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

Adam JANIAK, Janusz KOZŁOWSKI, Maciej LICHTENSTEIN
Politechnika Wrocławska

MINIMALIZACJA DŁUGOŚCI USZEREGOWANIA W JEDNOMASZYNOWYM PROBLEMIE SZEREGOWANIA Z ZADANYMI TERMINAMI DOSTĘPNOŚCI ZADAŃ ORAZ CZASAMI PRZEZBROJEŃ ZALEŻNYMI OD ZASOBÓW

Streszczenie. W niniejszej pracy rozpatrywany jest jednomaszynowy problem szeregowania z określonymi terminami dostępności zadań oraz czasami przezbrojeń zależnymi od dodatkowych zasobów przy kryterium minimalizacji długości uszeregowania. Badany problem rozpatrywany jest w czterech następujących przypadkach: (1) zasoby podzielne w sposób ciągły i występujące ograniczenie technologii grupowej, (2) zasoby podzielne w sposób dyskretny i występujące ograniczenie technologii grupowej, (3) zasoby podzielne w sposób ciągły i brak ograniczenia technologii grupowej oraz (4) zasoby podzielne w sposób dyskretny i brak ograniczenia technologii grupowej. Wykazano, że przypadki (2), (3) i (4) problemu należą do klasy problemów NP-trudnych. Dla przypadku (1) opracowano algorytm rozdziału zasobów oraz znaleziono szczególny przypadek rozwiązywalny wielomianowo.

MINIMIZING MAKESPAN IN THE SINGLE MACHINE SCHEDULING PROBLEM WITH GIVEN JOB RELEASE DATES AND RESORCE-DEPENDENT SETUP TIMES

Summary. In the paper a single machine scheduling problem with given jobs release dates and resource-dependent setup times is studied. The scheduling objective is the minimization of the makespan. The following four cases of the problem are considered: (1) continuously-divisible resources and group technology restriction, (2) discretely-divisible resources and group technology restriction, (3) continuously-divisible resources and no group technology restriction, and (4) discretely-divisible resources and no group technology restriction. It is proved that cases (2), (3) and (4) are NP-hard. The complexity status of the case (1) remains an open question, however, an algorithm of optimal resource allocation for fixed jobs sequence is developed, and polynomially solvable case is found.

1. Wprowadzenie

W literaturze naukowej z zakresu badań operacyjnych można znaleźć wiele prac poświęconych problemom szeregowania zadań z przezbrojeniami [1,3,7,10]. Tak duże zainteresowanie tymi zagadnieniami motywowane jest licznymi zastosowaniami praktycznymi, które spotyka się zarówno w przemysłowych procesach produkcyjnych, jak i w nowoczesnych systemach komputerowych. Jednocześnie daje się zauważyć duże zainteresowanie problemami szeregowania z parametrami zadań - takimi jak czasy ich wykonywania czy terminy ich dostępności - zależnymi od dodatkowych zasobów [5,6,7,9]. Jest to również spowodowane tym, że modele klasyczne, w których wszystkie parametry zadań są zadanymi stałymi, nie są wystarczające do opisu procesów występujących w praktyce.

W niniejszej pracy rozpatrywany jest problem łączący oba wyżej wymienione modele. Jest to jednomaszynowy problem szeregowania z określonymi terminami dostępności zadań oraz czasami przezbrojeń zależnymi od dodatkowych zasobów podzielnych w sposób ciągły lub dyskretny. Jako kryterium optymalizacji przyjęto minimalizację długości uszeregowania. Badany problem rozpatrywany jest w dwóch przypadkach: z ograniczeniem technologii grupowej – zabraniającym dzielenia rodzin zadań na mniejsze grupy – oraz bez tego ograniczenia.

Pozostała część pracy zorganizowana jest następująco. W następnym rozdziale badany problem jest precyzyjnie sformułowany. W rozdziale 3 przedstawione są własności badanego problemu w sytuacji występowania ograniczenia technologii grupowej. Rozdział 4 poświęcony jest przypadkowi problemu, w którym ograniczenie technologii grupowej nie występuje. Rozdział 5 stanowi podsumowanie pracy.

2. Sformułowanie problemu

Rozważany jednomaszynowy problem szeregowania sformułowany jest następująco. Dany jest zbiór $J = \{1,...,n\}$, n zadań oraz zbiór $F = \{I_f \mid f = 1,...,B\}$, B rodzin zadań. Każde zadanie $j \in J$ określone jest przez czas jego wykonywania $p_j \geq 0$, termin dostępności $r_j \geq 0$ oraz należy do jednej i tylko jednej rodziny $I_f \in F$. Dodatkowo, z każdą rodziną I_f

skojarzony jest sekwencyjnie niezależny czas przezbrojenia s_f, który jest wykonywany przed każdą grupą zadań z rodziny I_f. Czas ten opisany jest następującą liniową funkcją zasobową:

$$s_f = b_f - a_f u_f, u_f \in U_f, \quad (1)$$

gdzie: u_f jest ilością zasobu przydzieloną do przezbrojenia, b_f maksymalnym czasem przezbrojenia, a_f zasobowym współczynnikiem skrócenia czasu przezbrojenia, natomiast U_f jest zbiorem wartości zasobu jakie można przydzielić do przezbrojenia. 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. Ze względu na to, że czas przezbrojenia nie może być ujemny, dla każdej rodziny I_f następująca nierówność musi być spełniona: $\overline{u}_f \leq \frac{b_f}{a_f}$. W przypadku zasobu podzielnego w sposób dyskretny zbiór U_f zawiera k_f 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 $u_f^1 = 0$. Podobnie jak w przypadku ciągłym czas przezbrojenia nie może być wielkością ujemną, więc nierówność $\overline{u}_f^{k_f} \leq \frac{b_f}{a_f}$ musi być spełniona dla każdej rodziny $I_f \in F$. W obu modelach (tj. z zasobem ciągłym i dyskretnym) całkowita ilość zasobu przydzielona do przezbrojeń jest ograniczona, tzn. suma ilości zasobów przydzielonych do przezbrojeń nie może być większa niż zadana stała $\hat{U}$. Ponadto, w problemie może występować tzw. ograniczenie technologii grupowej GT, które zabrania dzielenia rodzin na mniejsze grupy (wszystkie zadania z danej rodziny wykonywane są łącznie). Problem polega na znalezieniu takiej sekwencji wykonywania zadań oraz takiego rozdziału zasobu, które minimalizują kryterium długości uszeregowania: $C_{\max} = \max_{j \in J}\{C_j\}$, gdzie C_j jest terminem zakończenia wykonywania zadania j.

Sekwencja wykonywania zadań w badanym problemie określona jest przez permutację $\sigma = (\sigma(1),...,\sigma(n))$ określoną na zbiorze J. Przy ustalonej permutacji σ, w sekwencji wykonywania zadań, możemy wyróżnić $K \leq n$ grup zadań (ang. *batch*) (grupę stanowią zadania należące do jednej rodziny). Oznaczmy przez $G(k)$ k-tą grupę zadań oraz przez $s(k)$ czas przezbrojenia występujący przed tą grupą. Dodatkowo, przez $u(k)$ oznaczmy ilość zasobu przydzieloną do przezbrojenia $s(k)$. Przy tych oznaczeniach pojedyncze rozwiązanie może być reprezentowane przez parę $(\sigma, \overline{u})$, gdzie $\overline{u} = [u(1),...,u(k)]$ jest wektorem rozdziału zasobu do kolejnych przezbrojeń $s(k)$, spełniającym ograniczenie $\sum_{k=1}^{K} u(k) \le \hat{U}$.

3. Problemy z ograniczeniem technologii grupowej GT

W przypadku występowania ograniczenia GT badane problemy, zgodnie z notacją zaproponowaną w [3], będziemy krótko oznaczać jako:

$$1 \mid r_j, s_f = b_f - a_f u_f, \sum u_f \le \hat{U}, GT \mid C_{\max} \tag{2}$$

w przypadku zasobów podzielnych w sposób ciągły oraz

$$1 \mid dscr, r_j, s_f = b_f - a_f u_f, \sum u_f \le \hat{U}, GT \mid C_{\max} \tag{3}$$

w przypadku zasobów podzielnych w sposób dyskretny.

Ze względu na to, że w przypadku ograniczenia GT w dowolnym rozwiązaniu każde przezbrojenie występuje dokładnie raz, pojedyncze rozwiązanie może być reprezentowane przez parę $(\sigma, \vec{u})$, gdzie $\vec{u} = [u_1, \ldots, u_B]$ jest wektorem rozdziału zasobu do przezbrojeń s_f, $I_f \in F$, spełniających ograniczenie $\sum_{f=1}^{B} u_f \le \hat{U}$. Dla tych przypadków problemów wykazano następujące własności.

Własność 1. *Dla problemów (2) i (3) istnieje rozwiązanie optymalne takie, że zadania wewnątrz rodzin uszeregowane są wg niemalejących wartości r_j.*

Dowód. Dowód powyższej własności można w prosty sposób uzyskać poprzez zamianę sąsiednich zadań i ze względu na prostotę zostanie pominięty. ■

Wprowadźmy następującą definicję.

Definicja 1. *Przez zagregowane zadanie będziemy rozumieć zadanie o czasie wykonywania $P_f = \sum_{j \in I_f} p_j$ i terminie dostępności $R_f = r_k - \sum_{j \in I_f \setminus \{k\}} p_j$, gdzie $k = \arg\max_{j \in I_f} r_j$.*

Dodatkowo, oznaczmy problemy szeregowania zadań zagregowanych przy kryterium minimalizacji długości uszeregowania, w których przezbrojenia występują między każdymi dwoma zagregowanymi zadaniami, w następujący sposób (odpowiednio dla zasobów podzielnych w sposób ciągły i dyskretny):

$$1|R_f, s_f = b_f - a_f u_f, \sum u_f \le \hat{U} | C_{\max} \tag{4}$$

$$1|dscr, R_f, s_f = b_f - a_f u_f, \sum u_f \le \hat{U} | C_{\max} \tag{5}$$

Ponadto, w problemach (4) i (5) zbiór zagregowanych zadań oznaczymy przez $H = \{1,...,B\}$, a przez C_f będziemy oznaczać termin zakończenia wykonywania zagregowanego zadania *f*.

Twierdzenie 1. *Problemy (2) i (3) są równoważne odpowiednio problemom (4) i (5).*

Dowód. Ze względu na to, że w problemie występuje ograniczenie technologii grupowej, a na podstawie Własności 1 wiemy, jak zadania są uszeregowane wewnątrz rodzin, możemy, jak łatwo zauważyć, całe rodziny zagregować do pojedynczych zadań o czasach wykonywania P_f i terminach dostępności R_f i sprowadzić problemy (2) i (3) do problemów szeregowania zagregowanych zadań. ■

W dalszej części tego rozdziału rozpatrywane będą problemy (4) i (5), ze względu na to, że są łatwiejsze w analizie.

Własność 2. *W problemach (4) i (5) przy ustalonym rozdziale zasobu $\bar{u}$ optymalną permutację uzyskujemy szeregując zadania według niemalejących wartości $R_f - s_f$.*

Dowód. W przypadku ustalonych wartości czasów przezbrojenia możemy je zagregować z zadaniami po nich występującymi, w taki sposób, że $R_f^{'} = R_f - s_f$ oraz $P_f^{'} = P_f + s_f$. Otrzymamy wówczas klasyczny problem szeregowania *B* zadań z czasami wykonywania $P_f^{'}$ i terminami dostępności $R_f^{'}$, który jest rozwiązywalny przez uszeregowanie zadań wg niemalejących wartości $R_f^{'}$ [8]. ■

Przypadek z zasobem podzielnym w sposób ciągły

Można łatwo wykazać, że w problemie (4) przy ustalonej permutacji σ optymalny rozdział zasobu realizowany jest przez następujący algorytm w O(*B*log*B*) krokach (w opisie algorytmu zakładamy, że permutacja zadań jest permutacją naturalną, tj. $\sigma(f) = f$ dla $f \in H$):

Algorytm R1

Krok 0: Inicjalizacja: $H := \{1,...,B\}$ $u_f := 0$ dla $f \in H$. Wyznacz czasy zakończenia wykonywania zadań C_f dla $f \in H$.

Krok 1: Znajdź zadanie $g = \max_{f \in H}\{f \mid R_f = C_f - P_f\}$. W przypadku gdy takie zadanie nie istnieje, to podstaw g=0.

Krok 2: Podstaw $H := H \setminus \{1,...,g\}$ oraz znajdź zadanie $h = \arg\max_{f \in H}\{a_f\}$.

Krok 3: Podstaw $u_h = \min\{\hat{U}, \overline{u}_h, \min_{f \in H}\{(C_f - P_f - R_f)/a_f\}$.

Krok 4: Podstaw $\hat{U} := \hat{U} - u_h$. Jeżeli $u_h = \overline{u}_h$, to podstaw $H := H \setminus \{h\}$.

Krok 5: Wyznacz czasy zakończenia wykonywania zadań C_f dla $f \in H$.

Krok 6: Jeżeli $H = \varnothing$ lub $\hat{U} = 0$, to STOP. W przeciwnym wypadku idź do Kroku 1.

Wprowadźmy następującą definicję:

Definicja 2. *W problemie (4) będziemy mówili, że zadanie f dominuje zadanie g, jeżeli* $R_f - b_f \le R_g - b_g - a_g\overline{u}_g$ *(co oznaczać będziemy przez* $f \prec g$*).*

Wykorzystując relację dominacji możemy sformułować następujące twierdzenie:

Twierdzenie 2. *Jeżeli w problemie (4) między każdymi dwoma zadaniami f i g zachodzi relacja dominacji (tj.* $f \prec g$ *lub* $g \prec f$*), to jest on rozwiązywalny w O(BlogB) krokach przy pomocy następującego algorytmu:*

Algorytm A1

Krok 1: Znajdź permutację σ *taką, że* $\sigma(f) \prec \sigma(f+1)$ *dla* $f = 1,...,B-1$.

Krok 2: Rozdziel zasób zgodnie z algorytmem R1.

Dowód. Dowód powyższego twierdzenia sprowadza się do zaobserwowania, że jeżeli $f \prec g$, to niezależnie od ilości przydzielonego zasobu do przezbrojeń zawsze będzie zachodziła nierówność $R_f - s_f \le R_g - s_g$, a na podstawie Własności 2 wiemy, że w takiej sytuacji zadanie *f* powinno być wykonywane przed zadaniem g. ■

Przypadek z zasobem podzielnym w sposób dyskretny

W dalszej części pokażemy, że w przypadku zasobu podzielnego w sposób dyskretny problem rozdziału zasobu przy ustalonej permutacji σ jest NP-trudny, co jednocześnie implikuje NP-trudność problemu (5).

Twierdzenie 3. *Problem optymalnego rozdziału zasobu przy ustalonej permutacji σ w problemie (5) jest NP-trudny.*

Dowód. Aby udowodnić NP-trudność badanego problemu transformacji wielomianowej, dokonano z NP-zupełnego problemu podziału (**PART**).

PART: Dany jest zbiór m elementów $N = \{1,...,m\}$. Dla każdego elementu $i \in N$ określona jest jego wartość q_i taka, że $\sum_{i \in N} q_i = 2Q$. Czy istnieje podzbiór $X \subseteq N$ taki, że $\sum_{i \in X} q_i = Q$?

Decyzyjna wersja problemu rozdziału zasobu (oznaczana dalej przez **RD**) sformułowana jest następująco:

RD: Dany jest problem $1 \mid dscr, R_f, s_f = b_f - a_f u_f, \sum u_f \le \hat{U} \mid C_{\max}$ (z ustaloną permutacją zadań σ). Czy w tym problemie istnieje rozdział zasobów $\vec{u}$ taki, że $C_{\max} \le Y$, gdzie Y jest zadaną stałą?

Na podstawie danych problemu **PART** budujemy następującą instancję problemu **RD**:

$B = m$; $R_f = 0, P_f = 1, a_f = 1, b_f = q_f, U_f = \{0; q_f\}$, dla f=1,...,B; $\hat{U} = Q$; $Y = B + Q$.

Zauważmy, że wartość kryterium $C_{\max}$ w tak zbudowanej instancji nie zależy od kolejności wykonywania zadań, ponieważ wynosi:

$$C_{\max} = \sum_{f=1}^{B} P_f + \sum_{f=1}^{B} s_f. \tag{6}$$

Ponadto, ponieważ zbiory U_f są dwuelementowe, rozdział zasobów możemy jednoznacznie zdefiniować przez dwa zbiory H_0 i H_u takie, że do H_0 należą te przezbrojenia, które nie otrzymały zasobów oraz do H_u należą te przezbrojenia, które zasób otrzymały. Ostatecznie, pojedyncze rozwiązanie problemu **RD** jest reprezentowane przez dwa zbiory H_0 i H_u takie, że $H_0 \cup H_u = H$ i $H_0 \cap H_u = \varnothing$. Aby problem **RD** posiadał rozwiązanie, muszą być spełnione następujące ograniczenia: $C_{\max} \le Y$ – nierówność wynikająca z wartości kryterium, $\sum_{j\in H_u} u_j \le \hat{U}$ – nierówność wynikająca z ograniczenia na ilość zużytego zasobu. Wykorzystując równanie (6) oraz podstawiając dane wynikające z transformacji do powyższych nierówności, otrzymujemy następujące nierówności: $B+\sum_{j\in H_0} q_j \le B+Q$ oraz $\sum_{j\in H_u} q_j \le Q$, które po dalszych przekształceniach mogą być równoważnie zapisane jako:

$$\sum_{j\in H_0} q_j \le Q, \tag{7}$$

$$\sum_{j\in H_0} q_j \ge Q. \tag{8}$$

Na podstawie (7) i (8) wnioskujemy, że problem RD ma rozwiązanie tylko wówczas, gdy $\sum_{j\in H_0} q_j = Q$.

(PART⇒RD) Załóżmy, że problem **PART** ma rozwiązanie, tzn. istnieje $X \subseteq N$ takie, że $\sum_{i\in X} q_i = Q$. Utwórzmy rozwiązanie problemu **TC** w następujący sposób: $H_0 = X$, $H_u = N \setminus X$. Wówczas mamy $\sum_{j\in H_0} q_j = Q$. Tak więc, jeżeli problem **PART** ma rozwiązanie, to problem **RD** również ma rozwiązanie.

(RD⇒PART) Załóżmy, że problem **RD** ma rozwiązanie, tzn. istnieje $H_0 \subseteq H$ taki, że $\sum_{j\in H_0} q_j = Q$. Utwórzmy rozwiązanie problemu **PART** w następujący sposób: $X = H_0$. Wówczas mamy $\sum_{i\in X} q_i = Q$. Tak więc, jeżeli problem **RD** ma rozwiązanie, to problem **PART** również ma rozwiązanie.

Ostatecznie możemy stwierdzić, że problem **RD** ma rozwiązanie wtedy i tylko wtedy, gdy problem **PART** ma rozwiązanie, a ponieważ problem **PART** jest problemem NP-zupełnym, to **RD** również jest problemem NP-zupełnym, a jego wersja optymalizacyjna jest NP-trudna. ■

4. Problemy bez ograniczenia GT

Problemy bez ograniczenia technologii grupowej zgodnie z notacją zaproponowaną w [3] będziemy krótko notować w następujący sposób:

$$1 | r_j, s_j = b_j - a_j u_j, \sum u_j \le \hat{U} | C_{\max} \tag{9}$$

w przypadku zasobów podzielnych w sposób ciągły oraz

$$1 \mid dscr, r_j, s_f = b_f - a_f u_f, \sum u_f \le \hat{U} \mid C_{\max} \qquad (10)$$

w przypadku zasobów podzielnych w sposób dyskretny.

W dalszej części pracy wykażemy, że problemy (9) i (10) należą do klasy problemów NP-trudnych już w przypadku przezbrojeń niezależnych od zasobów. W przypadku gdy czasy przezbrojeń nie zależą od zasobów (są zadanymi stałymi wielkościami), zarówno problem (9), jak i (10) sprowadza się do tego samego problemu, który będziemy krótko notować jako $1 \mid r_j, s_f \mid C_{\max}$.

Twierdzenie 4. *Problem $1 \mid r_j, s_f \mid C_{\max}$ jest NP-trudny.*

Dowód. Aby wykazać NP-trudność problemu $1 \mid r_j, s_f \mid C_{\max}$, transformacji wielomianowej dokonano z NP-trudnego problemu $1 \mid s_f \mid L_{\max}$ [4]. Oznaczmy przez **DL** decyzyjną wersję problemu $1 \mid s_f \mid L_{\max}$ oraz przez **DC** decyzyjną wersję problemu $1 \mid r_j, s_f \mid C_{\max}$. Definicje problemów są następujące:

DL: Dany jest zbiór $J' = \{1,...,n'\}$ n' zadań do wykonania na pojedynczej maszynie. Każde zadanie $j \in J'$ określone jest przez czas jego wykonywania p_j' oraz oczekiwany termin zakończenia jego wykonywania d_j'. Oznaczmy przez $F' = \{I_f' \mid f = 1,...,B'\}$ zbiór B' rodzin zadań. Dodatkowym parametrem f_j' każdego zadania jest przynależność do jednej z rodzin, tzn. jeżeli $f_j' = f$, to zadanie j należy do rodziny I_f'. Z każdą rodziną I_f' skojarzony jest czas przezbrojenia s_f', wykonywany przed każdą grupą zadań z rodziny I_f'. Pytanie: czy istnieje takie uszeregowanie zadań, że maksymalna nieterminowość $L_{\max} = \max_{j \in J'} \{C_j - d_j'\}$ jest nie większa niż $\hat{L}$?

DC: Dany jest zbiór $J = \{1,...,n\}$ n zadań do wykonania na pojedynczej maszynie. Każde zadanie $j \in J$ określone jest przez czas jego wykonywania p_j oraz termin dostępności r_j. Oznaczmy przez $F = \{I_f \mid f = 1,...,B\}$ zbiór B rodzin zadań. Dodatkowym parametrem f_j każdego zadania jest przynależność do jednej z rodzin, tzn. jeżeli $f_j = f$, to zadanie j należy do rodziny I_f. Z każdą rodziną I_f skojarzony jest czas przezbrojenia s_f wykonywany przed każdą grupą zadań z rodziny I_f. Pytanie: czy istnieje takie uszeregowanie zadań, że czas zakończenia wykonywania wszystkich zadań $C_{\max} = \max_{j\in J}\{C_j\}$ jest nie większy niż $\hat{C}$?

Odpowiednia instancja problemu **DC** jest zdefiniowana poprzez dane problemu **DL** w następujący sposób: $n = n'$; $B = B'$; $F = F'$; $s_f = s'_f$, dla $j = 1,...,B$; $p_j = p'_j$, dla $j = 1,...,n$; $r_j = d - d'_j + s_{f_j}$, dla $j = 1,...,n$ oraz $\hat{C} = \hat{L} + d$, gdzie $d = \max_{j\in J'}\{d'_j\}$.

W obu problemach pojedyncze rozwiązanie może być reprezentowane permutacją zadań. Niech $\sigma' = (\sigma'(1),...,\sigma'(n'))$ oznacza dowolne rozwiązanie problemu **DL** oraz niech $\sigma = (\sigma(1),...,\sigma(n))$ takie, że $\sigma(j) = \sigma'(n - j + 1)$, odpowiadające mu rozwiązanie problemu **DC**. Dodatkowo, w każdej permutacji można wyróżnić *K* grup zadań należących do tej samej rodziny. Oznaczmy przez $G'(i)$ ($G(i)$) *i*-tą grupę zadań w permutacji σ' (σ) oraz przez $s'(i)$ ($s(i)$) związane z nią przezbrojenie maszyny. Łatwo zauważyć, że $G'(i) = G(K - i + 1)$ oraz $s'(i) = s(K - i + 1)$, tzn. w *i*-tej grupie w rozwiązaniu σ' wykonywane są te same zadania co w grupie (*K-i*+1)-szej w rozwiązaniu σ (z tym że wykonywane są w odwrotnej kolejności) i poprzedza je taki sam czas przezbrojenia. Bez utraty ogólności możemy założyć, że permutacja σ' jest permutacją naturalną, tj. $\sigma'(j) = j$, dla $j = 1,...,n'$ i takie założenie przyjmiemy w dalszej części dowodu.

Dla każdej pary rozwiązań σ' i σ zachodzi następująca własność:

Własność 3. *Jeżeli w rozwiązaniu σ' problemu **DL** dla zadania j^* zachodzi $L'_{j^*} = L_{\max}$, to w odpowiadającym mu rozwiązaniu σ problemu **DC** wykonywanie zadania j^* rozpoczyna się dokładnie w jego terminie dostępności.*

Dowód. Oznaczmy przez $C'_{\max}$ długość uszeregowania w problemie **DL**. Dodatkowo, oznaczmy przez C'_j czas zakończenia wykonywania zadania *j* w problemie **DL** oraz przez S_j czas jego rozpoczęcia wykonywania w problemie **DC**. Załóżmy ponadto, że zadanie j^* wykonywane jest w grupie *G*(*i*) w problemie **DC**. Wówczas dla tego zadania zachodzi: $L_{\max} = L_{j^*} \geq L_{n'} = C'_{\max} - d_{n'} \geq C'_{\max} - d$, a stąd:

$$C'_{j^*} - d_{j^*} + d \geq C'_{\max}. \tag{11}$$

Zauważając, że: $\sum_{k=1}^{n-j^*} p_{\sigma(k)} = \sum_{k=j^*+1}^{n'} p_k^{'}$ i $\sum_{k=1}^{i} s(k) = \sum_{k=K-i+1}^{K} s^{'}(k) + s(i)$ oraz że $C_{\max}^{'} - C_{j^*}^{'} = \sum_{k=j^*+1}^{n'} p_k^{'} + \sum_{k=K-i+1}^{K} s^{'}(k)$, na podstawie (11) otrzymujemy: $d - d_{j^*} + s(i) \geq \sum_{k=j^*+1}^{n'} p_k^{'} + \sum_{k=K-i+1}^{K} s^{'}(k) + s(i)$ i dalej:

$$r_{j^*} \geq \sum_{k=1}^{n-j^*} p_{\sigma(k)} + \sum_{k=1}^{i} s(k) = C_{\sigma(n-j^*)}. \tag{12}$$

Ponieważ $S_{j^*} = \max\{r_{j^*}, C_{\sigma(n-j^*)}\}$, z (12) wiemy, że zadanie j^* rozpocznie się w swoim terminie dostępności w rozwiązaniu σ problemu **DC**, co kończy dowód Własności 3.■

Oznaczmy przez l pierwsze zadanie w permutacji $\sigma^{'}$, dla którego zachodzi $L_{\max}(\sigma) = L_l$. Zadanie to będzie ostatnim zadaniem w permutacji σ, którego czas rozpoczęcia wykonywania wynosi r_l. Niech w permutacji σ zadanie l będzie wykonywane w i-tej grupie. Z Własności 3 wynika następująca zależność:

$$C_{\max}(\sigma) = r_l + \sum_{k=l}^{n} p_{\sigma(k)} + \sum_{k=i+1}^{K} s(k) = r_l + \sum_{k=1}^{l} p_l^{'} + \sum_{k=1}^{K-i} s^{'}(k) =$$

$$d - d_l^{'} + s(i) + \sum_{k=1}^{l} p_k^{'} + \sum_{k=1}^{K-i} s^{'}(k) = d - d_l^{'} + \sum_{k=1}^{l} p_k^{'} + \sum_{k=1}^{K-i+1} s^{'}(k).$$

Następnie zauważając, że $C_l^{'} = \sum_{k=1}^{l} p_k^{'} + \sum_{k=1}^{K-i+1} s^{'}(k)$, otrzymujemy:

$$C_{\max}(\sigma) = d - d_l^{'} + C_l^{'} = L_l + d = L_{\max}(\sigma^{'}) + d.$$

Ostatecznie z powyższej równości możemy wywnioskować co następuje:

(DL⇒DC) Jeżeli problem **DL** ma rozwiązanie takie, że $L_{\max} \leq \hat{L}$, to problem **DC** ma rozwiązanie takie, że $C_{\max} = \hat{L} + d \leq \hat{C}$.

(DC⇒DL) Jeżeli problem **DC** ma rozwiązanie takie, że $C_{\max} \leq \hat{C}$, to problem **DL** ma rozwiązanie takie, że $L_{\max} = \hat{C} - d \leq \hat{L}$.

Ostatecznie, ponieważ problem **DL** jest NP-zupełny, to możemy stwierdzić, że problem **DC** jest również NP-zupełny, a jego wersja optymalizacyjna jest NP-trudna, co kończy dowód. ■

Dla problemu (9) opracowano algorytm optymalnego rozdziału zasobu przy ustalonej permutacji o złożoności O(n). Działanie tego algorytmu można przedstawić w następujących krokach:

Algorytm R2

Krok 0: Potraktuj kolejne grupy zadań jako rodziny i dokonaj agregacji zgodnie z Definicją 1.

Krok 1: Znajdź optymalny rozdział zasobów algorytmem R1.

5. Podsumowanie

W niniejszej pracy rozważano problem szeregowania z określonymi terminami dostępności zadań oraz czasami przezbrojeń zależnymi od dodatkowego zasobu. Jako kryterium optymalizacji przyjęto minimalizację długości uszeregowania. Badano następujące przypadki problemu: (1) ograniczenie technologii grupowej i zasoby podzielne w sposób ciągły, (2) ograniczenie technologii grupowej i zasoby podzielne w sposób dyskretny, (3) brak ograniczenia technologii grupowej i zasoby podzielne w sposób ciągły oraz (4) brak ograniczenia technologii grupowej i zasoby podzielne w sposób dyskretny.

Dla przypadków (1) i (3) skonstruowano algorytm optymalnego rozdziału zasobu przy ustalonej permutacji i wykazano, że jeżeli w przypadku (1) występuje relacja dominacji między każdymi dwoma zadaniami, to jest on rozwiązywalny wielomianowo w czasie O($n$log n). Wykazano również, że problem optymalnego rozdziału zasobów przy ustalonej sekwencji wykonywania zadań w przypadku problemu (2) jest zagadnieniem NP-trudnym, co jednocześnie implikuje NP-trudność przypadku (2) i (4) problemu. Ponadto wykazano, że problemy (3) i (4) badanego problemu, nawet w przypadku klasycznym - tj. z przezbrojeniami niezależnymi od zasobów - należą do klasy zagadnień NP-trudnych. Podsumowując, możemy stwierdzić, że złożoność obliczeniowa przypadku (1) problemu pozostaje nadal nieokreślona. Skonstruowanie wielomianowego algorytmu jego rozwiązania lub udowodnienie jego NP-trudności powinno być jednym z celów dalszych badań.

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. 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.
3. Graham R.L., Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G.: Optimization and approximation in sequencing and scheduling: a survey. Annals of Discrete Mathematics, vol. 3, 1979, pp.287-326.
4. Bruno J., Downey P.: Complexity of task sequencing with deadline, setup times and changeover costs. SIAM Journal of Comput., vol. 7, 1978, pp. 393-404.
5. 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.
6. 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.
7. 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.
8. Lawler E.L.: Optimal sequencing of a single machine subject to precedence constraints. Management Science, vol. 19, 1974, pp. 544-546.
9. Panwalkar S.S., Rajagopalan R.: Single machine sequencing with controllable processing times. European Journal of Operational Research, vol. 59, 1992, pp.298-302.
10. Potts C.N., Kovalyov M.Y.: Scheduling with batching: A review. European Journal of Operational Research, vol. 120, 2000, pp.228-249.

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

Abstract

Scheduling problems with set-ups have attracted a great attention in the literature over several recent years. This is motivated, beside obvious theoretical significance, by many applications, which can be found in production and computer systems. However, most of authors of the papers have treated set-up times as some fixed given parameters. In many practical situations set-up times may be compressed (i.e. shortened) through the consumption of some additional resources such as energy, financial outlay, catalyzer, etc. This paper deals with a single machine scheduling problem, in which the set-up times are described as linear, nonincreasing functions dependent on some discretely divisible resources. The model of the set-up time consists of two parts. One part is a constant and describes the maximal

requirements of set-up time when there are no resources assigned to it. The second part is variable and decreases the set-up time if some amount of resource is assigned to it. We consider the following problem: the minimization of the makespan value under a given constraint on the total resource consumption, and given job release dates. The following four cases of the problem are considered: (1) continuously-divisible resources and group technology restriction, (2) discretely-divisible resources and group technology restriction, (3) continuously-divisible resources and no group technology restriction, and (4) discretely-divisible resources and no group technology restriction. It is proved that cases (2), (3) and (4) are NP-hard. The complexity status of the case (1) remains an open question, however, an algorithm of optimal resource allocation for fixed jobs sequence is developed, and polynomially solvable case is found.