zadań $\mathbf{z} \in \mathbf{Z}$ przez $\mathbf{V}_k$ będziemy oznaczali zbiór dopuszczalnych prędkości realizacji wspomnianego wstępnego procesu, przyjmowanego jako termin dostępności zadań w przedziale $[r_k, r_{k+1})$, który jest zdefiniowany następująco:

$$\begin{aligned} V_k =\ & \Big\{\mathbf{v}(t) : \mathbf{v}(t) = [v_1(t), v_2(t), \ldots, v_n(t)] \wedge \\ & \wedge \forall_{1 \leqslant j \leqslant n} \forall_{t \in [r_k, r_{k+1})} \quad v_j(t) = f_j(u_j(t)) \wedge \\ & \wedge \mathbf{u}(t) \in \mathbf{U}^t\Big\}, \quad k = 0, 1, \ldots, n-1. \end{aligned}$$

$\mathbf{V}_0$ oznacza tutaj zbiór dopuszczalnych prędkości zmiany stanu terminów dostępności zadań w przedziale $[0, r_1)$.

Można łatwo wykazać, że jeśli funkcje f_j, $j = 1, 2, \ldots, n$ są funkcjami wklęsłymi i ściśle rosnącymi, wówczas zbiór dopuszczalnych prędkości $\mathbf{V}_k$ jest zbiorem wypukłym dla $k = 0, 1, \ldots, n-1$. Oznacza to, że otoczka wypukła zbioru $\mathbf{V}_k$ jest identyczna ze zbiorem $\mathbf{V}_k$.

Poniżej podamy dwa twierdzenia, które zostaną wykorzystane w dalszej części pracy do rozwiązania problemu rozdziału zasobu. Pierwsze z nich (podane tutaj bez dowodu) określa postać funkcji kryterialnej (składnika Δr_0).

Twierdzenie 2. *Czas potrzebny na przeprowadzenie terminów dostępności zadań ze stanu* $0 = [0, 0, \dots, 0]$ *do stanu* $\mathrm{x}^0 = [x_{10}^r, x_{20}^r, \dots, x_{n0}^r]$, *gdzie* $\mathrm{x}_{j0}^r = \mathrm{x}_j^r(\Delta r_0)$, *jest minimalny wtedy i tylko wtedy, gdy może być przedstawiony w następującej postaci:*

$$\Delta r_0^{\min} = \Delta r_0^{\min}(\mathrm{x}^0) = \inf \left\{ T > 0 : \frac{\mathrm{x}^0}{T} \in V_0 \right\}. \tag{4}$$

Drugie twierdzenie umożliwia istotne uproszczenie rozważań dotyczących omawianego problemu.

Twierdzenie 3. *Jeżeli w problemie* $1 \left| \dot{x}_j^r(t) = f_j(u_j(t)), \sum u_j(t) \leqslant \hat{R} \right| C_{\max}$, *funkcje* f_j *są wklęsłe i ściśle rosnące oraz dla ustalonej permutacji* $\mathrm{z} \in \mathrm{Z}$, *w przedziale* $[t_k, t_{k+1})$ *istnieje pewien zmienny w czasie rozdział zasobu* $\bar{\mathrm{u}}(t) \in \mathrm{U}_k$, *powodujący przejście terminów dostępności ze stanu* $x(t_k)$ *do stanu* $x(t_{k+1})$ *w czasie* $\Delta t_k = t_{k+1} - t_k$, *to istnieje w tym przedziale również stały w czasie rozdział* $\mathrm{u}(t) = \mathrm{u} = const$ $(\mathrm{u} \in \mathrm{U}_k)$ *równoważny* $\bar{\mathrm{u}}(t)$, *powodujący przejście z* $x(t_k)$ *do* $x(t_{k+1})$ *w tym samym czasie* Δt_k *co* $\bar{\mathrm{u}}(t)$.

Dowód. *Niech rozdziałowi zasobu* $\bar{\mathrm{u}}(t)$ *w przedziale* $[t_k, t_{k+1})$ *odpowiada prędkość zmiany stanu terminów dostępności* $\bar{\mathrm{v}}(t)$.

Załóżmy teraz, że $\bar{\mathrm{v}}(t) \in V_k$ *jest funkcją schodkową. Podzielmy przedział* $[t_k, t_{k+1})$ *na* l *podprzedziałów o długościach* $\Delta s_i = s_{i+1} - s_i$ $(t_k \leqslant s_i < t_{k+1})$, *przy czym* $\sum_{i=1}^{l} \Delta s_i = t_{k+1} - t_k$. *Niech w podprzedziale* $[s_i, s_{i+1})$ *funkcja* $\bar{\mathrm{v}}(t)$ *ma wartość* $\bar{\mathrm{v}}^i$. *Zatem warunek*

$$\int_{t_k}^{t_{k+1}} \bar{\mathrm{v}}(t)\, dt = \mathrm{x}(t_{k+1}) - \mathrm{x}(t_k)$$

można zapisać w równoważnej postaci:

$$\sum_{i=1}^{l} \bar{\mathrm{v}}^i \Delta s_i = \mathrm{x}(t_{k+1}) - \mathrm{x}(t_k)$$

lub

$$\sum_{i=1}^{l} \bar{\mathrm{v}}^i \frac{\Delta s_i}{t_{k+1} - t_k} = \frac{\mathrm{x}(t_{k+1}) - \mathrm{x}(t_k)}{t_{k+1} - t_k}.$$

Niech $\mathrm{v} = \dfrac{\mathrm{x}(t_{k+1}) - \mathrm{x}(t_k)}{t_{k+1} - t_k}$. *Kładąc* $\lambda_i = (\Delta s_i)/(t_{k+1} - t_k)$ *mamy:*

- $\lambda_i \geqslant 0$, $\quad (i = 1, 2, \dots, l)$
- $\sum_{i=1}^{l} \lambda_i = 1$.

Zatem

$$\sum_{i=1}^{l} \bar{\mathrm{v}}^{\mathrm{i}} \lambda_i = \mathrm{v}.$$

Ponieważ $\mathrm{co}(V_k) = V_k$ *(*$\mathrm{co}(\cdot)$ *oznacza otoczkę wypukłą zbioru), stąd*

$$\sum_{i=1}^{l} \bar{\mathrm{v}}^{\mathrm{i}} \lambda_i \in V_k,$$

a więc także

$$\mathrm{v} = const \in V_k.$$

Rozważmy teraz dowolną ciągłą funkcję $\bar{\mathrm{v}}(t)$*. Na podstawie podstawowego twierdzenia o całce Lebesgue'a każdą taką funkcję* $\bar{\mathrm{v}}(t)$ *można przybliżać z dokładnością* $\varepsilon > 0$ *przez funkcję schodkową. Ten przypadek jednak rozpatrzyliśmy już w pierwszej części dowodu.*

Korzystając z zależności

$$\bar{v}_j(t) = f_j(\bar{u}_j(t)) \quad (j = 1, 2, \ldots, n)$$

możemy wyliczyć stały w czasie rozdział

$$u_j = f_j^{-1}(v_j) \quad (j = 1, 2, \ldots, n).$$

■

4. Algorytm optymalnego rozdziału zasobu

W niniejszym rozdziale skonstruujemy algorytm rozwiązujący problem optymalnego rozdziału zasobu dla ustalonego uszeregowania zadań $z \in Z$. Problem ten jest skomplikowanym problemem optymalizacji dynamicznej. Wykorzystując własności wykazane w poprzednim rozdziale, sprowadzimy go do znacznie łatwiejszego zagadnienia programowania wypukłego.

Niech $x_{jk} = x_j(r_{k+1}) - x_j(r_k)$ oznacza rozmiar częściowy terminu dostępności zadania j, realizowany w przedziale $[r_k, r_{k+1}]$. Ponadto niech u_{jk} oznacza stałą ilość zasobu przydzieloną do terminu dostępności zadania j w przedziale $[r_k, r_{k+1}]$.

Wykorzystując fakt, że w dowolnym przedziale $[r_k, r_{k+1}]$ zachodzi $\int_{r_k}^{r_{k+1}} f_j(u_{jk})\, dt = x_j(r_{k+1}) - x_j(r_k) = x_{jk}$, $j = k+1, \ldots, n$, otrzymujemy zależność:

$$u_{jk} = f_j^{-1}\left(\frac{x_{jk}}{r_{k+1} - r_k}\right), k = 0, 1, \ldots, n-1,\; j = k+1, \ldots, n. \tag{5}$$

Uwzględniając ponadto, że $r_{k+1} - r_k = p_k$, $k = 1, 2, \ldots, n-1$ oraz $r_1 - r_0 = \Delta r_0$, otrzymujemy:

$$\begin{cases} u_{j0} = f_j^{-1}\left(\dfrac{x_{j0}}{\Delta r_0}\right), & j = 1, 2, \ldots, n \\ u_{jk} = f_j^{-1}\left(\dfrac{x_{jk}}{p_k}\right), & k = 1, 2, \ldots, n-1, \ j = k+1, \ldots, n, \\ \sum_{k=0}^{j-1} x_{jk} = \hat{x}_j^r, & j = 1, 2, \ldots, n. \end{cases} \tag{6}$$

Z ogólnych własności problemu wynika, że $\sum_{j=1}^{n} u_{j0} = \hat{R}$ oraz $\sum_{j=k+1}^{n} u_{jk} \leqslant \hat{R}$ dla $k = 1, 2, \ldots, n-1$.

Uwzględniając powyższe zależności w warunku $\sum_{j=1}^{n} u_{j0} = \hat{R}$ otrzymujemy następującą zależność:

$$\sum_{j=1}^{n} f_j^{-1}\left(\frac{\hat{x}_j^r - \sum_{k=1}^{j-1} x_{jk}}{\Delta r_0}\right) = \hat{R}. \tag{7}$$

Załóżmy, że potrafimy wyznaczyć analitycznie Δr_0 w postaci funkcji częściowych rozmiarów terminów dostępności zadań (w niektórych przypadkach jest to bardzo trudne i należy to zrobić numerycznie):

$$\Delta r_0 \;=\; G\left(\{x_{jk}\}_{\substack{k=1,2,\ldots,n-1 \\ j=k+1,\ldots,n}}\right). \tag{8}$$

Biorąc pod uwagę powyższe rozważania, problem optymalnego przydziału zasobu do terminów dostępności zadań dla ustalonego uszeregowania z można sprowadzić do następującego zagadnienia programowania wypukłego:

Zagadnienie programowania wypukłego (PW)

$$\Delta r_0 = G\left(\{x_{jk}\}_{\substack{k=1,2,\ldots,n-1 \\ j=k+1,\ldots,n}}\right) \longrightarrow \min, \tag{9}$$

przy ograniczeniach:

$$\sum_{j=1}^{n} f_j^{-1}\left(\frac{x_{jk}}{p_k}\right) \leqslant \hat{R}, \qquad k = 1, 2, \ldots, n-1, \tag{10}$$

$$\sum_{k=1}^{j-1} x_{jk} = \hat{x}_j^r, \qquad j = 1, 2, \ldots, n, \tag{11}$$

$$x_{jk} \geqslant 0, \ k = 1, 2, \ldots, n-1; \ j = k+1, \ldots, n. \tag{12}$$

Wyznaczony w ten sposób minimalny czas zakończenia wykonywania wszystkich zadań wynosi:

$$C_{\max}^{*} = C_{\max}(\mathbf{z}, \mathbf{u}_{\mathbf{z}}^{*}) = G\left(\{x_{jk}^{*}\}_{\substack{k=1,2,\ldots,n-1 \\ j=k+1,\ldots,n}}\right) + \sum_{j=1}^{n} p_j,$$

gdzie x^*_{jk} są optymalnymi rozmiarami częściowymi realizowanymi w poszczególnych przedziałach. Stąd możemy bardzo łatwo wyznaczyć optymalny rozdział zasobu $\mathbf{u}^*(t)$ w następujący sposób:

$$\begin{cases} u^*_{j0} = f_j^{-1}\left(\dfrac{x^*_{j0}}{\Delta r_0}\right), & (j = 1,2,\ldots,n), \\ u^*_{jk} = f_j^{-1}\left(\dfrac{x^*_{jk}}{p_k}\right), & (k = 1,2,\ldots,n-1;\ j = k+1,\ldots,n). \end{cases} \tag{13}$$

5. Algorytmy aproksymacyjne

Wykorzystując algorytm optymalnego rozdziału zasobu dla ustalonej permutacji wykonywania zadań (podany w poprzednim punkcie) podamy teraz ogólny schemat algorytmu aproksymacyjnego A, który umożliwi wyliczenie przybliżonego rozwiązania problemu $1\left|\prec, \dot{x}^r_j(t) = f_j\left(u_j(t)\right), \sum u_j(t) \leqslant \hat{R}\right| C_{\max}$.

Ogólny schemat algorytmu przybliżonego dla tego problemu ma następującą postać:

Krok 1. *Wybierz dopuszczalne uszeregowanie zadań* $\mathbf{z}^A \in \mathbf{Z}$.

Krok 2. *Wyznacz dla* $\mathbf{z}^A$ *OPTYMALNY rozdział zasobu* $\mathbf{u}^*_{\mathbf{z}^A} \in \mathbf{U}^t$ *za pomocą algorytmu optymalnego rozdziału zasobu.*

Wybór uszeregowania $\mathbf{z}^A \in \mathbf{Z}$ jest problemem dość złożonym i ze względu na ograniczenia nałożone na niniejszą pracę nie będziemy go szczegółowo omawiać. Można jednak udowodnić, że długość uszeregowania dla naszego problemu obliczona za pomocą przedstawionego schematu jest dla niektórych uszeregowań co najwyżej dwa razy większa niż długość uszeregowania optymalnego.

6. Uogólnienia

Istotnym rozszerzeniem modelu dynamicznego terminów dostępności analizowanego w poprzednich punktach jest wprowadzenie części stałej jako składowej terminu dostępności zadania. W bardzo wielu procesach przemysłowych w procesie przygotowawczym, determinującym termin gotowości zadania, można wyróżnić część stałą i część zależną od ilości przydzielonych zasobów. Przyjmujemy, że część stała odpowiada czasowi transportu elementów pomiędzy kolejnymi maszynami (jest ona stała, niezależna od czynników zewnętrznych), a część dynamiczna opisuje tę część procesu przygotowawczego, której tempo wykonywania jest zależne od ilości dostarczonych zasobów.

Formalna definicja tego problemu jest identyczna z definicją problemu wyjściowego podaną w rozdziale 2; zmianie uległa postać równania opisującego termin dostępności. W niniejszym problemie termin dostępności r_j zadania j opisany jest następującym równaniem różniczkowym

$$\begin{aligned} \dot{x}_j^r(t) &= \frac{dx_j^r(t)}{dt} = f_j\left(u_j(t)\right), \quad t \geqslant r_j^{\mathrm{const}}, \\ x_j^r(t) &= 0, \quad 0 \leqslant t < r_j^{\mathrm{const}}. \end{aligned} \tag{14}$$

Parametr r_j^{const} odpowiada stałej części terminu dostępności zadania j.

Łatwo można udowodnić, że niniejszy problem jest NP-trudny. Mimo znacznego podobieństwa do problemu omawianego w poprzedniej części pracy jego pełne rozwiązanie jest znacznie trudniejsze. Autorom udało się opracować algorytm optymalnego rozdziału zasobu dla ustalonej permutacji wykonywania zadań tylko przy założeniu, że terminy dostępności spełniają warunek $\max_j \{r_j^{\mathrm{const}}\} \leqslant r_1$. W takim przypadku można przekształcić problem optymalnego rozdziału zasobu dla ustalonej permutacji $z^A \in Z$ do zagadnienia programowania wypukłego. Ponadto zaproponowano algorytm aproksymacyjny dla tego problemu oraz przeprowadzono jego analizę eksperymentalną.

LITERATURA

1. Burkov W.N.: Sterowanie optymalne w kompleksach operacji, *IFAC*, 5:46–57, 1965.
2. Janiak A., Przysada J.: Minimalizacja nieterminowości wykonania zadań na równoległych maszynach, Zeszyty Naukowe Politechniki Śląskiej, s. Automatyka, z. 117, Gliwice 1996, str. 85–97.
3. Janiak A.: Dokładne i przybliżone algorytmy szeregowania zadań i rozdziału zasobów w dyskretnych procesach przemysłowych, Monografie 87/20, Politechnika Wrocławska, Instytut Cybernetyki Technicznej, 1991.
4. Janiak A.: Wybrane problemy i algorytmy szeregowania zadań i rozdziału zasobów. Problemy współczesnej nauki, Teoria i zastosowania, Informatyka, Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1999.
5. Józefowska J.: Dyskretno-ciągłe problemy szeregowania zadań, Rozprawy nr 318, Wydawnictwo Politechniki Poznańskiej, Poznań, 1997.

6. Węglarz J.: Multiprocessor Scheduling with Memory Allocation - A Deterministic Approach, *IEEE Transactions on Computers*, C-29(8):703–709, August 1980.

Recenzent: Prof. dr inż. T. Puchałka

Abstract

In this paper we present some results obtained in the research of dynamic models of job release dates. The presented analysis covers processes in which the main (proper) part of the processed job may be initiated after a certain preliminary process (pre-processing) has been completed. The change rate of this preliminary process depends dynamically on the amount of the allocated resource and can be a continous, non-negative and strictly increasing function of the resource. We present the solution of the problem of minimizing the time of processing a set of n jobs with dynamical (differential) models of job release dates on a single critical machine. The amount of resource available at each moment is known *a priori*. Many important properties of this problem have been proven. They are the base for construction of optimal resource allocation algorithm for jobs processed in a given permutation. The approximation algorithm for the scheduling problem is presented as well.