ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1998
Seria: AUTOMATYKA z. 123 Nr kol. 1389

Krzysztof CHUDZIK, Adama JANIAK
Politechnika Wrocławska

SZEREGOWANIE ZADAŃ Z PRZEZBROJENIAMI PRZY OGRANICZENIACH ZASOBOWYCH*

Streszczenie. Praca poświęcona jest jednomaszynowym problemom szeregowania zadań z przezbrojeniami. Zaprezentowano złożoność obliczeniową i przegląd rezultatów dla klasycznych problemów. Wprowadzono nowe modele, gdzie czas przezbrojenia jest funkcją przydzielonych zasobów. Wykazano ich złożoność obliczeniową.

TASK SCHEDULING WITH SETUPS AND RESOURCE CONSTRAINTS

Summary. The paper deals with the single machine scheduling problems with setups. Complexity and results for the classic problems are presented. New models with setups that are some functions of the alloted resources are introduced. Their computational complexity is considered.

1. Wprowadzenie

Przedmiotem niniejszego artykułu jest grupowanie (ang. *batching*) i szeregowanie zadań na pojedynczej maszynie z przezbrojeniami. Tematyka ta w ogólnym nurcie szeregowania zadań na maszynach [7] zajmuje ważne miejsce. Literatura z tej dziedziny jest niezwykle obszerna i ciągle prowadzone są badania owocujące dalszymi publikacjami [3, 9].

Problemy, którymi zajmiemy się, pojawiają się, gdy wykonywane są grupy zadań (ang. *batches*) pochodzące z większych zbiorów określanych mianem rodzin (ang. *families*). Rodzinę cechuje podobieństwo zadań pod względem wymagań produkcyjnych lub wymagania te są wręcz identyczne. W obrębie jednej grupy, której zadania pochodzą z jednej i tej samej rodziny, dla wykonania następnego zadania nie jest wymagana zmiana w systemie produkcyjnym, tzn. nie następuje przeorganizowanie produkcji i/lub wymiana narzędzi, czy jakiekolwiek inne istotne zmiany w systemie produkcji. Gdy zamierzamy wykonywać kolejną grupę zadań, przed rozpoczęciem jej wykonywania niezbędny może być czas przygotowawczy

* Praca finansowana ze środków Grant KBN No. 8 T11A 004 13 oraz Grant KBN No. 8 T11F 001 11

(*setup time*) potrzebny na przygotowanie maszyn. Oczywiście ponoszony jest przy tym pewien koszt przygotowania (*setup cost*), który może być również uwzględniany w funkcji kryterialnej.

Artykuł ten rozpoczniemy od przedstawienia ogólnego modelu matematycznego problemów rozpatrywanych w pracy (rozdz. 2). Następnie przedstawimy wyniki, znajdujące się w literaturze przedmiotu, dotyczące klasycznych zagadnień grupowania i szeregowania na pojedynczej maszynie (rozdz.3). W rozdziale 4 wprowadzone zostanie uogólnienie klasycznych modeli na modele, w których czas przezbrojenia jest funkcją ilości przydzielonych dodatkowych zasobów. W literaturze można znaleźć przykłady publikacji, gdzie rozważane są modele z czasami wykonania zadań, czy terminami dostępności zależnymi od dodatkowych zasobów [4, 5, 6] uwzględniające nawet przezbrojenia [2]. Modele z czasami przezbrojenia zależnymi od dodatkowych zasobów są nowymi modelami i podjęte są po raz pierwszy przez autorów w niniejszej publikacji. Badanie takich modeli motywowane jest przez ich praktyczne występowanie i wykorzystanie. Pojawiają się one, gdy czas przezbrojenia zależny jest od zasobów, np.: energii, mocy przerobowej, ilości różnego rodzaju czynników chemicznych czy wprost nakładów finansowych. W rozdziale 4 przedstawione będą własności problemów opartych na tych modelach, w szczególności złożoność obliczeniowa. Rozdział 5 zawiera wnioski płynące z pracy.

2. Sformułowanie ogólnego modelu problemów optymalizacyjnych

Zakładamy, że szeregowanie zadań odbywa się na pojedynczej maszynie.

Dany jest zbiór N zadań $J = \{ j : j = 1, 2, \ldots, N\}$. Maszyna może wykonywać w danej chwili czasu co najwyżej jedno zadanie i przerywanie wykonywania zadania rozpoczętego nie jest dozwolone. Każde z zadań może mieć określone następujące parametry: czas wykonywania p_j, termin dostępności r_j, pożądany termin zakończenia wykonywania zadania d_j, linię krytyczną (ang. *dead line*) $\underline{d}_j$, czas dostarczenia q_j, wagę w_j itd.

Dany jest zbiór B rodzin zadań $F = \{ I_f: f = 1, 2, \ldots, B\}$. Każde zadanie $j \in J$ należy dokładnie do jednej rodziny $I_f \in F$. Przynależność zadania do danej rodziny jest jego dodatkowym atrybutem. W procesie produkcyjnym pomiędzy kolejnymi zadaniami mogą wystąpić czasy przygotowawcze, tzw. przezbrojenia (ang. *setups*). Jeżeli zadania i oraz j należą do dwóch różnych rodzin: $i \in I_f$ oraz $j \in I_g$, gdzie $f \neq g$, to pomiędzy zadaniami wymagany jest czas przezbrojenia s_{fg}. Jeżeli zadania należą do tej samej rodziny, to przezbrojenie nie występuje, tzn. gdy $f = g$, to $s_{fg} = 0$. Jeżeli zadanie $j \in I_g$ wykonywane jest na maszynie jako pierwsze w sekwencji zadań, to przed nim występuje przezbrojenie s_{0g}.

Jeżeli dla każdej rodziny I_f i I_g $(f \neq g)$ czas przygotowawczy s_{fg} jest niezależny od pierwszej z nich, tzn. $s_{0g}=s_{fg}=s_g$, to mówimy, że czasy przygotowawcze są sekwencyjnie niezależne (*sequence independent*) i oznaczane są s_f. W przeciwnym wypadku mówimy, że czasy są sekwencyjnie zależne (*sequence dependent*).

Istnieją modele, gdzie odpowiednie czasy przygotowawcze można zastąpić przez odpowiednie koszty przygotowawcze (*setup costs*) : c_{fg}, c_f odpowiadają s_{fg}, s_f. Czasy i koszty można rozpatrywać też łącznie.

Rozpatrując sytuacje praktyczne, czyni się dodatkowe założenie, że czasy przygotowawcze (przezbrojenia) spełniają nierówność trójkąta, tzn. $s_{fh} \leq s_{fg} + s_{gh}$.

Istnieją pewne odmiany zagadnienia szeregowania z przezbrojeniami, wynikające z przyjętych założeń: 1) Grupowanie zadań (*batching*) w grupy (*batches*) - grupa (partia produkcyjna) składa się z zadań należących do tej samej rodziny. 2) Szeregowanie rodzin - rodziny stanowią niepodzielną całość, można zmieniać kolejność zadań w obrębie rodziny. W literaturze podaje się dla określenia tego zagadnienia, że istnieje wymóg technologii grupowej (*group technology*). Oznaczać będziemy ten fakt przez GT w notacji problemów.

Głównym zadaniem optymalizacyjnym w problemach z przezbrojeniami maszyn jest takie zaplanowanie produkcji (uszeregowanie zadań), aby minimalizować łączny czas i/lub koszt przygotowania maszyn do produkcji (uwzględniając wymagania na terminy wykonania określonej produkcji). Przy konstruowaniu kryteriów koszty przezbrajania występują jawnie i są składnikiem funkcji kryterialnej. Czasy przezbrojeń najczęściej występują niejawnie, tzn. nie są wprost składnikiem funkcji kryterialnej, ale wpływają na terminy zakończenia wykonywania zadań. Ich istnienie zaznacza się w opisie modelu problemu. Jako funkcje kryterialne występują typowe funkcje znane z ogólnej teorii szeregowania: C_{max}, L_{max}, ΣC_j, $\Sigma w_j C_j$, ΣT_j, $\Sigma w_j T_j$, ΣU_j, $\Sigma w_j U_j$, do których dodaje się sumę kosztów w wybranych modelach.

3. Jednomaszynowe problemy szeregowania zadań z przezbrojeniami niezależnymi od dodatkowych zasobów – złożoność obliczeniowa

W tabelach 1 i 2 zebrano informacje o złożoności obliczeniowej jednomaszynowych problemów szeregowania zadań z przezbrojeniami.

Dla problemów otwartych podano najnowsze źródło (o ile takie znaleziono), gdzie stwierdzano otwartość problemu.

Jeżeli NP-trudność problemu bardziej złożonego wynika z NP-trudności prostszego, to podano ten fakt, np. NP-trudność $1 \mid GT \mid C_{max} + \Sigma c_{fg}$ wynika z NP-trudności $1 \mid GT \mid \Sigma c_{fg}$.

Z pracy [7] wynika, że z NP-trudności dla kryterium C_{max} wynika NP-trudność dla L_{max}, natomiast z NP-trudności dla kryterium L_{max} wynika NP-trudność dla ΣU_j. Własność ta pozwala szybko stwierdzić złożoność wielu problemów, i podajemy ją, jeśli zachodzi.

Problemy równoważne problemowi komiwojażera (ang. *Travelling Salesman Problem*) oznaczać będziemy – równoważny TSP.

Należy dodać, że problemy z kryterium ΣT_j są co najmniej NP-trudne, ponieważ problem $1 \mid\mid \Sigma T_j$ jest NP-trudny [7].

4. Jednomaszynowe problemy szeregowania zadań z przezbrojeniami zależnymi od dodatkowych zasobów

W rozdziale tym rozszerzymy ogólny model prezentowany w rozdz. 2. Wprowadzimy uogólnienie dotyczące zasobów. Przyjmiemy, że czasy przezbrojeń są liniową nierosnącą funkcją przydzielanych w sposób ciągły zasobów nieodnawialnych. Skupimy się na przypadkach, które mogą być rozwiązane optymalnie w czasie wielomianowym. Z rezultatów zebranych w tabelach 1 i 2 wynika, że problemy z przezbrojeniami sekwencyjnie zależnymi są NP-trudne lub otwarte, dlatego rozważać będziemy modele z przezbrojeniami sekwencyjnie niezależnymi.

Będziemy oznaczać: π – permutacja zadań, $\pi(i)$ – zadanie z pozycji i w permutacji π, Π – permutacja rodzin (w problemach, w których przyjęto założenie o stosowaniu technologii grupowej – GT), $\Pi(i)$ – rodzina z pozycji i w permutacji Π, $\delta(j)$ – pozycja zadania lub rodziny j w permutacji zadań lub rodzin, C_j, $C_{\Pi(i)}$ – czasy zakończenia wykonywania odpowiednio zadania j, rodziny $\Pi(i)$.

Oznaczmy: s_f – czas przezbrojenia dla rodziny I_f.

W problemach, w których nie obowiązuje założenie o technologii grupowej – GT, przyjmować będziemy, że przezbrojenie s_f związane jest z zadaniem, które poprzedza. Wobec tego: $s(j)$ - przezbrojenie występujące przed zadaniem j:

$s(j) = s_f$, gdy $j \in f \wedge \pi(\delta(j) - 1) \notin I_f$, $s(j) = \varnothing$ w przeciwnym wypadku.

gdzie $\varnothing$ oznacza brak przezbrojenia przed zadaniem.

Tabela 1

Złożoność obliczeniowa jednomaszynowych problemów szeregowania zadań z przezbrojeniami niezależnymi od dodatkowych zasobów – istnieje wymóg technologii grupowej (GT)

Lp.	Problem	Złożoność	Źródło – Komentarz
1	$1 \mid s_f, \mathrm{GT} \mid C_{\max}$	$O(N)$	Monma & Potts (1989) [8]
2	$1 \mid s_f, \mathrm{GT} \mid L_{\max}$	$O(N \log N)$	Potts & Van Wassenhove (1992) [9]
3	$1 \mid s_f, \mathrm{GT} \mid \Sigma C_i$	$O(N \log N)$	Potts & Van Wassenhove (1992) [9]
4	$1 \mid s_f, \mathrm{GT} \mid \Sigma w_i C_i$	$O(N \log N)$	Potts & Van Wassenhove (1992) [9]
5	$1 \mid s_f, \mathrm{GT} \mid \Sigma U_i$	Otwarty	Potts & Van Wassenhove (1992) [9]
6	$1 \mid s_f, \mathrm{GT} \mid \Sigma w_i U_i$	NP-trudny	$1 \mid\mid \Sigma w_i U_i$ – NP-trudny [7]
7	$1 \mid \mathrm{GT} \mid \Sigma c_f$	$O(N)$	Permutacja dowolna – $O(N)$ to koszt naliczenia kryterium
8	$1 \mid \mathrm{GT} \mid C_{\max} + \Sigma c_f$	$O(N)$	Permutacja dowolna – $O(N)$ to koszt naliczenia kryterium
9	$1 \mid \mathrm{GT} \mid L_{\max} + \Sigma c_f$	$O(N \log N)$	Potts & Van Wassenhove (1992) [9]
10	$1 \mid \mathrm{GT} \mid \Sigma C_i + \Sigma c_f$	$O(N \log N)$	Potts & Van Wassenhove (1992) [9]
11	$1 \mid \mathrm{GT} \mid \Sigma w_i C_i + \Sigma c_f$	$O(N \log N)$	Potts & Van Wassenhove (1992) [9]
12	$1 \mid \mathrm{GT} \mid \Sigma U_i + \Sigma c_f$	Otwarty	Potts & Van Wassenhove (1992) [9]
13	$1 \mid \mathrm{GT} \mid \Sigma w_i U_i + \Sigma c_f$	NP-trudny	$1 \mid\mid \Sigma w_i U_i$ – NP-trudny [7]
14	$1 \mid s_{fg}, \mathrm{GT} \mid C_{\max}$	NP-trudny	Monma & Potts (1989) [8] – Równoważny TSP
15	$1 \mid s_{fg}, \mathrm{GT} \mid L_{\max}$	NP-trudny	Zdrzałka (1992) [11]– Wynika z $1 \mid s_{fg}, \mathrm{GT} \mid C_{\max}$
16	$1 \mid s_{fg}, \mathrm{GT} \mid \Sigma C_i$	Otwarty	Zdrzałka (1992) [11]
17	$1 \mid s_{fg}, \mathrm{GT} \mid \Sigma w_i C_i$	Otwarty	Nie znaleziono rezultatów w literaturze
18	$1 \mid s_{fg}, \mathrm{GT} \mid \Sigma U_i$	NP-trudny	Wynika z $1 \mid s_{fg}, \mathrm{GT} \mid L_{\max}$
19	$1 \mid s_{fg}, \mathrm{GT} \mid \Sigma w_i U_i$	NP-trudny	$1 \mid\mid \Sigma w_i U_i$ – NP-trudny [7]
20	$1 \mid \mathrm{GT} \mid \Sigma c_{fg}$	NP-trudny	Zdrzałka (1992) [11] – Równoważny TSP
21	$1 \mid \mathrm{GT} \mid C_{\max} + \Sigma c_{fg}$	NP-trudny	Wynika z $1 \mid \mathrm{GT} \mid \Sigma c_{fg}$
22	$1 \mid \mathrm{GT} \mid L_{\max} + \Sigma c_{fg}$	NP-trudny	Wynika z $1 \mid \mathrm{GT} \mid \Sigma c_{fg}$
23	$1 \mid \mathrm{GT} \mid \Sigma C_i + \Sigma c_{fg}$	NP-trudny	Wynika z $1 \mid \mathrm{GT} \mid \Sigma c_{fg}$
24	$1 \mid \mathrm{GT} \mid \Sigma w_i C_i + \Sigma c_{fg}$	NP-trudny	Zdrzałka (1992) [11] – Wynika z $1 \mid \mathrm{GT} \mid \Sigma c_{fg}$
25	$1 \mid \mathrm{GT} \mid \Sigma U_i + \Sigma c_{fg}$	NP-trudny	Wynika z $1 \mid \mathrm{GT} \mid \Sigma c_{fg}$
26	$1 \mid \mathrm{GT} \mid \Sigma w_i U_i + \Sigma c_{fg}$	NP-trudny	$1 \mid\mid \Sigma w_i U_i$ – NP-trudny [7]

Czas przezbrojenia s_f dany jest następującą funkcją:

$$s_f = b_f - a_f u_f \wedge 0 \le u_f \le \beta_f \wedge f = 1, \ldots, B.$$

Parametr u_f jest ilością nieodnawialnego zasobu przydzielonego do przezbrojenia, natomiast wartość β_f wynika z ograniczeń technologicznych.

Rozdział zasobów dla całej permutacji oznaczymy jako wektor $u = [u_1, \ldots, u_j, \ldots, u_N]$ (oczywiście $u_j = 0$ jeśli $s(j) = \varnothing$) lub $u = [u_1, \ldots, u_B]$ przy założeniu GT.

Tabela 2

Złożoność obliczeniowa jednomaszynowych problemów szeregowania zadań z przezbrojeniami niezależnymi od dodatkowych zasobów – nie istnieje wymóg technologii grupowej

Lp.	Problem	Złożoność	Źródło – Komentarz
1	$1 \mid s_f \mid C_{max}$	O(N)	Monma & Potts (1989) [8]
2	$1 \mid s_f \mid L_{max}$	NP-trudny	Bruno & Downey (1978) [1]
3	$1 \mid s_f \mid \Sigma C_j$	Otwarty	Potts & Van Wassenhove (1992) [9]
4	$1 \mid s_f \mid \Sigma w_j C_j$	Otwarty	Webster & Baker (1995) [10]
5	$1 \mid s_f \mid \Sigma U_j$	NP-trudny	Wynika z $1 \mid s_f \mid L_{max}$
6	$1 \mid s_f \mid \Sigma w_j U_j$	NP-trudny	$1 \mid\mid \Sigma w_j U_j$ – NP-trudny [7]
7	$1 \mid\mid \Sigma c_f$	O(N)	Permutacja dowolna – O(N) to koszt naliczenia kryterium
8	$1 \mid\mid C_{max} + \Sigma c_f$	O(N)	Permutacja dowolna – O(N) to koszt naliczenia kryterium
9	$1 \mid\mid L_{max} + \Sigma c_f$	NP-trudny	Bruno & Downey (1978) [1]
10	$1 \mid\mid \Sigma C_j + \Sigma c_f$	Otwarty	Potts & Van Wassenhove (1992) [9]
11	$1 \mid\mid \Sigma w_j C_j + \Sigma c_f$	Otwarty	Potts & Van Wassenhove (1992) [9]
12	$1 \mid\mid \Sigma U_j + \Sigma c_f$	NP-trudny	Wynika z $1 \mid\mid L_{max} + \Sigma c_f$
13	$1 \mid\mid \Sigma w_j U_j + \Sigma c_f$	NP-trudny	$1 \mid\mid \Sigma w_j U_j$ – NP-trudny [L]
14	$1 \mid s_{fg} \mid C_{max}$	NP-trudny	Monma & Potts (1989) [8] – Równoważny TSP
15	$1 \mid s_{fg} \mid L_{max}$	NP-trudny	Zdrzałka (1992) [11] – Wynika z $1 \mid s_{fg} \mid C_{max}$
16	$1 \mid s_{fg} \mid \Sigma C_j$	Otwarty	Potts & Van Wassenhove (1992) [9]
17	$1 \mid s_{fg} \mid \Sigma w_j C_j$	Otwarty	Potts & Van Wassenhove (1992) [9]
18	$1 \mid s_{fg} \mid \Sigma U_j$	NP-trudny	Wynika z $1 \mid s_{fg} \mid L_{max}$
19	$1 \mid s_{fg} \mid \Sigma w_j U_j$	NP-trudny	$1 \mid\mid \Sigma w_j U_j$ – NP-trudny [7]
20	$1 \mid\mid \Sigma c_{fg}$	NP-trudny	Zdrzałka (1992) [11]– Równoważny TSP
21	$1 \mid\mid C_{max} + \Sigma c_{fg}$	NP-trudny	Wynika z $1 \mid\mid \Sigma c_{fg}$
22	$1 \mid\mid L_{max} + \Sigma c_{fg}$	NP-trudny	Wynika z $1 \mid\mid \Sigma c_{fg}$
23	$1 \mid\mid \Sigma C_j + \Sigma c_{fg}$	NP-trudny	Wynika z $1 \mid\mid \Sigma c_{fg}$
24	$1 \mid\mid \Sigma w_j C_j + \Sigma c_{fg}$	NP-trudny	Zdrzałka (1992) [11] – Wynika z $1 \mid\mid \Sigma c_{fg}$
25	$1 \mid\mid \Sigma U_j + \Sigma c_{fg}$	NP-trudny	Wynika z $1 \mid\mid \Sigma c_{fg}$
26	$1 \mid\mid \Sigma w_j U_j + \Sigma c_{fg}$	NP-trudny	$1 \mid\mid \Sigma w_j U_j$ – NP-trudny [7]

Dodatkowo może wystąpić ograniczenie na globalną ilość zasobów przydzielonych do wszystkich występujących w harmonogramie przezbrojeń:

$\Sigma u_j \leq R$ dla $j \in J \wedge s(j) \neq \varnothing$

lub

$\Sigma u_f \leq R$ dla f takich, że $I_f \in F$ przy założeniu GT.

Kryterium postaci $K \wedge \Sigma u_j$, gdzie K klasyczne kryterium, np. C_{max}, L_{max}, itd., oznacza, że poszukiwane jest rozwiązanie Pareto optymalne, tzn. zbiór punktów kompromisowych (K^P, ρ^P) pomiędzy przeciwstawnymi wartościami kryteriów K i $\rho = \Sigma u_j$ wraz z odpowiadającymi każdemu punktowi optymalną permutacją π^P oraz optymalnym rozdziałem zasobów u^P.

4.1. Szeregowanie zadań z przezbrojeniami zależnymi od dodatkowych zasobów i kryterium C_{max}

Twierdzenie 1: Problem $1 \mid s_f = b_f - a_f u_f,\ \Sigma u_j \leq R \mid C_{max}$ jest rozwiązywalny optymalnie przez następujący algorytm w czasie $O(B \log B)$:

Algorytm 1:

0. Pogrupuj zadania rodzinami i wybierz dowolną permutację rodzin Π. Podstaw $F := \{I_f: I_f = \Pi(1), \ldots, \Pi(B)\}$, $u_f := 0$ dla $I_f \in F$.
1. Znajdź rodzinę $I_f \in F$, dla której $a_f = \max_{g \in F} \{a_g\}$, a następnie podstaw $u_f := \min\{\beta_f, R\}$, $R := R - u_f$, $F := F - \{I_f\}$.
2. Jeśli $F \neq \varnothing$ i $R > 0$ wróć do kroku 1, w przeciwnym wypadku stop – uzyskano optymalny rozdział zasobów.

Twierdzenie 2: Problem $1 \mid s_f = b_f - a_f u_f,\ C_{max} \leq \underline{C} \mid \Sigma u_j$ rozwiązywalny jest optymalnie przez następujący algorytm w czasie $O(B \log B)$:

Algorytm 2:

0. Pogrupuj zadania rodzinami i wybierz dowolną permutację rodzin Π. Podstaw $F := \{I_f: I_f = \Pi(1), \ldots, \Pi(B)\}$, $u_f := 0$ dla $I_f \in F$, $C := C_{\Pi(B)}$, $\rho := 0$.
1. Znajdź rodzinę $I_f \in F$, dla której $a_f := \max_{I_g \in F} \{a_g\}$, a następnie podstaw $u_f := \min\{\beta_f, (C - \underline{C})/a_f\}$, $\rho := \rho + u_f$, $C := C - a_f u_f$, $F := F - \{I_f\}$.
2. Jeśli $F \neq \varnothing$ i $C > \underline{C}$ wróć do kroku 1.
3. Jeśli $F = \varnothing$ i $C > \underline{C}$ – problem nie ma rozwiązania, w przeciwnym wypadku uzyskano optymalny rozdział zasobu, ρ jest minimalną ilością przydzielonego zasobu tak, by zachować warunek $C_{max} \leq \underline{C}$.

Rozpatrując problem $1 \mid s_f = b_f - a_f u_f,\ C_{max} \leq \underline{C} \mid \Sigma u_j$ łatwo zauważyć, że sens ma tylko rozpatrywanie $\underline{C} \in [C^{min}, C^{max}]$, gdzie C^{min} jest wartością C_{max} dla optymalnej permutacji, gdzie każdemu przezbrojeniu s_f przydzielany jest zasób $u_f = \beta_f$, natomiast C^{max} gdy $u_f = 0$.

Rozpatrując problem $1 \mid s_f = b_f - a_f u_f \mid C_{max} \wedge \Sigma u_j$ poszukujemy rozwiązań Pareto optymalnych dla wartości C_{max} również ze zdefiniowanego powyżej przedziału $[C^{min}, C^{max}]$.

Twierdzenie 3: Problem $1 \mid s_f = b_f - a_f u_f \mid C_{max} \wedge \Sigma u_j$ rozwiązywalny jest optymalnie przez następujący algorytm w czasie $O(B \log B)$:

Algorytm 3:

0. Pogrupuj zadania rodzinami i wybierz dowolną permutację rodzin Π. Podstaw $F := \{I_f: I_f = \Pi(1), \ldots, \Pi(B)\}$, $k := 0$, $u_f^k := 0$ dla $I_f \in F$, $C^k := C_{\Pi(B)}$, $\rho^k := 0$.
1. Podstaw $k := k+1$ i znajdź rodzinę $I_f \in F$, dla której $a_f = \max_{I_g \in F} \{a_g\}$, następnie podstaw $u_f^k := \beta_f$, $C^k := C^{k-1} - a_f\beta_f$, $\rho^k := \rho^{k-1} + \beta_f$, $F := F - \{I_f\}$, $u_g^k := u_g^{k-1}$ dla $I_g \in F$.
2. Jeśli $F \neq \varnothing$ wróć do kroku 1.
3. Stop – Kolejne pary (C^0, ρ^0), ..., (C^k, ρ^k), ..., (C^n, ρ^n) dla $k = 0, \ldots, n$, tworzą kolejne punkty załamania odcinkami liniowej krzywej punktów kompromisowych, przy czym permutacja Π jest niezmienna, natomiast rozdział zasobów w kolejnych punktach wyznaczają wektory u^k. Suma rozdzielonych zasobów w kolejnych punktach wynosi ρ^k.

Formalne dowody powyższych trzech twierdzeń są proste i dlatego pominiemy je. Opierają się one na spostrzeżeniu, że dla permutacji optymalnej najmniejszą możliwą liczbę przezbrojeń uzyskamy wykonując zadania rodzinami, przy czym permutacja rodzin jest dowolna. Wszystkie zadania, łącznie ze związanymi z nimi niezbędnymi przezbrojeniami, znajdują się na ścieżce krytycznej. Zasób przydzielany jest więc w pierwszej kolejności tam, gdzie uzyskamy największe skrócenie przezbrojenia, czyli do przezbrojenia o maksymalnym a_f.

4.2. Szeregowanie zadań z przezbrojeniami zależnymi od dodatkowych zasobów i kryterium L_{max}

Przypomnimy na początku pewną własność. Problem $1 \mid s_f, GT \mid L_{max}$ jest rozwiązywany optymalnie przez tzw. rozszerzoną regułę EDD (ang. *extended Earliest Due Date* - eEDD) [9]. Działa on w dwóch fazach: 1) zadania w obrębie rodzin szeregowane są według nierosnących d_j, 2) rodziny, traktowane jako zagregowane zadania, szeregowane są według niemalejących D_f, gdzie D_f jest terminem pożądanego zakończenia wykonywania zagregowanego zadania. Na zagregowane zadanie składa się przezbrojenie występujące przed rodziną i zadania należące do rodziny. Parameter D_f wyznaczany jest następująco [9]: Zakładamy, że zadania z rodziny f indeksowane są $i = j, \ldots, k$ i uszeregowane według nierosnących d_i, wtedy $D_f = \min_{j \le i \le k} \left\{ d_i + \sum_{h=i+1}^{k} p_h \right.$. Reguła szeregowania nie zależy od czasów trwania takiego zagregowanego zadania i pozostaje więc słuszna, gdy czas ten jest zmienny. W naszym przypadku zmienność ta powodowana jest przez przezbrojenie zależne od dodatkowego zasobu.

Twierdzenie 4: Problem $1 \mid s_f = b_f - a_f u_f, \Sigma u_f \le R, \mathrm{GT} \mid L_{max}$ rozwiązywalny jest optymalnie przez następujący algorytm w czasie $O(B^2)$:

Algorytm 4:

0. Uszereguj zadania i rodziny algorytmem eEDD (otrzymujemy $\Pi = \Pi(1), \ldots, \Pi(B)$), podstaw $u_{\Pi(i)} = 0$, znajdź nieterminowości wykonania poszczególnych rodzin (zadań zagregowanych) $L_{\Pi(i)} := C_{\Pi(i)} - D_{\Pi(i)}$ dla i = 1, ..., B, oraz podstaw $l := B$.
1. Znajdź najmniejszy indeks k, $1 \le k \le l$, dla którego zachodzi $L_{\Pi(k)} = \max_{1 \le i \le l}\{L_{\Pi(i)}\}$.
2. Następnie znajdź $I := \{\Pi(j) \colon 1 \le j \le k \wedge u_{\Pi(j)} = 0\}$. Jeśli $I = \varnothing$ lub $R = 0$ – stop – otrzymano rozwiązanie optymalne.
3. Znajdź indeks t, dla którego $a_{\Pi(t)} = \max_{\Pi(j) \in I}\{a_{\Pi(j)}\}$ i następnie podstaw $u_{\Pi(t)} := \min\{\beta_{\Pi(t)}, R, (L_{\Pi(k)} - \max_{1 \le i < t}\{L_{\Pi(i)}\})/ a_{\Pi(t)}\}$, $R := R - u_{\Pi(t)}$, $L_{\Pi(i)} := L_{\Pi(i)} - a_{\Pi(t)} u_{\Pi(t)}$ dla $t \le i \le B$. Ostatecznie podstaw $l := k$ i wróć do kroku 1.

Twierdzenie 5: Problem $1 \mid s_f = b_f - a_f u_f, L_{max} \le L, \mathrm{GT} \mid \Sigma u_f$ rozwiązywalny jest optymalnie przez następujący algorytm w czasie $O(B^2)$:

Algorytm 5:

0. Uszereguj zadania i rodziny algorytmem eEDD (otrzymujemy $\Pi = \Pi(1), \ldots, \Pi(B)$), podstaw $u_{\Pi(i)} := 0$, znajdź nieterminowości wykonania poszczególnych rodzin (zadań zagregowanych) $L_{\Pi(i)} := C_{\Pi(i)} - D_{\Pi(i)}$ dla i = 1, ..., B, oraz podstaw $L := \max_{1 \le i \le B} L_{\Pi(i)}$, $l := B$, $\rho := 0$.
1. Znajdź najmniejszy indeks k, $1 \le k \le l$, dla którego zachodzi $L_{\Pi(k)} = \max_{1 \le i \le l}\{L_{\Pi(i)}\}$.

2. Następnie znajdź $I := \{\Pi(j)\colon 1 \le j \le k \wedge u_{\Pi(j)} = 0\}$, jeśli $I = \varnothing$ i $L_{\Pi(k)} > \underline{L}$ – stop – problem nie ma rozwiązania.
3. Znajdź indeks t, dla którego $a_{\Pi(t)} = \max_{\Pi(j)\in I}\{a_{\Pi(j)}\}$ i następnie podstaw $u_{\Pi(t)} := \min\{\beta_{\Pi(t)}, (L - \underline{L}) / a_{\Pi(t)}, (L_{\Pi(k)} - \max_{1\le i<t}\{L_{\Pi(i)}\}) / a_{\Pi(t)}\}$, $\rho := \rho + u_{\Pi(t)}$, $L := L - a_{\Pi(t)} u_{\Pi(t)}$, $L_{\Pi(i)} := L_{\Pi(i)} - a_{\Pi(t)} u_{\Pi(t)}$ dla $t \le i \le B$. Jeśli $L = \underline{L}$, to przejdź do kroku 4, w przeciwnym wypadku podstaw $l := k$ i wróć do kroku 1.
4. Stop – wektor u jest optymalnym rozdziałem zasobów, ρ jest minimalną całkowitą ilością przydzielonego zasobu przy spełnieniu ograniczenia $L_{max} \le \underline{L}$.

Rozpatrując problem $1 \mid s_f = b_f - a_f u_f, L_{max} \le \underline{L}, \mathrm{GT} \mid \Sigma u_f$ łatwo zauważyć, że sens ma tylko rozpatrywanie $\underline{L} \in [L^{min}, L^{max}]$, gdzie L^{min} jest wartością L_{max} dla optymalnej permutacji, gdzie każdemu przezbrojeniu s_f przydzielany jest zasób $u_f = \beta_f$, natomiast L^{max} gdy $u_f = 0$.

Rozpatrując problem $1 \mid s_f = b_f - a_f u_f, \mathrm{GT} \mid L_{max} \wedge \Sigma u_f$ poszukujemy rozwiązań Pareto optymalnych dla wartości L_{max} również ze zdefiniowanego powyżej przedziału $[L^{min}, L^{max}]$.

Twierdzenie 6: Problem $1 \mid s_f = b_f - a_f u_f, \mathrm{GT} \mid L_{max} \wedge \Sigma u_f$ rozwiązywalny jest optymalnie przez następujący algorytm w czasie $O(B^2)$:

Algorytm 6:

0. Uszereguj zadania i rodziny algorytmem eEDD (otrzymujemy $\Pi = \Pi(1), \ldots, \Pi(B)$), podstaw $F := \{I_f : I_f = \Pi(1), \ldots, \Pi(B)\}$, $m := 0$, $u_{\Pi(i)}^m := 0$, $L_{\Pi(i)} := C_{\Pi(i)} - D_{\Pi(i)}$ dla $i = 1, \ldots, B$ oraz $L^m := \max_{1\le i\le B} L_{\Pi(i)}$, $\rho^m := 0$, $l := B$.
1. $m := m + 1$. Znajdź najmniejszy indeks k, $1 \le k \le l$, dla którego zachodzi $L_{\Pi(k)} = \max_{1\le i\le l}\{L_{\Pi(i)}\}$.
2. Następnie znajdź $I := \{\Pi(j)\colon 1 \le j \le k \wedge u_{\Pi(j)} = 0\}$. Jeśli $I = \varnothing$ – przejdź do kroku 4.
3. Znajdź indeks t, dla którego $a_{\Pi(t)} = \max_{\Pi(j)\in I}\{a_{\Pi(j)}\}$ i następnie podstaw $u_{\Pi(t)} := \min\{\beta_{\Pi(t)}, (L^{m-1} - L^{min}) / a_{\Pi(t)}, (L_{\Pi(k)} - \max_{1\le i<t}\{L_{\Pi(i)}\}) / a_{\Pi(t)}\}$ oraz podstaw $\rho^m := \rho^{m-1} + u_{\Pi(t)}$, $L^m := L^{m-1} - a_{\Pi(t)} u_{\Pi(t)}$, $L_{\Pi(i)} := L_{\Pi(i)} - a_{\Pi(t)} u_{\Pi(t)}$ dla $t \le i \le B$. $F := F - \{I_{\Pi(t)}\}$, $u_g^m := u_g^{m-1}$ dla $I_g \in F$. Jeśli $L^m = L^{min}$, to przejdź do kroku 4, w przeciwnym wypadku podstaw $l := k$ i wróć do kroku 1.
4. Stop – Kolejne pary $(L^0, \rho^0), \ldots, (L^m, \rho^m), \ldots, (L^n, \rho^n)$ dla $m = 0, \ldots, n$, tworzą kolejne punkty załamania odcinkami liniowej krzywej punktów kompromisowych, przy czym permutacja Π jest niezmienna, natomiast rozdział zasobów w kolejnych

punktach wyznaczają wektory u^m. Suma rozdzielonych zasobów w kolejnych punktach wynosi ρ^m.

Pominiemy dowody formalne, gdyż nie są one trudne. Opierają się one na poprawności reguły eEDD dla dowolnych czasów wykonania zadań zagregowanych, o czym już wspomniano, jak również na zasadzie przydzielania maksymalnej ilości zasobów do przezbrojenia o największej wartości parametru a i leżącego na ścieżce krytycznej.

4.3. Problemy otwarte

Autorzy rozważają również problemy:

$1 \mid s_f = b_f - a_f u_f, \Sigma u_j \le R, \mathrm{GT} \mid \Sigma w_j C_j$, $\quad 1 \mid r_j, s_f = b_f - a_f u_f, \Sigma u_j \le R, \mathrm{GT} \mid C_{\max}$,

$1 \mid s_f = b_f - a_f u_f, \Sigma w_j C_j \le \underline{C}, \mathrm{GT} \mid \Sigma u_j$, $\quad 1 \mid r_j, s_f = b_f - a_f u_f, C_{\max} \le \underline{C}, \mathrm{GT} \mid \Sigma u_j$,

$1 \mid s_f = b_f - a_f u_f, \mathrm{GT} \mid \Sigma w_j C_j \wedge \Sigma u_j$, $\quad 1 \mid r_j, s_f = b_f - a_f u_f, \mathrm{GT} \mid C_{max} \wedge \Sigma u_j$.

W chwili pisania artykułu kwestia ich złożoności obliczeniowej pozostaje jednak otwarta.

5. Wnioski

Jak wynika z lektury rozdziałów 3 i 4, niewiele jest problemów z przezbrojeniami rozwiązywalnych w czasie wielomianowym. Wszystkie problemy z przezbrojeniami sekwencyjnie zależnymi należą do klasy NP-trudnych lub od wielu lat pozostają otwarte. Przypadki z przezbrojeniami sekwencyjnie niezależnymi są łatwiejsze do rozwiązywania. Pewna liczba tych problemów okazała się rozwiązywalna optymalnie w czasie wielomianowym, łącznie z wybranymi problemami, gdzie czas przezbrojenia jest funkcją przydzielonych dodatkowo zasobów. Ponieważ generalnie problemy szeregowania zadań z przezbrojeniami są NP-trudne, celowe jest konstruowanie dobrych algorytmów przybliżonych.

LITERATURA

1. Bruno J., Downey P.: *Complexity of task sequencing with deadlines, set-up times and changeover costs,* SIAM Jurnal on Computing 7 393-404, 1978.
2. Cheng T.C.E., Kovalyov M.Y.: *Single machine batch scheduling with deadlines and recource dependent processing times,* Operations Research Letters 17, p. 243-249, 1995.

3. Chudzik K., Janiak A.: *Szeregowanie zadań z przezbrojeniami maszyn: przegląd zagadnień i literatury,* Automatyka, Tom 1, Zeszyt 1, Wydawnictwo AGH, Kraków, s. 71-79, 1997.

4. Janiak A., Kovalyov M.Y.: *Single machine scheduling with deadlines and and resource dependent processing times,* Working Paper, Institute of Engineering Cybernetics, Wrocław University of Technology, 1993.

5. Janiak A.: *Dokładne i przybliżone algorytmy szeregowania zadań i rozdziału zasobów w dyskretnych procesach przemysłowych,* Prace Naukowe Instytutu Cybernetyki Technicznej Politechniki Wrocławskiej, Seria: Monografie, Wrocław 1991.

6. Janiak A.: *Single machine scheduling problem with a common deadline and resource dependent release dates,* European J. Oper. Res. 53, p. 317-325, 1991.

7. Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G., Shmoys D.B.: *Sequencing and scheduling algorithms and complexity,* Report BS-R8909, Centre for Mathematics and Computer Science, Amsterdam, The Nederlands, 1989.

8. Monma C.L., Potts C.N.: *On the complexity of scheduling with batch setup times, Operations Research,* Vol. 37, No. 5, p. 798-804, 1989.

9. Potts C.N., Van Wassenhove L.N.: *Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity,* J. Opl. Res. Soc. 43, p. 395-406, 1992.

10. Webster S., Baker K.R.: *Scheduling groups of jobs on a single machine,* Operations Research, vol 43, No. 4, p. 692-703, 1995.

11. Zdrzałka S.: *Harmonogramowanie zadań, grupowanie i porcjowanie z przezbrojeniami maszyn,* Zeszyty Naukowe Politechniki Śląskiej, seria: Automatyka, z. 109, s. 353-366, 1992.

Recenzent:Prof. dr hab. Jerzy Klamka

Abstract

The paper deals with the single machine scheduling problems with setups. Problems of task scheduling with setups are an important part of the general scheduling theory. Many practical scheduling problems involve processing several families of related jobs on common facilities where a setup time and/or cost is incurred whenever there is a switch form processing a job in one family to a job in another family. The main aim of optimisation is task grouping (batching) and sequencing in order to minimise appearance of setup in production time and cost. The paper contains some introduction into tasks scheduling problems with setups on a single machine. The general mathematical model of the problems under consideration is presented. Models with and without group technology assumption are considered. Computational complexity and results for the classic problems are presented. New models with setups that are some functions of the alloted resources are introduced. Their computational complexity is considered. Algorithms for the cases solvable in polynomial time are presented.