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

Adam JANIAK, Maciej LICHTENSTEIN, Piotr SŁONIŃSKI
Politechnika Wrocławska

MINIMALIZACJA SUMY WAŻONYCH CZASÓW ZAKOŃCZENIA WYKONYWANIA ZADAŃ W JEDNOMASZYNOWYM PROBLEMIE SZEREGOWANIA Z CZASAMI PRZEZBROJEŃ ZALEŻNYMI OD ZASOBÓW

Streszczenie. W niniejszej pracy rozpatrywany jest jednomaszynowy problem szeregowania zadań z przezbrojeniami sekwencyjnie niezależnymi i ograniczeniem technologii grupowej. Przyjęto, że czasy przezbrojeń opisane są przez nierosnące, liniowe funkcje zależne od dodatkowego zasobu podzielnego w sposób ciągły lub dyskretny. Jako kryterium optymalności przyjęto minimalizację ważonych czasów zakończenia wykonywania zadań. Wykazano, że problem z zasobem podzielnym w sposób dyskretny jest problemem NP-trudnym. Ponadto, wykazano szereg własności badanego problemu oraz zaproponowano kilka algorytmów przybliżonych, których efektywność zbadano eksperymentalnie.

SINGLE MACHINE SCHEDULING PROBLEM WITH RESOURCE-DEPENDENT SETUP TIMES AND THE TOTAL WEIGHTED COMPLETION TIME CRITERION

Summary. In the paper a single machine scheduling problem with sequence-independent setup times and group technology is considered. The setup times are given as some nonincreasing, linear functions dependent on additional continuously or discretely-divisible resources. The scheduling criterion is the minimization of the total weighted completion time. It is shown that the problem with discretely-divisible resources is NP-hard. Additionally, some specific properties of the problem are proven, and several approximation algorithms are proposed. The efficiency of these algorithms is verified experimentally.

1. Wprowadzenie

Problemy szeregowania zadań z parametrami zależnymi od zasobów (takimi jak czasy wykonywania, terminy dostępności, czy czasy przezbrojeń) były szeroko rozpatrywane w literaturze naukowej w ciągu ostatnich kilku lat. Poza oczywistym znaczeniem teoretycznym, badania nad tymi problemami motywowane są różnymi zastosowaniami praktycznymi, które można znaleźć zarówno w systemach produkcyjnych, jak i systemach komputerowych. W literaturze ostatnich lat można znaleźć wiele rezultatów dla problemów szeregowania z czasami wykonywania zależnymi od zasobów ([2,4]) lub wcześniej z tzw. czasami kontrolowanymi (ang. *controllable processing times*) ([6,9]). Część autorów rozpatrywała również problemy z terminami dostępności oraz czasami przezbrojeń zależnymi od zasobów ([3,5]). W dziedzinie szeregowania zadań z przezbrojeniami można znaleźć wiele prac przeglądowych. Ostatnio opublikowane to [1] i [7].

W niniejszej pracy rozważany jest jednomaszynowy problem szeregowania zadań z czasami przezbrojeń zależnymi od zasobów przy kryterium minimalizacji sumy ważonych czasów zakończenia wykonywania zadań. W rozpatrywanym problemie czas przezbrojenia opisany jest nierosnącą, liniową funkcją zależną od zasobu podzielnego w sposób ciągły lub w sposób dyskretny.

Pozostała część pracy zorganizowana jest następująco. W następnym rozdziale badany problem zostanie precyzyjnie sformułowany. W rozdziale 3 przedstawione są podstawowe własności badanego problemu, które zachodzą dla obu przypadków, tj. dla zasobu podzielnego w sposób ciągły, jak i podzielnego w sposób dyskretny. Rozdziały 4 i 5 poświęcone są odpowiednio problemowi z zasobem podzielnym w sposób ciągły i w sposób dyskretny. Cała praca jest podsumowana w rozdziale 6.

2. Sformułowanie problemu

Dany jest zbiór $J=\{1,\dots,n\}$, n zadań do wykonania na pojedynczej maszynie. Każde zadanie $j\in J$ jest scharakteryzowane przez czas jego wykonywania $p_j \geq 0$, wagę (priorytet) $w_j \geq 0$ i należy do jednej i tylko jednej rodziny $I_f \in F$, gdzie $F=\{I_f \mid f=1,\dots,B\}$ jest zbiorem B rodzin zadań. Z każdą rodziną $I_f \in F$ skojarzone jest przezbrojenie maszyny o czasie trwania s_f, które musi być wykonane przed rozpoczęciem wykonywania zadań z rodziny I_f. Czas przezbrojenia s_f zadany jest następującym modelem:

$$s_f = b_f - a_f u_f,\ u_f \in U_f,$$

gdzie $b_f \geq 0$ jest maksymalnym czasem przezbrojenia, $a_f \geq 0$ współczynnikiem kompresji czasu przezbrojenia (współczynnikiem wykorzystania zasobu), natomiast u_f ilością zasobu przydzielonego do przezbrojenia. Zbiór wszystkich możliwych wartości zasobu jakie mogą być przydzielone do przezbrojenia, jest zadany przez U_f. W przypadku gdy zasób jest podzielny w sposób ciągły, to $U_f = [0;\overline{u}_f]$, gdzie $\overline{u}_f \geq 0$ jest maksymalną ilością zasobu, jaka może być przydzielona do przezbrojenia s_f. Ze względu na to, że czas przezbrojenia nie może być ujemny, dla każdej rodziny $I_f \in F$ musi być spełniona następująca nierówność: $\overline{u}_f \leq \frac{b_f}{a_f}$. W przypadku zasobu podzielnego w sposób dyskretny zbiór U_f zawiera $k_f \geq 1$ wartości zasobu, jakie mogą być przydzielone do przezbrojenia, tj. $U_f = \{\overline{u}_f^1, ..., \overline{u}_f^{k_f}\}$, gdzie $\overline{u}_f^j > \overline{u}_f^{j-1}$ dla $j = 2,...,k_f$. Bez utraty ogólności możemy przyjąć, że $\overline{u}_f^1 = 0$. Podobnie jak w przypadku ciągłym czas przezbrojenia nie może być wielkością ujemną, więc dla każdej rodziny $I_f \in F$ nierówność $\overline{u}_f^{k_f} \leq \frac{b_f}{a_f}$ musi być spełniona. W obu modelach (tj. z zasobem ciągłym i dyskretnym) całkowita ilość zasobu przydzielona do przezbrojeń jest ograniczona, tzn. $\sum_{f=1}^{B} u_f \leq \hat{U}$. W badanym problemie występuje dodatkowo tzw. ograniczenie technologii grupowej (GT - ang. *group technology*), zabraniające dzielenia rodzin zadań na mniejsze grupy (pomiędzy zadaniami należącymi do jednej rodziny nie może być wykonywane zadanie należące do innej rodziny). Problem polega na znalezieniu takiej sekwencji wykonywania zadań oraz takiego przydziału zasobów do przezbrojeń, które minimalizują kryterium sumy ważonych czasów zakończenia wykonywania zadań: $T = \sum_{j=1}^{n} w_j C_j$, gdzie C_j jest czasem zakończenia wykonywania zadania j. W związku z powyższym, pojedyncze rozwiązanie problemu będzie dalej reprezentowane parą $(\pi, \overline{u})$, gdzie $\pi = (\pi(1),...,\pi(n))$ jest permutacją na zbiorze J określającą sekwencję wykonywania zadań, natomiast $\overline{u} = [u_1,...,u_B]$ jest wektorem ilości zasobów przydzielonych poszczególnym przezbrojeniom. W dalszej części pracy problem ten będziemy notować krótko jako:

$$1 | s_f = b_f - a_f u_f, \sum u_f \leq \hat{U}, GT | \sum w_j C_j \tag{1}$$

dla przypadku z zasobem ciągłym lub

$$1 | dscr, s_f = b_f - a_f u_f, \sum u_f \leq \hat{U}, GT | \sum w_j C_j \tag{2}$$

dla przypadku z zasobem dyskretnym.

3. Własności problemu

W tej części pracy zostaną przedstawione własności problemu, które zachodzą zarówno w przypadku zasobu podzielnego w sposób ciągły, jak i w sposób dyskretny.

Własność 1. *W rozwiązaniu optymalnym zadania w obrębie rodzin uszeregowane są według reguły SWPT (ang. shortest weighted processing time first [8]), tzn. według niemalejących wartości* p_j / w_j.

Dowód. Dowód tej własności może być wykonany przez zamianę sąsiednich zadań i ze względu na prostotę zostanie pominięty. ■

Wprowadźmy następującą definicję:

Definicja 1. *Przez zagregowane zadanie będziemy rozumieli zadanie o czasie trwania (zależnym od zasobu)* $P_f = \sum_{j \in I_f} p_j + b_f - a_f u_f = b_f' - a_f u_f$ *oraz wadze* $W_f = \sum_{j \in I_f} w_j$.

Dodatkowo, problemy szeregowania zadań zagregowanych przy kryterium minimalizacji sumy ważonych czasów zakończenia ich wykonywania będziemy oznaczać jako:

$$1 \,|\, P_f = b_f' - a_f u_f, \sum u_f \le \hat{U} \,|\, \sum W_f C_f \tag{3}$$

w przypadku problemu z zasobem podzielnym w sposób ciągły oraz

$$1 \,|\, dscr, P_f = b_f' - a_f u_f, \sum u_f \le \hat{U} \,|\, \sum W_f C_f \tag{4}$$

w przypadku problemu z zasobem podzielnym w sposób dyskretny, gdzie C_f oznacza termin zakończenia wykonywania zagregowanego zadania *f*. W problemach tych zbiór zadań oznaczymy przez $K = \{1, ..., B\}$, a pojedyncze rozwiązanie będzie reprezentowane parą $(\sigma, \bar{u})$, gdzie $\sigma = (\sigma(1), ..., \sigma(B))$ jest permutacją określającą sekwencję wykonywania zagregowanych zadań, natomiast $\bar{u} = [u_1, ..., u_B]$ jest wektorem ilości zasobów przydzielonych poszczególnym zagregowanym zadaniom.

Twierdzenie 1. *Problemy (1) i (2) są równoważne odpowiednio problemom (3) i (4).*

Dowód. Załóżmy, że w problemie (1) (lub (2)) moment rozpoczęcia wykonywania rodziny I_f (moment rozpoczęcia przezbrojenia) jest równy *S*. Bez utraty ogólności możemy założyć, że do rodziny I_f należą zadania od *k* do *l* i że są wykonywane w tej właśnie kolejności.

Wówczas część kryterium, związana z wykonaniem tej rodziny (oznaczona przez T_f), będzie miała następującą postać:

$$T_f = \sum_{j=k}^{l} w_j C_j = (s + s_f + p_k)w_k + (S + s_f + p_k + p_{k+1})w_{k+1} + ... + (S + s_f + p_k + ... + p_l)w_l =$$

$$(S + s_f + \sum_{j=l}^{l} p_j)\sum_{j=k}^{l} w_j - \sum_{i=k}^{l-1} w_i \sum_{j=i+1}^{l} p_j = (S + s_f + \sum_{j \in I_f} p_j)\sum_{j \in I_f} w_j - \sum_{i=k}^{l-1} w_i \sum_{j=i+1}^{l} p_j = (S + P_f)W_f - A.$$

Należy zauważyć, że składnik $(S + P_f)W_f$ jest równoważny wpływowi na kryterium, jakie miałoby rozpoczęcie wykonywania zadania zagregowanego f również w momencie S. Pozostały składnik $A = \sum_{i=k}^{l-1} w_i \sum_{j=i+1}^{l} p_j$ nie zależy od momentu rozpoczęcia wykonywania rodziny, tylko od kolejności wykonywania zadań w obrębie rodziny, a ta, jak wiemy z Własności 1, powinna być ustalona zgodnie z regułą SWPT, więc jest stała. Należy przypomnieć, że w problemach (1) i (2) obowiązuje ograniczenie GT, więc poszczególne rodziny wykonywane są jedna po drugiej i mogą być zastąpione przez zagregowane zadania.

■

Dla problemów (3) i (4) zachodzi następująca własność:

Własność 2. *Przy ustalonym wektorze rozdziału zasobów $\bar{u}$ optymalną permutację uzyskujemy przez uszeregowanie zadań zgodnie z regułą SWPT, tj. wg niemalejących wartości P_f / W_f.*

Dowód. Dowód tej własności, podobnie jak Własności 1, może być wykonany przez zamianę sąsiednich zadań i zostanie pominięty. ■

4. Problem z zasobem podzielnym w sposób ciągły

W tym rozdziale rozpatrywany będzie problem z zasobem podzielnym w sposób ciągły, tj. problem (3). Wykazano następujące własności problemu:

Własność 3. *Przy ustalonej permutacji σ optymalny rozdział zasobów dla problemu (3) uzyskujemy w O(Blog B) krokach za pomocą następującego algorytmu:*

Algorytm R

Krok 0: Inicjalizacja: $K := \{1,..., B\}$, $u_f := 0$ dla $f \in K$.

Krok 1: Znajdź zadanie $\sigma(g) \in K$ takie, że $a_{\sigma(g)} \sum_{i=g}^{B} W_{\sigma(i)} = \max_{f \in K} \left\{ a_{\sigma(f)} \sum_{i=f}^{B} W_{\sigma(i)} \right\}$.

Krok 2: Podstaw $K := K \setminus \{\sigma(g)\}$, $u_{\sigma(g)} := \min\{\hat{U}, \overline{u}_{\sigma(g)}\}$, $\hat{U} = \hat{U} - u_{\sigma(g)}$.

Krok 3: Jeżeli $K = \varnothing$ lub $\hat{U} = 0$, to STOP. W przeciwnym przypadku przejdź do Kroku 1.

Dowód. Optymalność tego algorytmu wynika z tego, że zasób przydzielany jest w pierwszej kolejności tym zadaniom, dzięki którym otrzymamy największy spadek wartości funkcji kryterialnej. Ze względu na to, że funkcja kryterialna jest liniowa, takie postępowanie jest poprawne. ∎

Własność 4. *Istnieje rozwiązanie optymalne problemu (3), w którym co najwyżej jedno zadanie ma przydzieloną pośrednią wartość zasobu ($0 < u_f < \overline{u}_f$). Pozostałe zadania albo mają przydzieloną maksymalną możliwą ilość zasobu ($u_f = \overline{u}_f$), albo nie mają przydzielonego zasobu wcale ($u_f = 0$).*

Dowód. Własność ta jest bezpośrednim wnioskiem z działania algorytmu rozdziału zasobu R. ∎

Na podstawie wyżej przedstawionych własności sformułowano następujące twierdzenie:

Twierdzenie 2. *Problem (3) jest rozwiązywalny w czasie* $O(B^{\hat{U}+1} \log B)$.

Dowód. Na podstawie Własności 2 wiemy, że przy ustalonym rozdziale zasobu optymalną permutację uzyskujemy przez uszeregowanie zadań wg reguły SWPT. Z kolei wektorów rozdziału zasobu spełniających Własność 4 jest $O(B^{\hat{U}})$ (przy założeniu że wszystkie wielkości w problemie są całkowite). Dla każdego takiego wektora otrzymujemy optymalną permutację w czasie $O(B \log B)$ i wyliczamy wartość kryterium. Spośród tak wyliczonych wartości wybieramy najmniejszą. Całą procedurę można zrealizować więc w czasie $O(B^{\hat{U}+1} \log B)$. ∎

Ze względu na to, że Twierdzenie 2 podaje algorytm wykładniczy rozwiązania problemu, zaproponowano trzy przybliżone algorytmy heurystyczne.

4.1. Algorytmy przybliżone

Algorytmy oznaczono odpowiednio H1, H2, H3. Algorytmy te działają wg następującego schematu i mogą być zrealizowane w czasie $O(B \log B)$:

Krok 1: Wyznacz permutację σ zgodnie z pewną regułą.

Krok 2: Dla wyznaczonej permutacji wyznacz optymalny rozdział zasobów zgodnie z algorytmem R.

Algorytmy H1, H2 i H3 różnią się tylko regułą wyznaczania permutacji w Kroku 1. Dla poszczególnych algorytmów reguły te są następujące:

H1: Wyznaczamy permutację σ taką, że $b'_{\sigma(f)} \le b'_{\sigma(f+1)}$ dla $f = 1,\dots,B-1$.

H2: Wyznaczamy permutację σ taką, że $\frac{b'_{\sigma(f)}}{W_{\sigma(f)}} \le \frac{b'_{\sigma(f+1)}}{W_{\sigma(f+1)}}$ dla $f = 1,\dots,B-1$.

H3: Wyznaczamy permutację σ taką, że $W_{\sigma(f)} \le W_{\sigma(f+1)}$ dla $f = 1,\dots,B-1$.

4.2. Analiza eksperymentalna

W celu przebadania jakości rozwiązań dostarczonych przez algorytmy H1-H3 wygenerowano 1100 instancji badanego problemu. Parametry instancji były generowane zgodnie z rozkładem jednostajnym na następujących przedziałach: $b'_f \in [1;100]$, $a_f \in [1;10]$, $W_f \in [1;10]$, $\overline{u}_f \in [0; b'_f / a_f]$, $\hat{U} \in \left[0; \sum_{f \in K} \overline{u}_f\right]$. Wygenerowano dwie grupy instancji: (1) 100 instancji dla B=20 oraz (2) 1000 instancji dla B=100. Wszystkie instancje rozwiązano algorytmami H1-H3, dodatkowo instancje (1) rozwiązano optymalnie zgodnie z Twierdzeniem 2. Jakość rozwiązań dostarczanych przez algorytmy oceniono za pomocą następujących wskaźników:

$\eta_A = \frac{T(A) - OPT}{OPT}$ – dla instancji z grupy (1), gdzie $T(A)$ oznacza rozwiązanie dostarczone przez algorytm $A \in \{H1, H2, H3\}$ oraz OPT oznacza rozwiązanie optymalne,

$\chi_A = \frac{T(A) - T(H2)}{T(H2)}$ – dla instancji z grupy (2), gdzie $T(A)$ oznacza rozwiązanie dostarczone przez algorytm $A \in \{H1, H2, H3\}$ oraz $T(H2)$ oznacza rozwiązanie dostarczone przez algorytm H2.

Uśrednione wyniki eksperymentów przedstawione są w tablicy 1.

Tablica 1
Wyniki eksperymentów obliczeniowych dla algorytmów H1-H3

A	H1	H2	H3
$\overline{\eta}_A[\%]$	27,80	1,23	138,12
$\overline{\chi}_A[\%]$	23,45	0,00	122,22

Wyniki eksperymentów pokazują jednoznaczną przewagę algorytmu H2. Algorytm H3 jest praktycznie bezużyteczny (nigdy nie dostarczył najlepszego rozwiązania).

5. Problem z zasobem podzielnym w sposób dyskretny

W rozdziale tym rozpatrywany będzie problem z zasobem podzielnym w sposób dyskretny, tj. problem (4). Dla tego problemu na wstępie sformułujmy następujące twierdzenie:

Twierdzenie 3. *Problem (4) należy do klasy problemów NP-trudnych.*

Dowód. W celu udowodnienia NP-trudności problemu (4) transformacji wielomianowej dokonano z NP-zupełnego problemu podziału z równymi mocami zbiorów (ang. **ECP** - *Equal Cardinality Partition*). Problem **ECP** zdefiniowany jest następująco:

ECP (Equal Cardinality Partition): Dany jest zbiór $2m$ elementów $N = \{1,...,2m\}$. Dla każdego elementu $i \in N$ określona jest jego wartość $q_i \geq 0$ taka, że $\sum_{i=1}^{2m} q_i = 2Q$. Czy istnieje podzbiór $X \subseteq N$ taki, że $\sum_{i \in X} q_i = Q$ i $|X| = m$?

Decyzyjna wersja problemu (4), oznaczona dalej przez **TC**, zdefiniowana jest następująco:

TC: Dany jest problem $1|dscr, P_j = b_j' - a_j u_j, \sum u_j \leq \hat{U} | \sum W_j C_j$. Czy w tym problemie istnieje uszeregowanie zadań i rozdział zasobów takie, że $T = \sum_{j=1}^{B} W_j C_j \geq Y$, gdzie Y jest pewną zadaną stałą?

Na podstawie instancji problemu **ECP** tworzymy następującą instancję problemu TC:

$B=2m$, $a_j = 1$, $b_j' = W_j = c_j$, $U_j = \{0, c_j\}$, $c_j = A + q_j$ dla $j = 1,...,B$, gdzie A jest pewną dużą stałą (co najmniej większą niż Q^2), $\hat{U} = mA + Q$, $Y = (A^2(m + m^2) + Q(2A + 2mA) + 2Q^2)/2$.

Zauważmy, że w dowolnym rozwiązaniu powyższej instancji problemu **TC** zadania dzielą się na dwa zbiory: K_u - te zadania, którym przydzielono zasób, oraz K_0 - te zadania, którym zasobu nie przydzielono. Zadania ze zbioru K_u mają zerowy czas wykonywania i są wykonywane na początku uszeregowania nie wpływając na wartość kryterium. Dla zadań ze zbioru K_0 zachodzi równość $P_f = W_{f1}$ w związku z czym zadania te wykonywane są w dowolnej kolejności (patrz Własność 2) po zadaniach ze zbioru K_u. Tak więc pojedyncze rozwiązanie może być reprezentowane przez dwa zbiory K_0 i K_u takie, że $K_0 \cup K_u = K$ oraz $K_0 \cap K_u = \emptyset$. Wartość kryterium w dowolnym rozwiązaniu powyższej instancji możemy zapisać następująco:

$$T = \sum_{f \in K_0} W_f C_f = \sum_{f,g \in K_0, f \le g} c_f c_g = \tfrac{1}{2}\left(\sum_{f \in K_0} c_f^2 + \left(\sum_{f \in K_0} c_f \right)^2 \right).$$

Po podstawieniu wartości $c_f = A + q_f$ oraz kilku przekształceniach otrzymujemy:

$$2T = A^2(|K_0| + |K_0|^2) + (2A + 2|K_0|A) \sum_{f \in K_0} q_f + \sum_{f \in K_0} q_f^2 + \left(\sum_{f \in K_0} q_f \right)^2.$$

Aby problem **TC** posiadał rozwiązanie (tzn. aby odpowiedź na postawione pytanie była: TAK), muszą być spełnione następujące nierówności: $T \le Y$ - nierówność wynikająca z wartości kryterium oraz $\sum_{f \in K_0} u_f \le \hat{U}$ - nierówność wynikająca z ograniczenia na ilość zużytego zasobu. Po podstawieniu do tych nierówności wielkości wynikających z transformacji oraz po dokonaniu prostych przekształceń mogą one być równoważnie zapisane jako:

$$A^2(|K_0| + |K_0|^2) + (2A + 2|K_0|A) \sum_{f \in K_0} q_f + \sum_{f \in K_0} q_f^2 + \left(\sum_{f \in K_0} q_f \right)^2 \le A^2(m + m^2) + Q(2A + 2mA) + 2Q^2 \tag{5}$$

$$|K_0|A + \sum_{f \in K_0} q_f \ge mA + Q \tag{6}$$

(ECP⇒TC) Załóżmy, że problem **ECP** ma rozwiązanie, tzn. istnieje $X \subseteq N$ takie, że $\sum_{i \in X} q_i = Q$ i $|X| = m$. Utwórzmy rozwiązanie problemu **TC** w następujący sposób $K_0 = X$, $K_u = N \setminus X$. Wówczas mamy: $|K_0| = m$ oraz $\sum_{f \in K_0} q_f = Q$. Podstawiając te wielkości do nierówności (5) i (6) oraz wykorzystując oszacowanie $\sum_{f\in K_0} q_f^2 \le Q^2$, możemy łatwo stwierdzić, że są one spełnione. Tak więc, jeżeli problem **ECP** ma rozwiązanie, to problem **TC** również ma rozwiązanie.

(**TC⇒ECP**) Załóżmy, że problem **TC** ma rozwiązanie, tzn. że nierówności (5) i (6) są spełnione. Istnieje stała A na tyle duża, że nierówność (5) jest spełniona tylko wówczas, gdy $|K_0| \le m$. Z kolei istnieje stała A, na tyle duża, że nierówność (6) jest spełniona tylko wówczas, gdy $|K_0| \ge m$. Stąd otrzymujemy, że problem **TC** ma rozwiązanie tylko wówczas, gdy $|K_0| = m$. Podstawiając $|K_0| = m$ w nierównościach (5) i (6) otrzymujemy:

$$(2A+2mA)\sum_{f\in K_0} q_f + \sum_{f\in K_0} q_f^2 + \left(\sum_{f\in K_0} q_f\right)^2 \le Q(2A+2mA)+2Q^2 \tag{7}$$

$$mA + \sum_{f\in K_0} q_f \ge mA + Q \tag{8}$$

Nierówność (8) jest spełniona tylko wówczas, gdy $\sum_{f\in K_0} q_f \ge Q$. Ponadto istnieje na tyle duża stała A, że nierówność (9) jest spełniona tylko wówczas, gdy $\sum_{f\in K_0} q_f \le Q$. Ostatecznie otrzymamy, że nierówności te są spełnione tylko wtedy, gdy $\sum_{f\in K_0} q_f = Q$. Ponieważ, jak wcześniej pokazaliśmy, $|K_0| = m$, podstawiając $X = K_0$ otrzymujemy rozwiązanie problemu **ECP**. Do kompletności dowodu pozostaje oszacowanie stałej A. Łatwo sprawdzić, że zgrubne oszacowanie $A = 16Q^4$ wystarczy, aby wszystkie powyższe rozważania były prawdziwe, a podana transformacja wielomianowa. Ostatecznie możemy stwierdzić, że problem **TC** ma rozwiązanie wtedy i tylko wtedy, gdy problem **ECP** ma rozwiązanie, a ponieważ problem **ECP** jest problemem NP-zupełnym, to **TC** również jest problemem NP-zupełnym, a jego wersja optymalizacyjna jest NP-trudna. ■

Dla problemu z zasobem dyskretnym nie udało się niestety znaleźć algorytmu optymalnego rozdziału zasobu przy ustalonej permutacji (istnieje przypuszczenie, że problem rozdziału zasobu jest NP-trudny). Podobnie, dla tego problemu nie zachodzi Własność 4, co go znacznie komplikuje. W związku z tym przy konstrukcji algorytmów przybliżonych zastosowano nieco inne podejście.

5.1. Algorytmy przybliżone

Dla problemu (4) zaproponowano trzy następujące algorytmy oznaczone odpowiednio **D1, D2, D3**. Algorytmy te działają wg następującego schematu i mogą być zrealizowane w czasie $O(B \log B)$:

Krok 1: Wyznacz rozdział zasobów $\bar{u}$ zgodnie z pewną regułą.

Krok 2: Dla wyznaczonego rozdziału zasobu, wyznacz optymalną permutację σ zgodnie z regułą SWPT.

Algorytmy **D1, D2 i D3** różnią się tylko regułą wyznaczania rozdziału zasobu w Kroku 1. Dla poszczególnych algorytmów reguły te są następujące:

D1: Przydziel zasób (w maksymalnej możliwej ilości) do zadania, które ma największy współczynnik W_j. Postępuj tak, aż do wykorzystania zasobu.

D2: Przydziel zasób w (maksymalnej możliwej ilości) do zadania, które ma największy współczynnik a_j. Postępuj tak, aż do wykorzystania zasobu.

D3: Przydziel zasób (maksymalnej możliwej ilości) do zadania, które ma najmniejszy współczynnik B_j / W_j. Postępuj tak, aż do wykorzystania zasobu.

5.2. Analiza eksperymentalna

W celu przebadania jakości rozwiązań dostarczonych przez algorytmy D1-D3 wygenerowano 1100 instancji badanego problemu. Parametry instancji były generowane zgodnie z rozkładem jednostajnym na następujących przedziałach: $b_j' \in [1;100]$, $a_j \in [1;10]$, $W_j \in [1;10]$, $k_j \in [1;10]$, $\bar{u}_j^1 = 0$, $\bar{u}_j^{k_j} \in [0; b_j' / a_j]$, $\bar{u}_j^j \in [0; \bar{u}_j^{k_j}]$ dla $j = 2,\ldots,k_j - 1$, $\hat{U} \in \left[0; \sum_{j \in K} \bar{u}_j^{k_j}\right]$. Wygenerowano dwie grupy instancji: (1) 100 instancji dla B=10 oraz (2) 1000 instancji dla B=100. Wszystkie instancje rozwiązano algorytmami D1-D3, dodatkowo instancje (1) rozwiązano optymalnie poprzez przegląd zupełny. Jakość rozwiązań dostarczanych przez algorytmy oceniono za pomocą następujących wskaźników: $\eta_A = \frac{T(A) - OPT}{OPT}$ – dla instancji z grupy (1), gdzie $T(A)$ oznacza rozwiązanie dostarczone przez algorytm $A \in \{D1, D2, D3\}$ oraz OPT oznacza rozwiązanie optymalne;

$\chi_A = \dfrac{T(A)-T(D2)}{T(D2)}$ – dla instancji z grupy (2), gdzie $T(A)$ oznacza rozwiązanie dostarczone przez algorytm $A \in \{D1, D2, D3\}$ oraz $T(D2)$ oznacza rozwiązanie dostarczone przez algorytm D2.

Uśrednione wyniki eksperymentów przedstawione są w tablicy 2.

Tablica 2

Wyniki eksperymentów obliczeniowych dla algorytmów D1-D3

A	D1	D2	D3
$\overline{\eta}_A[\%]$	156,10	2,37	3,97
$\overline{\chi}_A[\%]$	143,90	3,09	6,38

Wyniki eksperymentów pokazują, że algorytm D1 jest praktycznie bezużyteczny. Różnica między algorytmami D2 i D3 jest nieznaczna, z niewielką przewagą algorytmu D2. Jednak dla znacznej liczby instancji, to właśnie algorytm D3 dostarczał najlepszego rozwiązania. Złożoność obliczeniowa tych algorytmów jest taka sama, a czas działania jest porównywalny, co nie pozwala nam wskazać lepszego z nich. Zarówno algorytm D2, jak i D3 dostarczają rozwiązań o porównywalnej jakości.

6. Podsumowanie

W niniejszej pracy rozpatrywano problem szeregowania zadań z przezbrojeniami sekwencyjnie niezależnymi zadanymi jako nierosnące, liniowe funkcje zależne od dodatkowych zasobów. W rozpatrywanym problemie przyjęto występowanie ograniczenia technologii grupowej. Jako kryterium minimalizacji przyjęto sumę ważonych czasów zakończenia wykonywania. Rozpatrzono dwa przypadki problemu: pierwszy - dla zasobu podzielnego w sposób ciągły oraz drugi - dla zasobu podzielnego w sposób dyskretny. Wykazano szereg specyficznych własności problemu oraz udowodniono, że problem z zasobem podzielnym w sposób dyskretny należy do klasy problemów NP-trudnych. Dla obu przypadków problemu zaproponowano przybliżone algorytmy heurystyczne, których efektywność sprawdzono eksperymentalnie. Pomimo wielu starań nie udało się określić złożoności obliczeniowej problemu z zasobem podzielnym w sposób ciągły. Dalsze badania będą skierowane na badanie wyżej przedstawionego problemu przy innych kryteriach optymalizacji oraz z dodatkowymi parametrami zadań, takimi jak terminy dostępności, linie krytyczne itp.

LITERATURA

1. Allahverdi A., Gupta J.N.D, Aldowaisan T.: A review of scheduling research involving setup considerations. Omega International Journal of Management Science, vol. 27, 1999, pp.219-239.
2. Janiak A.: Scheduling independent one-processor tasks with linear models of release dates under a given maximum schedule length to minimize resource consumption. Journal of System Analysis – Modeling – Simulation, vol. 7, 1990, pp. 885-890.
3. Janiak A.: Single machine scheduling problem with a common deadline and resource dependent release dates. European Journal of Operational Research, vol. 53, 1991, pp. 317-325.
4. Janiak A., Kovalyov M.Y.: Single machine scheduling subject to deadlines and resource dependent processing times. European Journal of Operational Research, vol. 94, 1996, pp.284-291.
5. Cheng T.C.E., Janiak A., Kovalyov M.Y.: Single machine batch scheduling with resource dependent setup and processing times. European Journal of Operational Research, vol. 135, 2001, pp.177-183.
6. Panwalkar S.S., Rajagopalan R.: Single machine sequencing with controllable processing times. European Journal of Operational Research, vol. 59, 1992, pp.298-302.
7. Potts C.N., Kovalyov M.Y.: Scheduling with batching: A review. European Journal of Operational Research, vol. 120, 2000, pp.228-249.
8. Smith W.E.: Various optimizers for single-stage production. Naval Research Logistics Quarterly, vol. 3, 1956, pp. 59-66.
9. Vickson R.G.: Two single-machine sequencing problems involving controllable processing times. AIIE Transactions, vol. 12, 1980, 258-262.

Recenzent: Dr hab. inż. Ewa Dudek-Dyduch, Prof. AGH

Abstract

Scheduling problems with setups consideration have many practical applications, which can be found either in production as well as in computer systems. In such problems jobs are grouped into families, and during job processing, between two jobs which belong to different families a setup is inserted. The setup time may depend either on the family which is processed and the family which will be processed (sequence-dependent setups) or only on the family which will be processed (sequence-independent setups). There also may be considered a Group Technology (GT) restriction which require that job families cannot be split into smaller batches. In this paper we consider a single machine scheduling problem with GT restriction and with sequence-independent setup times given as some linear, nonincreasing functions dependent on resources. The objective is to find a sequence of families and the sequence of jobs within the families, which minimize the total weighted completion time under a given constraint on the total resource consumption. The cases with continuously and discretely divisible resource are considered. We show that the problem with discretly-divisible resource is NP-hard, and we propose an approximation algorithm, which efficiency is verified by some experimental analysis. The computational complexity of the problem with continuously-divisible resource remains an open question, however, we have derived some specific problem properties, which allow us to propose several approximation algorithms. Performed extensive numerical experiments show that the proposed algorithms can be used to solve a wide range of problem instances.