ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2006
Seria: AUTOMATYKA z. 144 Nr kol. 1727

Mateusz GORCZYCA, Adam JANIAK
Politechnika Wrocławska

MINIMALIZACJA DŁUGOŚCI USZEREGOWANIA ZADAŃ W JEDNOPROCESOROWYM PROBLEMIE Z DYNAMICZNYMI MODELAMI TERMINÓW ICH DOSTĘPNOŚCI

Streszczenie. W niniejszej pracy rozpatrywany jest problem minimalizacji długości uszeregowania zadań o dynamicznych modelach terminów dostępności na pojedynczym procesorze. Prędkość zmiany stanu terminu gotowości w danej chwili zależy od ilości przydzielonego zasobu. Zasób jest odnawialny, stały w czasie i podzielny w sposób ciągły, a jego ilość ograniczona. Wykazano własność problemu i skonstruowano algorytm optymalnego rozdziału zasobu.

MAKESPAN MINIMIZATION ON A SINGLE PROCESSOR WITH DYNAMIC MODELS OF TASK RELEASE DATES

Summary. In this paper, we consider the problem of makespan minimization on a single processor with dynamic models of task release dates. The speed of change of the release date state in every moment depends on amount of alloted resource. The resource is renewable, constant, continuously divisible and its amount is limited. A property of the problem is proved and based on it algorithm of optimal resource allocation is provided.

1. Wstęp

W przemyśle często mamy do czynienia z sytuacją, gdzie występuje pojedynczy procesor krytyczny (np. bardzo drogi) i zależy nam na optymalnym uszeregowniu na nim zadań dla ustalonego kryterium. Przygotowanie zadania do wykonania na tym procesorze polega na wykonaniu procesu obróbki wstępnej, której czas zakończenia interpretowany jest jako termin dostępności zadania. Szybkość lub czas realizacji tego procesu często zależy od pewnego dodatkowego zasobu podzielnego w sposób ciągły i ograniczonego, np. paliwa, energii, gazu itd. [3].

Dynamiczny model zadania (typu prędkość/zasób) został wprowadzony przez Burkova [1], natomiast dynamiczny model terminu dostępności wprowadził Janiak [3]. Rozpatrywany w niniejszej pracy problem jednoprocesorowy z dynamicznymi terminami dostępności zadań, sformułowany w rozdziale 2, badano dla kryterium długości uszeregowania [4] oraz zużycia zasobu [5] . W rozdziale 3 wykazano nową własność analizowanego problemu, którą następnie w rozdziale 4 wykorzystano do sformułowania algorytmu optymalnego rozdziału zasobu.

2. Sformułowanie problemu

Dany jest zbiór n niepodzielnych zadań $J = \{1, \ldots, j, \ldots, n\}$ do wykonania na jednym procesorze (maszynie krytycznej). Czas wykonywania p_j każdego zadania $j \in J$ jest z góry zadany. Proces przygotowania zadania do wykonania jest dany modelem dynamicznym:

$$\dot{x}_j(t) \triangleq \frac{dx_j(t)}{dt} = f_j(u_j(t)),$$

gdzie $x_j(t)$ oznacza stan procesu zadania j w chwili t, $u_j(t)$ oznacza ilość zasobu przydzieloną w chwili t do procesu zadania j, natomiast $f_j(\cdot)$ to funkcja ciągła, rosnąca, wklęsła i spełniająca warunek $f_j(0) = 0$. Dla każdego zadania $j \in J$ dany jest stan początkowy $x_j(t = 0) = 0$ oraz stan końcowy $\hat{x}_j$, który musi być osiągnięty, aby zadanie było gotowe do wykonania. Zadanie staje się dostępne, gdy jego proces osiągnie swój stan końcowy, zatem termin dostępności $r_j \triangleq \min\{t : x_j(t) = \hat{x}_j\}$.

Zasób jest odnawialny, a jego globalna ilość $\hat{U}$ dostępna do rozdzielenia pomiędzy procesy jest z góry zadana, stała w czasie i ograniczona. Oznaczmy przez $\mathbf{u}(t) \triangleq [u_1(t), \ldots, u_j(t), \ldots, u_n(t)]$ funkcję wektorową nazywaną dalej *rozdziałem zasobu*, która spełnia warunki:

- $u_j(t),\ j = 1, \ldots, n$ jest przedziałami ciągłe,
- $\sum_{j=1}^{n} u_j(t) \leqslant \hat{U}$,
- $r_j,\ j = 1, \ldots, n$ jest skończone (przy czym zachodzi $\int_0^{r_j} x_j(t)dt = \hat{x}_j$),
- $u_j(t) \geqslant 0$ dla $t \in [0, r_j]$,
- $u_j(t) = 0$ dla $t \in (-\infty, 0) \cup (r_j, \infty)$.

Zadania są wykonywane zgodnie z kolejnością π (gdzie π jest permutacją zbioru J), $S_{\pi(j)} = \max\{r_{\pi(j)}, C_{\pi(j-1)}\}$, $j = 1, \ldots, n$, gdzie S_j oraz $C_j = S_j + p_j$ oznaczają czas odpowiednio rozpoczęcia i zakończenia wykonywania zadania j (oraz $C_{\pi(0)} = 0$).

Problem polega na znalezieniu takiej kolejności wykonywania zadań π^* oraz takiego rozdziału zasobu $\mathbf{u}^*(t)$, które zminimalizują długość uszeregowania $C_{\max} = \max\{C_1, \ldots, C_n\}$. Rozpatrywany problem jest silnie NP-trudny [4].

3. Własności problemu

W dalszych rozważaniach wykorzystamy następujący lemat i twierdzenie, udowodnione odpowiednio w [2] i [4]:

Lemat 1. *[2] Dla dowolnych* $0 \leqslant u_1 < u_2$, $x_1 \geqslant 0$, $x_2 > 0$, $T_1 > 0$, $T_2 > 0$ *oraz ciągłej, wklęsłej i rosnącej funkcji* $f(\cdot)$ *spełniającej warunek* $f(0) = 0$, *jeżeli zachodzi* $x_1 = f(u_1) \cdot T_1$ *oraz* $x_2 = f(u_2) \cdot T_2$, *to:*

(a) dla każdej liczby u'_1 *takiej, że* $u_1 < u'_1 \leqslant f^{-1}(K)$, *istnieje liczba* u'_2 *taka, że* $f^{-1}(K) \leqslant u'_2 < u_2$ *oraz*

$$x_1 + x_2 = f(u_1) \cdot T_1 + f(u_2) \cdot T_2 = f(u'_1) \cdot T_1 + f(u'_2) \cdot T_2, \quad (1)$$

$$u_1 \cdot T_1 + u_2 \cdot T_2 \geqslant u'_1 \cdot T_1 + u'_2 \cdot T_2, \quad (2)$$

gdzie $K = \frac{T_1 \cdot f(u_1) + T_2 \cdot f(u_2)}{T_1 + T_2}$;

(b) dla każdej liczby u'_2 *takiej, że* $f^{-1}(K) \leqslant u'_2 < u_2$, *istnieje liczba* u'_1 *taka, że* $u_1 < u'_1 \leqslant f^{-1}(K)$ *oraz spełnione są warunki (1) i (2).*

Twierdzenie 1. *[4] Dla każdego dopuszczalnego rozdziału zasobu powodującego pewną zmianę stanów procesów zadań w przedziale czasu o niezerowej długości istnieje dopuszczalny rozdział zasobu stały w tym przedziale czasu i powodujący taką samą zmianę stanów procesów zadań.*

Z twierdzenia 1 wynika, że istnieje rozwiązanie optymalne, w którym rozdział zasobu jest stały w przedziałach $(r_{\pi^*(j)}, r_{\pi^*(j+1)}]$, $j = 1, \ldots, n-1$ oraz $[0, r_{\pi^*(1)}]$. W celu uproszczenia oznaczeń w dalszej części artykułu będziemy przyjmowali, że π jest naturalną permutacją zbioru J, tj. $\pi(j) = j$, $j = 1, \ldots, n$.

Własność 1. *Dla każdej kolejności wykonywania zadań* π *istnieje optymalny rozdział zasobu spełniający nastęujące warunki:*

i). Rozdział zasobu jest stały w przedziałach $[0, S_1]$, $(S_j, S_{j+1}]$, $j = 1, \ldots, n-1$.

ii). $S_j = C_{j-1}$, $j = 2, \ldots, n$.

iii). $\sum_{j=1}^{n} u_j(t) = \hat{U}$ *dla* $t \in [0, S_1 = r_1]$.

iv). Przydział zasobu $u_j(t)$ *jest niemalejący od momentu* $t = 0$ *do momentu* S_j.

Dowód. Wykażemy najpierw, że dla każdego rozdziału zasobu $\mathbf{u}(t)$ stałego w przedziałach $(r_j, r_{j+1}]$, $j = 1, \ldots, n-1$ oraz $[0, r_1]$ istnieje rozdział zasobu $\mathbf{u}'(t)$ spełniający warunek *(i)* i dający rozwiązanie o niegorszej wartości funkcji kryterialnej. Dla rozdziału zasobu $\mathbf{u}(t)$ zachodzi $S_1 = r_1$ oraz $S_j \geqslant r_j$, $j = 2, \ldots, n$. Jeśli $S_j = r_j$, $j = 1, \ldots, n$, to $\mathbf{u}(t) = \mathbf{u}'(t)$, w przeciwnym przypadku weźmy $k = \min\{l : l \in J, S_l > r_l\}$. Przydział zasobu $\mathbf{u}(t)$ składa się w przedziale $(S_{k-1}, S_k]$ z co najmniej dwóch schodków, które w myśl twierdzenia 1 możemy w tym przedziale zastąpić stałym przydziałem zasobu. Postępując w ten sposób z rozdziałem zasobu $\mathbf{u}(t)$ w przedziałach $(S_j, S_{j+1}]$, $j = k, \ldots, n-1$ dla $k < n$ otrzymamy rozdział zasobu $\mathbf{u}'(t)$.

Wykażemy teraz, że dla każdego rozdziału zasobu $\mathbf{u}(t)$ spełniającego warunek *(i)* istnieje rozdział zasobu $\mathbf{u}'(t)$ spełniający warunki *(i)* oraz *(ii)* i dający rozwiązanie o niegorszej wartości funkcji kryterialnej. Dla rozdziału zasobu $\mathbf{u}(t)$ istnieje co najmniej jedno zadanie $j \in \{2, \ldots, n\}$ takie, że $S_j > C_{j-1}$. Zmodyfikujmy teraz rozwiązanie dosuwając zadania w prawo do zadania n i oznaczając zmienione czasy rozpoczęcia zadań S'_j, $j = 1, \ldots, n$ (przy czym $S'_n = S_n$). Oczywiście, zachodzi $r_j \leqslant S_j \leqslant S'_j$, $j = 1, \ldots, n$. Rozdział zasobu $\mathbf{u}(t)$ jest funkcją schodkową, przy czym w każdym z przedziałów $(S'_j, S'_{j+1}]$, $j = 1, \ldots, n-1$ oraz $[0, S_1]$ występuje co najmniej jeden schodek. W myśl twierdzenia 1 możemy w każdym z tych przedziałów, w których występuje więcej niż jeden schodek, zastąpić te schodki stałym przydziałem zasobu, otrzymując w ten sposób rozdział zasobu $\mathbf{u}'(t)$.

W dalszej części artykułu dla każdego rozdziału zasobu spełniającego warunki *(i)* i *(ii)* przez x_{ji} oraz u_{ji} będziemy oznaczać odpowiednio zmianę stanu procesu zadania oraz stałą ilość zasobu przydzieloną w przedziale $(S_i, S_{i+1}]$ dla $i = 1, \ldots, n-1$ lub $[0, S_1]$ dla $i = 1$ do j-tego zadania, $j = 1, \ldots, n$. Mnożąc równanie $\frac{dx}{dt} = f_j(u_j(t))$ przez dt i całkując obustronnie otrzymujemy $\int_{S_i}^{S_{i+1}} dx = \int_{S_i}^{S_{i+1}} f_j(u_j(t))dt$, czyli $x_{ji} = f_j(u_{ji}) \cdot (S_{i+1} - S_i)$ (podobnie dla przedziału $[0, S_i]$ otrzymamy $x_{j0} = f_j(u_{j0}) \cdot S_0$). Po prostych przekształceniach dostajemy $u_{ji} = f^{-1}\left(\frac{x_{ji}}{S_{i+1}-S_i}\right)$ (oraz podobnie $u_{j0} = f_j^{-1}\left(\frac{x_{j0}}{S_1}\right)$). Wykażemy teraz, że dla każdego rozdziału zasobu $\mathbf{u}(t)$ spełniającego warunki *(i)* i *(ii)* istnieje rozdział zasobu $\mathbf{u}'(t)$ spełniający warunki *(i)-(iii)* i dający rozwiązanie o niegorszej wartości funkcji kryterialnej. Dla rozdziału zasobu $\mathbf{u}(t)$ zachodzi $\sum_{j=1}^{n} u_{j0} = \sum_{j=1}^{n} f_j^{-1}\left(\frac{x_{j0}}{S_1}\right) \leqslant \hat{U}$, przy czym funkcja $\sum_{j=1}^{n} f_j^{-1}(\cdot)$ jest oczywiście rosnąca. Wynika stąd, że wartość S_1 można zastąpić niewiększą wartością S'_1 taką, że $\sum_{j=1}^{n} f_j^{-1}\left(\frac{x_{j0}}{S'_1}\right) = \hat{U}$. Dosuwając w lewo do momentu S'_1 część rozdziału zasobu $\mathbf{u}(t)$ dla $t \in (S_1, S_n]$ oraz zadania $2, \ldots, n$, do momentu $C'_1 = S'_1 + p_1$ zakończenia zadania 1, otrzymujemy rozwiązanie o rozdziale zasobu $\mathbf{u}'(t)$.

Wykażemy teraz, że dla każdego rozdziału zasobu $\mathbf{u}(t)$ spełniającego warunki *(i)-(iii)* i nie spełniającego warunku *iv)* istnieje rozdział zasobu $\mathbf{u}'(t)$ spełniający warunki *(i)-(iv)* i dający rozwiązanie o niegorszej wartości funkcji kryterialnej. Weźmy dwa sąsiednie przedziały czasu T_i oraz T_{i+1}, $i = 1, \ldots, j-1$ rozdziału zasobu $\mathbf{u}(t)$, gdzie $T_i = (S_{i-1}, S_i]$, $i = 2, \ldots, n$ oraz $T_1 = [0, S_1]$. Załóżmy, że k spośród $n-i$ zadań ma w przedziałach czasu T_i oraz T_{i+1} malejący przydział zasobu. Dla uproszczenia niech będą to zadania $i+1, \ldots, i+k$. Niech $k < n-i$ (dla $k = n-i$ należy z dalszego wywodu usunąć równania (4), (6) i (8)). Oznaczmy $K_{ji} = f_j^{-1}\left(\frac{f_j^{-1}(u_{ji}) \cdot |T_i| + f_j^{-1}(u_{ji+1}) \cdot |T_{i+1}|}{|T_i| + |T_{i+1}|}\right)$, $j = i, \ldots, n$, gdzie $|T_i|$ jest długością przedziału T_i. Weźmy dowolny wektor $\mathbf{u}'_i$ spełniający warunki:

$$\begin{aligned} u'_{ji} &= K_{ji},\ j = i+1, \ldots, i+k, & (3)\\ u_{ji} \leqslant u'_{ji} &\leqslant K_{ji},\ j = i+k+1, \ldots, n, & (4)\\ u'_{ii} &= u_{ii}, & (5)\\ \sum_{j=i}^{n} u'_{ji} &= \min\left\{\hat{U},\ u_{ii} + \sum_{j=i+1}^{n} K_{ji}\right\}. & (6) \end{aligned}$$

Korzystając z lematu 1 można łatwo wykazać, że dla takiego wektora $\mathbf{u}'_i$ zawsze istnieje wektor $\mathbf{u}'_{i+1}$ taki, że rozdział zasobu $\mathbf{u}'(t)$ powoduje w przedziałach T_i i T_{i+1} taką samą zmianę stanów zadań jak rozdział zasobu $\mathbf{u}(t)$, przy czym wektor $\mathbf{u}'_{i+1}$ spełnia następujące warunki:

$$\begin{aligned} u'_{j,i+1} &= K_{ji},\ j = i+1, \ldots, i+k, & (7)\\ u_{j,i+1} \geqslant u'_{j,i+1} &\geqslant K_{ji},\ j = i+k+1, \ldots, n, & (8)\\ \sum_{j=i+1}^{n} u'_{j,i+1} &\leqslant \sum_{j=i+1}^{n} u_{j,i+1}. & (9) \end{aligned}$$

Postępując w opisany sposób z wszystkimi sąsiednimi przedziałami zawierającymi zadania o malejącym przydziale zasobu otrzymamy rozdział zasobu $\mathbf{u}'(t)$.

□

4. Algorytm optymalnego rozdziału zasobu

Z własności 1 wynika, że dla danej kolejności π optymalnym rozdziałem zasobu jest rozdział spełniający warunki *(i)*-*(iv)* i minimalizujący wartość S_1. Równanie $\int_0^{r_j} x_j(t)dt = \hat{x}_j$, $j = 1, \ldots, n$ można dla $j = 2, \ldots, n$ zapisać w postaci $\sum_{i=0}^{j-1} x_{ji} = \hat{x}_j$ oraz dla $j = 1$ mamy $x_{10} = \hat{x}_1$. Korzystając z tych równań oraz z zależności $\sum_{j=1}^{n} u_{j0} = \hat{U}$ możemy zapisać:

$$f_1^{-1}\left(\frac{\hat{x}_1}{S_1}\right) + \sum_{j=2}^{n} f_j^{-1}\left(\frac{\hat{x}_j - \sum_{i=1}^{j-1} x_{ji}}{S_1}\right) = \hat{U}.$$

Załóżmy, że potrafimy z powyższego równania wyznaczyć analitycznie S_1 w postaci funkcji częściowych rozmiarów terminów dostępności zadań (w niektórych przypadkach jest to bardzo trudne i należy to zrobić numerycznie): $S_1 = G\left(\{x_{ji}\}_{j=i+1,\ldots,n}^{i=1,\ldots,n-1}\right)$. Problem znalezienia optymalnego rozdziału zasobu dla danej kolejności π sprowadza się do następującego problemu programowania wypukłego:

$$\min\ S_1 = G\left(\{x_{ji}\}_{j=i+1,\ldots,n}^{i=1,\ldots,n-1}\right) \tag{10}$$

$$\text{p.o.}\ \sum_{j=i+1}^{n} f_j^{-1}\left(\frac{x_{ji}}{p_i}\right) \leqslant \hat{U},\ \ i = 1,\ldots,n-1, \tag{11}$$

$$\sum_{i=1}^{j-1} x_{ji} \leqslant \hat{x}_j,\ \ j = 2,\ldots,n, \tag{12}$$

$$\frac{x_{ji}}{p_i} \leqslant \frac{x_{j,i+1}}{p_{i+1}},\ \ i = 1,\ldots,n-2,\ \ j = i+2,\ldots,n, \tag{13}$$

$$x_{ji} \geqslant 0,\ \ i = 1,\ldots,n-1,\ \ j = i+1,\ldots,n. \tag{14}$$

Ograniczenie (11) wynika ze sformułowania problemu oraz z wykorzystania zależności $S_{i+1} - S_i = p_i$, $i = 1,\ldots,n-1$ wynikającej z własności 1. Ograniczenia (12) i (14) wynikają ze sformułowania problemu, natomiast ograniczenie (13) to inaczej zapisany warunek *(iii)*.

Optymalny rozdział zasobu dla danej kolejności π jest wyznaczany przy wykorzystaniu równań $u_{j0}^* = f_j^{-1}\left(\frac{x_{j0}^*}{S_1}\right)$, $j = 1,\ldots,n$ oraz $u_{ji}^* = f_j^{-1}\left(\frac{x_{ji}^*}{p_i}\right)$, $i = 1,\ldots,n-1$, $j = i+1,\ldots,n$ na podstawie zmian stanów procesów x_{ji}^*, $i = 0,\ldots,n-1$, $j = i+1,\ldots,n$, ($x_{j0}^* = \hat{x}_j - \sum_{i=1}^{j-1} x_{ji}^*$, $j = 2,\ldots,n$, $x_{10}^* = \hat{x}_1$), będących rozwiązaniem problemu programowania wypukłego.

5. Podsumowanie

Nowa własność problemu wykazana dla wypukłych funkcji prędkość/zasób, często występujących w praktyce, pozwala znaleźć optymalny rozdział zasobu przez rozwiązanie problemu optymalizacji wypukłej, znacznie prostszego do rozwiązania niż problem optymalizacji dynamicznej. Własność ta pozwala również wprowadzić dodatkowe ograniczenia do omawianego problemu optymalizacji wypukłej. Autorzy są w trakcie badań potwierdzających skuteczność zaproponowanego algorytmu oraz sprawdzania istnienia podobnej własności dla modelu terminów dostępności zadań złożonego z części stałej i dynamicznej.

LITERATURA

1. Burkov V.N.: Optimal project control. W: Prace IV Kongresu IFAC, t.35, Warszawa 1969, p. 46–57.
2. Gorczyca M., Janiak A.: Minimalizacja poziomu zasobu przy ograniczeniu na długość uszeregowania zadań o modelach dynamicznych na procesorach równoległych. Zeszyty Naukowe Politechniki Śląskiej, s. Automatyka, z. 144, Gliwice 2006, s. 143–149.

3. Janiak A.: Minimization of the blooming mill standstills mathematical model - suboptimal algorithms. Mechanika AGH 8, 1989, p. 37–49.
4. Janiak A.: Wybrane problemy i algorytmy szeregowania zadań i rozdziału zasobów. Akademicka Oficyna Wydawnicza PLJ, Warszawa 1999.
5. Janiak A., Słoniński, P.: Minimalizacja sumarycznej ilości zużytego zasobu w problemie szeregowania zadań o dynamicznych modelach terminów dostępności. Zeszyty Naukowe Politechniki Śląskiej, s. Automatyka, z. 129, Gliwice 2000, s. 179–190.

Recenzent: Prof. dr hab. inż. Tadeusz Czachórski

Abstract

The makespan minimization problem on a single processor with dynamic models of task release dates is considered in the paper. For each task a concave function is given, that relates the speed of the change of releasing process state at a time to the amount of continuously divisible resource allotted to this task. The resource is renewable and its amount is given in advance. For each task there is also given its beginning and its final state, which must be achieved in order to release this task. The objective is to find resource allocation over time and a sequence of tasks that minimize the makespan.

A new property of the problem is proved. Along with already proved theorems, this property allows us to construct optimal resource allocation algorithm for a given task sequence. The optimal resource allocation is obtained by solving a convex optimization problem.