ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1988
Seria: AUTOMATYKA z.94 Nr kol.970

Adam Janiak
Politechnika Wrocławska

PERMUTACYJNY PROBLEM PRZEPŁYWOWY Z LINIOWYMI MODELAMI OPERACJI

Streszczenie. W artykule dokonano przeglądu problematyki rozdziału zasobów i szeregowania operacji na maszynach. Szczegółowo przedstawiono rezultaty uzyskane dla klasycznego permutacyjnego problemu przepływowego. Następnie problem ten uogólniono na przypadek z liniowymi modelami operacji. Szczegółowo rozpatrzono dwumaszynowy problem przepływowy, a uzyskane rezultaty uogólniono na przypadek wielomaszynowy.

1. Wstęp

Jeden z ważnych aktualnie kierunków rozwojowych automatyki dotyczy sterowania dyskretnymi procesami technologicznymi i produkcyjnymi. Obszerna klasa tych problemów sprowadza się do problemów szeregowania zadań na maszynach i równoczesnego rozdziału ograniczonych zasobów różnych rodzajów [3 , 29]. Analogiczne problemy występują w komputerowych systemach sterowania [7 , 43 , 1]. Proces produkcyjny charakteryzuje się "przepływem" materiałów i półproduktów w postaci pojedynczych elementów, podzespołów i zespołów. Elementami mogą być detale, kęsiska, wlewki, odkuwki, programy maszyn cyfrowych. Elementy te są poddawane operacjom obróbki na kolejnych stanowiskach /gniazdach/ produkcyjnych wyposażonych w maszyny. Przez maszyny należy rozumieć: maszyny do obróbki mechanicznej /np. obrabiarki/, maszyny budowlane, piece grzewcze, zgniatacze, klatki walcownicze, stanowiska montażu, środki transportu, procesory systemów komputerowych itp. Operacjami natomiast mogą być: obróbka mechaniczna, czynności montażu, czynności transportowe, wytop stali, walcowanie, podgrzewanie, obróbka ręczna, obliczenia w systemach komputerowych itp.

Proces przetwarzania elementów charakteryzuje ciąg kolejno po sobie następujących operacji. Tego rodzaju ciąg operacji jest nazywany zadaniem. W ogólności, zadanie w dyskretnym systemie produkcyjnym stanowi zbiór operacji uwarunkowanych czasowo. W ogólnym przypadku zadania /operacje/ do swego wykonania mogą wymagać, oprócz maszyn, także innych dodatkowych zasobów dostępnych w ograniczonych ilościach, a czasy trwania operacji mogą zależeć od ilości zasobów im przydzielonych. Problem optymalizacji procesu produkcyjnego w tym przypadku można scharakteryzować następująco.

Dany jest zbiór zadań produkcyjnych, zbiór maszyn oraz ograniczone iloś zasobów, przeznaczone do wykonania tych zadań. Należy znaleźć takie dop szczalne przyporządkowanie operacji do poszczególnych maszyn, taką dop szczalną kolejność wykonywania operacji na tych maszynach oraz taki d puszczalny rozdział ograniczonych zasobów pomiędzy operacje /lub pomię maszyny/, aby przyjęte kryterium efektywności było optymalizowane.

Niniejsza praca poświęcona jest pewnej klasie wspomnianych powyżej problemów szeregowania z rozdziałem ograniczonych zasobów, a mianowicie permutacyjnym problemom taśmowym przy założeniu, że czasy trwania operacji /na pewnych maszynach/ zależą liniowo od ilości ograniczonych zas bów /rozdział 4/. Wcześniej w rozdziale 2 dokonano krótkiego przeglądu problematyki rozdziału zasobów i szeregowania operacji na maszynach. A następnie w rozdziale 3 omówiono w miarę szczegółowo dotychczasowe wy niki osiągnięte dla klasycznego permutacyjnego problemu przepływowego. Uwagi końcowe znajdują się w rozdziale 5.

2. Przegląd problematyki rozdziału zasobów i szeregowania operacji na m szynach

Począwszy od wczesnych lat 60 bardzo intensywnie była rozwijana teoria rozdziału zasobów przy ustalonej kolejności wykonywania operacji, w szczególności dotyczy to tzw. zagadnienia planowania sieciowego /anal zy sieciowej, activity networks/. W sposób istotny wyróżniono tu proble PERT/koszt /time-cost tradeoff/ [28]. W problemie tym przyjmuje się mod le czynności /operacji/ w postaci koszt/czas. Zakłada się tutaj, że cza wykonania czynności może zmieniać się w sposób ciągły w pewnym zadanym przedziale, przy czym zmniejszenie czasu wykonania czynności powoduje wzrost kosztu jej wykonania. Zwykle przyjmowano modele czynności w post ci nierosnących funkcji liniowych lub odcinkami liniowych lub wypukłych bądź też wklęsłych. W klasycznym sformułowaniu problemu PERT/koszt wyzn cza się najmniejszy koszt wykonania całej sieci czynności /tzw. projekt przy założeniu, że ten projekt musi być wykonany w zadanym czasie. Ogól niej, zwykle wyznacza się tzw. całą krzywą projektu, rozwiązując powyżs problem dla całego dopuszczalnego przedziału zmienności czasu wykonania projektu.

Rozpatrywano także problemy równomiernego wykorzystania zasobów /res urce leveling/ oraz problemy rozdziału zasobów, których ilość jest ogra niczona w każdej chwili czasu /constrained - resource project scheduli

Podsumowanie wyników osiągniętych przy wspomnianych wyżej modelach operacji można znaleźć w monografii [8].

Znacznie ogólniejsze modele operacji w postaci równań różniczkowych, wiążące prędkość wykonywania operacji z przydzielonymi ilościami zasobów w każdej chwili czasu /przy zadanych rozmiarach operacji/, zostały wprowadzone w [5] /w warunkach deterministycznych/ oraz w [4] /w warunkach probabilistycznych. Rozpatrywano szereg istotnych problemów rozdziału zasobów przy powyższych modelach operacji. Wyniki uzyskane /w warunkach deterministycznych/ do 1976 roku podsumowano w [42]. Niestety, dalsze bardzo istotne wyniki nie doczekały się - jak dotąd - monograficznego ujęcia.

Z kolei począwszy od lat 50 [26] datuje się intensywny /szczególnie okres lat 70/ rozwój teorii szeregowania zadań na maszynach. Otrzymano szereg interesujących rezultatów dotyczących złożoności obliczeniowej, algorytmów wielomianowych, przybliżonych /aproksymacyjnych/ oraz dokładnych opartych z reguły na metodzie podziału i ograniczeń. Podsumowanie tych wyników można znaleźć, np.w [31]. Prawie we wszystkich pracach przyjmowano, klasyczny dla tej teorii, model operacji, podawany w postaci ustalonego czasu wykonywania oraz zakładano, że do wykonania danej operacji potrzebna jest tylko jedna maszyna o pojemności jeden, tzn. maszyna, która w każdej chwili czasowej może wykonywać co najwyżej jedną operację. Jednakże przy próbach zastosowania otrzymanych rezultatów do sterowania dyskretnymi procesami przemysłowymi zauważono, że takie modele są zbyt dużym uproszczeniem skomplikowanej rzeczywistości. Dlatego też, klasyczne sformułowania rozszerzano o różne dodatkowe elementy. Począwszy od pracy Gareya i Johnsona z 1975 [10] datuje się rosnące zainteresowanie problemami szeregowania z ograniczeniami zasobowymi. A mianowicie, zakłada się tutaj, że czas wykonania operacji na danej maszynie jest nadal ustalony, ale operacja do swego wykonania oprócz maszyny wymaga dodatkowych podzielnych w sposób dyskretny zasobów, dostępnych w ograniczonej ilości. Następnie problemy te były rozpatrywane przy różnych założeniach odnośnie do maszyn, operacji, ilości zasobów podzielnych w sposób dyskretny. Wszystkie dotychczasowe wyniki badań dotyczące tych problemów zostały przedstawione w monografii [2].

Z przeglądu dokonanego powyżej, wynika, że do tej pory zajmowano się problemami szeregowania zadań na maszynach, w najogólniejszej postaci, przy założeniu, że operacja do swego wykonania może wymagać jednej lub więcej maszyn oraz pewnych dodatkowych podzielnych w sposób dyskretny zasobów. W najogólniejszym przypadku można mówić nawet o wielu /skończonych, z góry ustalonych/ sposobach wykonywania operacji [37]. Przy czym czas wykonania operacji /danym sposobem/ był z góry ustalony. Przyjęcie takiego założenia stanowi w wielu sytuacjach praktycznych istotne uproszczenie rzeczywistości, bowiem czas trwania operacji może w ogólnym przypadku zmieniać się w sposób ciągły w określonych granicach, wynikających z technologicznych uwarunkowań i może być zmienną decyzyjną podlegającą optymalizacji. Na przykład, czas trwania operacji może zależeć od ilości przydzielonej tej operacji /lub maszynie, na której jest wykonywana/ zasobów podzielnych w sposób ciągły. Przy czym globalnie dostępna ilość zasobów jest ograniczona, a z uwarunkowań technologicznych wynika, że dana operacja może pobierać zasoby ze z góry określonego ciągłego przedziału.

Ogólnie rzecz biorąc, czasy wykonywania operacji na maszynach mogą być pewnymi nierosnącymi funkcjami ilości ograniczonych podzielnych w sposób ciągły zasobów przydzielonych do ich wykonania. Zasobami takimi mogą być: nakłady finansowe, energia, tlen, gaz, paliwo, katalizator itp. Powyższy model skrótowo oznaczany będzie czas/zasób.

W problemach szeregowania zadań na maszynach wspomniane wyżej modele operacji zostały po raz pierwszy wprowadzone w [25]. Niezależnie, w [41] wprowadzono podobne /odwrotne/ modele operacji /liniowa zależność kosztu wykonania operacji od jej czasu zakończenia, tzn. modele koszt/czas/ dla pewnego jednomaszynowego problemu szeregowania operacji. Te ostatnie modele były następnie rozpatrywane w pracach [40, 39, 32]. Natomiast modele czas/zasób dla jednomaszynowych problemów szeregowania operacji przy różnych założeniach odnośnie funkcji kryterialnych, ograniczeń kolejnościowych, momentów dostępności operacji były rozpatrywane w [15,19,22,23,24], dla problemu gniazdowego w [14], a dla niepermutacyjnego problemu taśmowego w [17,20].

Niniejsza praca będzie pierwszą próbą zastosowania modeli czas/zasób do permutacyjnego problemu taśmowego, co będzie miało miejsce w rozdziale 4. Zanim to jednak zostanie zrobione, w następnym rozdziale krótko podsumowane zostaną rezultaty osiągnięte dla tego problemu, ale przy klasycznym założeniu o zadanych czasach wykonywania operacji.

3. Podsumowanie rezultatów uzyskanych dla klasycznego permutacyjnego problemu przepływowego

Klasyczny permutacyjny problem przepływowy można sformułować w następujący sposób. Dany jest zbiór zadań $J_1, J_2, \ldots, J_n$, z których każde musi być wykonywane po kolei na maszynach $M_1, M_2, \ldots, M_m$. Zadania na każdej z maszyn muszą być wykonywane w tej samej kolejności /permutation schedule/. Każda z maszyn może wykonywać co najwyżej jedno zadanie w każdej chwili czasu. Każde zadanie J_i, $i=1,2,\ldots,m$, zatem składa się z ciągu m operacji $O_{i1}, O_{i2}, \ldots, O_{im}$, a operacja O_{iv} odpowiada wykonywaniu zadania J_i na maszynie M_v w czasie p_{iv} bez przerwań /zadania niepodzielne/. Problem polega na znalezieniu takiej kolejności wykonywania zadań $\pi \in \Pi$ /gdzie Π jest zbiorem wszystkich permutacji liczb $1,\ldots,n$/ jednakowej dla wszystkich maszyn, aby zminimalizować czas wykonania wszystkich zadań $C_{max}(\pi)$.

Fundamentalny algorytm rozwiązujący powyższy problem dla przypadku $m = 2$ został podany w 1954 roku przez Johnsona [26], który wykazał, że istnieje optymalne uszeregowanie zadań, w którym zadanie J_j poprzedza J_k, jeśli $\min\{p_{j1}, p_{k2}\} \leq \min\{p_{j2}, p_{k1}\}$. Z powyższego wynika, że problem ten można rozwiązać w $O(n \cdot \log n)$ krokach, wykonując najpierw zadania z $p_{j1} \leq p_{j2}$ w porządku niemalejących p_{j1}, a następnie pozostałe zadania w porządku nierosnących p_{j2}.

Jednakże już dla $m = 3$ powyższy problem staje się silnie NP-trudny [11]. /Definicje pojęć związanych z teorią złożoności obliczeniowej można znaleźć np. w [1]/. Nawet dla $m = 2$, jeśli dopuści się niezerowe momenty dostępności zadań albo ograniczenia kolejnościowe pomiędzy zadaniami /nawet w postaci drzewa/, to problemy te stają się już silnie NP-zupełne [11]. Dopuszczenie przerywalności operacji zwykle powoduje ułatwienie /w sensie złożoności obliczeniowej/ rozwiązania problemu. Jednakże w tym przypadku nawet przy dopuszczeniu przerywalności operacji problem z $m = 2$ i niezerowymi momentami dostępności zadań oraz problem z $m = 3$ pozostają nadal silnie NP-trudnymi [12].

Do optymalnego rozwiązania ogólnego problemu stosowano różne implementacje metody podziału i ograniczeń. Stosowano różne strategie podziału drzewa rozwiązań, różne dolne ograniczenia i kryteria eliminacyjne, pozwalające na zaniechanie poszukiwania rozwiązania optymalnego w pewnych partiach drzewa rozwiązań. Z istotnych prac należałoby tutaj wymienić pracę [30], w której węzeł drzewa rozwiązań charakteryzuje się częściowym uporządkowaniem zadań oraz pracę [13], w której w każdym węźle jest pełne rozwiązanie dopuszczalne.

Do rozwiązania rozpatrywanego problemu stosowano także algorytmy suboptymalne /przybliżone/. Najbardziej aktualnego przeglądu i eksperymentalnego porównania tych metod dokonano w [38], z którego wynika, że najlepszą - jak dotąd - metodą przybliżoną jest heureza zaproponowana w [6].

Niestety analizę najgorszego przypadku udało się przeprowadzić dla niewielu algorytmów heurystycznych rozwiązujących rozważany problem. W pracy [12] wykazano, że dla dowolnego tzw. zwartego uszeregowania /busy schedule, charakteryzującego się tym, że w każdej chwili czasu od momentu rozpoczęcia aż do momentu zakończenia wykonywania zadań co najmniej jedna maszyna realizuje jakieś zadanie/ współczynnik najgorszego przypadku równa się m. Przy czym granica ta jest osiągalna nawet dla tzw. LPT uszeregowania, w którym zadania szeregowane są zgodnie z nierosnącymi sumami czasów wykonywania operacji zadania. W tej samej pracy podano także inny przybliżony algorytm, bazujący na algorytmie Johnsona / o złożoności obliczeniowej $O(m \cdot n \cdot \log n)$ o współczynniku najgorszego przypadku równym $\lceil \frac{m}{2} \rceil$ /gdzie $\lceil \frac{m}{2} \rceil$ najmniejsza liczba całkowita, taka, że $\lceil \frac{m}{2} \rceil \geq \frac{m}{2}$ /. Przykład osiągalności tego ograniczenia udało się pokazać tylko dla $m = 3$.

Rozpatrywano także permutacyjny problem przepływowy przy założeniu, iż jeśli rozpoczęto realizację jakiegoś zadania, to musi ono być realizowane bez przerwań aż do jego zakończenia /tzw. problem "bez czekania" /no wait//. Innymi słowy, w każdym zadaniu po rozpoczęciu jego wykonywania operacje muszą być wykonywane natychmiast jedna po drugiej. Problemy takie występują na przykład w hutach, gdzie pewne wlewki nie mogą wystygnąć i muszą być walcowane mając odpowiednio wysoką temperaturę. Problem taki [35] można sformułować jako problem komiwojażera z miastami 0,1,...,n i z następującymi odległościami międzymiastowymi:

$$c_{jk} = \max_{1 \leq i \leq m} \left\{ \sum_{h=1}^{i} p_{jh} - \sum_{h=1}^{i-1} p_{kh} \right\} \quad (j,k = 0,1,\dots,n),$$

gdzie $p_{oi} = 0 \quad (i = 1,\dots,m)$.

Można wykazać, że przypadek tego problemu z dwiema maszynami można rozwiązać w wielomianowym czasie w $O(n^2)$ krokach [36]. Natomiast przypadek z czterema maszynami jest już silnie NP-trudny [33]. Problem z trzema maszynami jest problemem otwartym, tzn. ani nie wykazano jego NP-trudności, ani nie znaleziono algorytmu wielomianowego.

Wprowadzenie ograniczenia "bez czekania" może w sposób istotny zwiększyć optymalną wartość kryterium C_{max}(no wait). Można jednakże pokazać, że:

$$C^*_{max}(\text{no wait})/C^*_{max} < m, \quad \text{dla} \quad m \geqslant 2,$$

gdzie C^*_{max} jest optymalną wartością kryterium C_{max} dla problemu bez ograniczenia: "bez czekania".

W następnym rozdziale uogólnimy rozpatrywany powyżej permutacyjny problem kolejnościowy na przypadek, gdy czasy wykonywania operacji na pewnych maszynach zależą liniowo od ilości przydzielonych im zasobów.

4. Permutacyjny problem przepływowy z ograniczeniami zasobowymi

W niniejszym rozdziale rozpatrzymy permutacyjny problem przepływowy /sformułowany w poprzednim rozdziale/ przy założeniu, że czasy wykonywania operacji na pewnych maszynach liniowo zależą od ilości przydzielonych im zasobów. Nie umniejszając zbytnio ogólności rozważań, celem zwięzłego zapisu, założymy, że czasy wykonywania wszystkich operacji zależą od ilości przydzielonych im zasobów, przy czym każda maszyna M_v, $v = 1,2,\dots,$ dysponuje ilością zasobów U_v. Model operacji O_{iv} jest następujący:

$$p_{iv} \triangleq p_{iv}(u_{iv}) \triangleq b_{iv} - a_{iv}u_{iv}, \tag{1}$$

gdzie $b_{iv} > 0$, $a_{iv} > 0$ są zadanymi parametrami modelu, a u_{iv} jest ilością zasobu przydzielonego do wykonania operacji O_{iv}. Zbiór dopuszczalnych rozdziałów zasobów jest następujący:

$$U \triangleq \Big\{u \in R^{mn}:\ u \triangleq [u_1,\ldots,u_v,\ldots,u_m] \wedge$$
$$\wedge \forall (v = 1,\ldots,m)\Big[(u_v \triangleq [u_{1v},\ldots,u_{iv},\ldots,u_{iv}]) \wedge$$
$$\wedge \forall (i = 1,\ldots,n)\,(\alpha_{iv} \le u_{iv} \le \beta_{iv}) \quad \wedge \quad \sum_{i=1}^{n} u_{iv} \le U_v\Big]\Big\},$$

gdzie α_{iv}, β_{iv} $(0 \le \alpha_{iv} \le \beta_{iv} \le b_{iv}/a_{iv})$ są zadanymi technologicznymi ograniczeniami na zakres zasobów przydzielanych operacji O_{iv}; $\sum_{i=1}^{n} \alpha_{iv} \le U_v \le \sum_{i=1}^{n} \beta_{iv}$.

Problem tutaj polega na znalezieniu takiej kolejności wykonywania zadań $\pi^* \in \Pi$ oraz takiego rozdziału zasobów $u^* \in U$, aby czas wykonania wszystkich zadań $C_{max}(\pi, u)$ osiągnął wartość minimalną, tzn.:

$$\min_{\pi \in \Pi} \min_{u \in U} C_{max}(\pi, u) = C_{max}(\pi^*, u^*).$$

W poprzednim rozdziale zostało podkreślone, że dla klasycznego permutacyjnego problemu przepływowego granica między wielomianowo rozwiązywalnym problemem a NP-trudnym leży między m = 2 a m = 3. Jak się za chwilę okaże, dla problemu z zasobami leży ona "niżej", bo już problem dwumaszynowy, z zasobami tylko na jednej z maszyn, i z identycznymi współczynnikami nachylenia $a_{iv} = a_v$, $i = 1,2,\ldots,n$, $v = 1$ albo 2 jest NP-trudny. Dokładniej, decyzyjna wersja tego problemu jest NP-zupełna. Można ją sformułować następująco:

DWP: Dany jest dwumaszynowy system wykonywania n niepodzielnych zadań o czasach wykonywania p_{i2} na drugiej maszynie, modele (1) zadań na pierwszej maszynie, z parametrami $a_{i1}=a_1$, b_{i1}, $\alpha_{i1}=0$, $_{i1}$, $i=1,2,\ldots,n$, oraz liczbami U_1 i y /gdzie n, a_1, b_{i1}, p_{i2} są nieujemnymi liczbami całkowitymi, a U_1 oraz y są nieujemnymi liczbami wymiernymi/.

Pytanie: Czy istnieje $\pi \in \Pi$ oraz $u \in U$ takie, że $C_{max}(\pi, u) \le y$?.

<u>Własność 1.</u>

DWP jest NP-zupełna[32].

W dowodzie korzysta się z decyzyjnej wersji problemu PODZIAŁU ZBIORU, o której wiadomo [27], że jest NP-zupełna. Ma ona postać:

DWPZ: Dane są liczby naturalne k, e_1, $e_2,\ldots,e_k$, E takie, że $\sum_{i=1}^{k} e_i = 2E$. Pytanie: czy istnieje $Q \subset K \triangleq \{1,2,\ldots,k\}$, taki, że $\sum_{i \in Q} e_i = E$?

Wykorzystując następujący szczególny przypadek problemu

DWP: $n = k+1$, $a_1 = 1$, $b_{i1} = 2e_i$, $\beta_{i1} = 2e_i$, $p_{i2} = e_i$,

dla $i = 1,2,\ldots, k=n-1$, $b_{n1} = E$, $\beta_{n1} = 0$, $p_{n2} = 2E$, $U_1 = 2E$, $y = 4E$, można wykazać wielomianową transformowalność DWPZ do DWP. /Przynależność DWP do klasy zadań NP jest oczywista/.

Ponieważ DWP jest NP-zupełna, więc permutacyjny problem przepływowy przy $m = 2$, $a_{i1} = a_1$, $p_{i2} = b_{i2}$, $i = 1,2,\ldots,n$, jest NP-trudny /oznaczmy ten ostatni problem przez P2/, co powoduje, że nie istnieje algorytm dokładny o złożoności wielomianowej /jeżeli $P \neq NP$/ rozwiązujący ten problem. Nie znaczy to jednak, że przy przyjęciu dodatkowych ograniczeń nie można podać algorytmów efektywnych. Łatwo zauważyć, że problem P2 dla przypadku z ustalonym rozdziałem zasobów może być trywialnie rozwiązany algorytmem Johnson w $O(n \cdot \log n)$ krokach; z drugiej strony, jeśli permutacja π jest ustalona, wówczas może być on rozwiązany w $O(n)$ krokach za pomocą algorytmu podobnego do algorytmu przedstawionego w [15]. Z kolei łatwo można pokazać, że jeśli w P2 $p_{i2} = p_2$, wówczas następujący algorytm w $O(n^2)$ krokach daje optymalne rozwiązanie:

Algorytm 1:

Krok 1. Podstaw $N = \{1,2,\ldots,n\}$, $k:=1$

Krok 2. Wyznacz $p_{i1}:=p_{i1}(\min(\beta_{i1},U_1))$ dla $i \in N$.

Krok 3. Znajdź $j \in N$ spełniające $p_{j1} = \min_{i \in N} \{p_{i1}\}$, następnie podstaw $\pi^*(k):=j$, $u^*_{j1}:=\min\{\beta_{i1},U_1\}$ $U_1:=U_1-u^*_{j1}$, $k:=k+1$ oraz $N:=N \setminus \{j\}$. Jeśli $N \neq \emptyset$, przejdź do Kroku 2, w przeciwnym przypadku – STOP: π^* oraz u^* dają optymalne rozwiązanie.

Natomiast przypadek problemu P2 z $b_{i1}=b_1$, $\beta_{i1}=\beta_1$, $i=1,2,\ldots,n$, można trywialnie rozwiązać w $O(n \cdot \log n)$ krokach szeregując zadania zgodnie z nierosnącymi wartościami p_{i2} (otrzymujemy permutację π^*) oraz rozdzielając zasoby zgodnie z następującym algorytmem /złożoność obliczeniowa $O(n)$/:

Algorytm 2:

Krok 1. Dla $i:=1$ do n wykonaj Krok 2

Krok 2. Podstaw $u^*_{\pi^*(i)1} := \min\{\beta_{\pi^*(i)1}, U_1\}$, $U_1:=U_1-u^*_{\pi^*(i)1}$.

W powyższym algorytmie i dalej przez $u^*_{\pi} \in U$ będzie oznaczany optymalny rozdział zasobów dla permutacji $\pi \in \Pi$, tzn. rozdział, dla którego:

$$C_{max}(\pi, u^*_{\pi}) \triangleq \min_{u \in U} C_{max}(\pi, \cdot).$$

Dla ogólnego problemu P2 /bez specyficznych założeń upraszczających/ zaproponowany zostanie następujący algorytm heurystyczny o złożoności obliczeniowej $O(n^2)$:

Algorytm heurystyczny:

Krok 1. Podstaw $N:=\{1,2,\ldots,n\}$, $k:=1$, $l:=n$.

Krok 2. Wyznacz $p_{i1} = p_{i1}(\min(\beta_{i1},U_1))$ dla $i \in N$.

Krok 3. Znajdź $j \in N$ oraz $q \in \{1,2\}$ spełniające $p_{jq} = \min_{i \in N}\{p_{i1}, p_{i2}\}$. Jeśli $q=1$, podstaw $\pi_H(k):=j$, $u^*_{\pi_H(k)1}:=\min(\beta_{\pi_H(k)1}, U_1)$. $U_1:=U_1-u^*_{\pi_H(k)1}$, $k:=k+1$, $N:=N-\{j\}$ i jeśli $N \neq \emptyset$, przejdź do Kroku 2, w przeciwnym przypadku przejdź do Kroku 4. Natomiast jeśli $q = 2$, podstaw $\pi_H(l):=j$, $l:=l-1$, $N:=N-\{j\}$ i jeśli $N \neq \emptyset$, przejdź do Kroku 2, w przeciwnym przypadku przejdź do Kroku 4.

Krok 4. Dla $i:=k$ do n wykonaj:

podstaw $u^*_{\pi_H(i)1} := \min\left(\beta_{\pi_H(i)}, U_1\right)$, $U_1 := U_1 - u^*_{\pi_H(i)1}$. Następnie

STOP – π_H wraz z $u^*_{\pi_H}$ jest poszukiwanym rozwiązaniem przybliżonym.

Łatwo zauważyć, że powyższy algorytm heurystyczny spełnia warunki konieczne optymalności dla problemu P2, tzn. $\min_{u \in U} C_{max}(\pi_H, \cdot) = C_{max}(\pi_H, u^*_{\pi_H})$ oraz $\min_{\pi \in \Pi} C_{max}(\cdot, u^*_{\pi_H}) = C_{max}(\pi_H, u^*_{\pi_H})$.

Przeprowadzono analizę najgorszego przypadku dla powyższego algorytmu przybliżonego. Z analizy tej wynika, że współczynnik najgorszego przypadku dla tego algorytmu jest równy 2. Można także pokazać, że granica ta jest osiągalna /tzn. podać przykład, dla którego ten współczynnik wynosi dokładnie 2/. Z punktu widzenia analizy najgorszego przypadku uzyskany wynik nie jest rewelacyjny, ponieważ można łatwo pokazać, że dla heurezy, w której bierzemy dowolną permutację, a następnie dla niej znajdujemy optymalny rozdział zasobów, ten współczynnik wynosi także 2 /i jest osiągalny/. Tym niemniej zaproponowana heureza wykazuje bardzo wysoką dokładność eksperymentalną. A mianowicie, przetestowano 360 losowo generowanych przykładów, dla każdego z nich wyznaczano następujący współczynnik $w = \left[(C^H_{max} - C^*_{max})/C^*_{max}\right] \cdot 100\%$, gdzie C^H_{max} – wartość kryterium otrzymana za pomocą proponowanego algorytmu, C^*_{max} optymalna wartość uzyskana za pomocą algorytmu podziału i ograniczeń, przedstawionego w dalszej części tego rozdziału. Arytmetyczna średnia 360 wyliczonych współczynników w wynosiła 0.00625%, a wartość maksymalna 0.47921%.

Obecnie przejdziemy do przestawienia algorytmu dokładnego rozwiązującego problem P2, a następnie pokażemy, jak ten algorytm uogólnić na ogólny przypadek z dowolną ilością maszyn m i podamy wyniki obliczeniowe dla pewnych przykładów z 5 maszynami i 20 zadaniami /100 operacjami/. Zanim jednak to zrobimy, przedstawimy najpierw pewne własności problemu, na których bazuje algorytm.

Zaczniemy od pewnych definicji i oznaczeń.

Trójkę liczb $\langle \pi(1), \pi(j), \pi(n) \rangle$, dla której zachodzi

$$C_{max}(\pi, u^*_\pi) = \sum_{i=1}^{j} \left(b_{\pi(i)1} - a_{\pi(i)1} u^*_{\pi(i)1}\right) + \sum_{i=j}^{n} p_{\pi(i)2},$$

będzie nazywana drogą krytyczną dla π /dokładniej, ta trójka wyznacza drogę krytyczną w π/. Wartość $C_{max}(\pi, u^*_\pi)$ będzie nazywana długością drogi krytycznej w π, a liczba naturalna j będzie nazywana krytyczną pozycją w π.

Łatwo zauważyć, że może być wiele dróg krytycznych /oczywiście o tej samej długości $C_{max}(\pi, u^*_\pi)$/ w π przy rozdziale u^*_π. Niech k_π oznacza liczbę wszystkich dróg krytycznych w π; oraz niech $\langle j_1, j_2, \ldots, j_k, \ldots, j_{k_\pi} \rangle$ będzie /uporządkowanym/ ciągiem kolejnych krytycznych pozycji w π /tzn. $j_1 < j_2 < \ldots, j_{k_\pi}$ /.

Podciągi zadań $\langle J_{\pi(1)},\dots,J_{\pi(j_1)}\rangle,\dots,$
$\langle J_{\pi(j_{k-1})}, J_{\pi(j_{k-1}+1)},\dots,J_{\pi(j_k)}\rangle,\dots,\langle J_{\pi(j_{k_\pi})}, J_{\pi(j_{k_\pi}+1)},\dots,J_{\pi(n)}\rangle$
permutacji π będą nazywane, odpowiednio, pierwszą,...,k-tą,..., i $k_\pi+1$-szą /ostatnią/ sekcję zadań w π /przy rozdziale zasobów u^*_π/.

Można wykazać, że prawdziwa jest następująca własność:

Własność 2:

Dla każdej permutacji zadań $\pi \in \Pi$, jeśli permutacja β została otrzymana z π przez pewną zmianę kolejności wykonywania zadań i jeśli $C_{max}(\beta, u^*_\beta) < C_{max}(\pi, u^*_\pi)$, wówczas w β:

/1/ co najmniej jedno zadanie z pierwszej sekcji zostało przesunięte za ostatnie zadanie z tej sekcji; lub

/2/ co najmniej jedno zadanie k-tej sekcji jest przesunięte przed pierwsze albo za ostatnie zadanie tej sekcji, dla k=2 lub 3,..., lub k_π; lub

/3/ co najmniej jedno zadanie z ostatniej /$k_\pi+1$-ej/ sekcji w π zostało przesunięte przed pierwsze zadanie tej sekcji.

Wykorzystując powyższe własności eliminacyjne, skonstruowano algorytm, oparty na metodologii podziału i ograniczeń, celem rozwiązania problemu P2.

Algorytm ten był testowany na 360 losowo generowanych przykładach. Dokładniej, było po 30 przykładów z 50,100,150 i 200 operacjami, rozpatrywanymi dla trzech skrajnych wartości U_1 /dokładniej dla wartości $0.1 \cdot \sum \beta_1$, $0.5 \cdot \sum \beta_1$, $0.9 \cdot \sum \beta_1$/. Maksymalny czas rozwiązania przykładów z 200 operacjami wynosił 116 sekund /na maszynie cyfrowej ODRA 1325/. Dla 93.88 % generowanych 360 przykładów, algorytm znajdował rozwiązanie optymalne w pierwszym węźle drzewa rozwiązań, a dla 5.00 % w drugim węźle. Rozwiązanie początkowe w algorytmie było uzyskiwane za pomocą wspomnianej heurezy. Powyższe bardzo skrótowo przedstawione rezultaty badań empirycznych świadczą o tym, że zaproponowany algorytm, mimo nie najlepszego oszacowania w sensie analizy najgorszego przypadku, charakteryzuje się bardzo dobrą oceną eksperymentalną; a zaproponowany algorytm dokładny, z kolei, charakteryzuje się szybkim rozpoznawaniem rozwiązania optymalnego.

Przejdźmy obecnie do ogólnego m maszynowego permutacyjnego problemu przepływowego. Można pokazać, że przedstawione własności problemu P2, sformułowane we Własności 2, można uogólnić na ogólny m maszynowy przypadek. Na bazie tych uogólnionych własności można skonstruować algorytm typu podziału i ograniczeń. W celu wyznaczenia optymalnego rozdziału zasobów $u^*_\pi \in U$ dla ustalonej permutacji $\pi \in \Pi$ zastosowano pewną modyfikację algorytmu podanego w [34]. Ten uogólniony algorytm testowano na 25 przykładach, każdy z nich miał 5 maszyn i 20 zadań /tzn. 100 operacji/.

Niestety wyniki nie były już tak znakomite jak dla algorytmu opracowanego dla dwu maszyn, w którym istotne znaczenie miało zastosowanie wspomnianego algorytmu heurystycznego i bardzo dobrych dolnych ograniczeń. Dla tego ogólnego algorytmu czasy obliczeń /ALGOL na ODRZE 1300/ były od kilku do kilkudziesięciu minut. Dla 6 przykładów na 25, tzn. dla 24% badanych przykładów, nie znaleziono rozwiązania optymalnego w ciągu godziny. Wspomniany algorytm wymaga dalszych ulepszeń, w szczególności opracowania nowych, w sposób istotny lepszych algorytmów heurystycznych, dających rozwiązanie początkowe w dokładnym algorytmie. Istotny wpływ ma także opracowanie lepszych dolnych ograniczeń, jednakże - zdaniem autora - zbyt wiele tutaj nie da się już zrobić /mając na myśli dolne ograniczenia wyznaczane w wielomianowym czasie/.

5. Zakończenie

W dalszych badaniach dotyczących szeregowania operacji na maszynach przy założeniu, że czasy trwania operacji mogą zmieniać się w pewnych przedziałach zmienności, warto byłoby zająć się modelami różniczkowymi [5,4] /dynamicznymi/ operacji. Modele te dokładniej modelują rzeczywistość, ponieważ pozwalają na zmianę ilości przydzielonych zasobów do operacji w trakcie ich wykonywania, co jest praktycznie niemożliwe przy modelach rozpatrywanych do tej pory. Pierwsze początki już zrobiono, np. w [44] podano pewien algorytm heurystyczny dla problemu z równoległymi maszynami o ograniczonych przepustowościach, dla stałej globalnej ilości zasobów i jednakowymi momentami dostępności operacji. Z kolei w [18,21,16] rozpatrywano pewne problemy przy zmiennej w czasie globalnej ilości zasobów, różnych momentach dostępności operacji i przy założeniu nieograniczonej przepustowości maszyn.

Rezultaty badań problemów szeregowania zadań na maszynach przy wspomnianych dynamicznych modelach operacji mogą mieć szczególnie istotne zastosowanie w elastycznie automatyzowanych procesach produkcyjnych.

LITERATURA

[1] Błażewicz J.: Problemy optymalizacji kombinatorycznej - złożoność obliczeniowa, algorytmy aproksymacyjne, PWN, Warszawa 1986.

[2] Błażewicz J., Cellary W., Słowiński R., Węglarz J.: Scheduling under Resource Constraints: Deterministic Models, J.C. Baltzer, Basel, 1986.

[3] Bubnicki Z.: Problemy optymalnego sterowania kompleksami operacji, Prace konferencji nt.: Problemy automatyki i informatyki, tom: Informatyka i automatyzacja kompleksowa, Gliwice, 1973.

[4] Bubnicki Z.: Optimal control of complex of operations with random parameters. Podstawy Sterowania, No 1, 1971, ss. 3-10.

[5] Burkov V.N.: Optimal project control, 4th Congress of IFAC, Technical Session, 35, 1969, ss. 46-57.

[6] Campbell H.G., Dudek R.A., Smith M.L.: A heuristic algorithm for the n job, m machine sequencing problem. Mgmt Sci 16, 1970.

[7] Cellary W.: Rozdział zasobów systemów komputerowych - próby podejścia globalnego, Rozprawy Nr 136 , Wydawnictwo Uczelniane Politechniki Poznańskiej, Poznań 1981.

[8] Elmaghraby S.E.: Activity networks, J. Wiley and Sons, New York,197

[9] Fulkerson D.R.: A network flow computation for project cost curves, Manag. Sci. 7, 1961, ss. 167-178.

[10] Garey M.R., Johnson D.S.: Complexity results for multiprocessor scheduling under resource constraints. SIAM J. on Computing 4. 1975, pp. 397-411.

[11] Garey M.R., Johnson D.S., Sethi R.: The complexity of flow-shop and job-shop scheduling. Math.Oper.Res. 1, 1976, ss. 117-129.

[12] Gonzalez T., Sahni S.: Flowshop and jobshop schedules: complexity and approximation. Oper. Res. 26, 1978, ss. 36-52.

[13] Grabowski J., Skubalska E., Smutnicki Cz.: On flow shop scheduling with release and due dates to minimize maximum lateness. J. Opl.Res. Soc. vol. 34, No 7, 1983, ss. 615-620.

[14] Grabowski J., Janiak A.: Job-shop scheduling with resource-time models of operations. Europ. Journal of Oper. Res. 28, 1987, ss. 58-73

[15] Janiak A.: Time-optimal Control in a Single Machine Problem with Resource Constraints. Automatica vol. 22, No 6, 1986, ss. 745-747.

[16] Janiak A., Stankiewicz A.: On Time-Optimal Control of a Sequence of Activities under Time-Variable Resource. IEEE Trans. on Aut. Control 1988 /w druku/.

[17] Janiak A.: General flow-shop scheduling with resource constraints. Internat. Journal of Production Research, 1988 /w druku/.

[18] Janiak A., Stankiewicz A.: The equivalence of local and global time-optimal control of a complex of operations. Internat. Journal of Control. vol. 38, No 6, 1983, ss. 1149-1165.

[19] Janiak A.: One - machine scheduling with allocation of continuously-divisible resource and with no precedence constraints. Kybernetika, vol. 23, No 4, 1987, ss. 289-293.

[20] Janiak A.: Some problems in flow-shop sequencing. Modelling, Simulation and Control, C, vol. 13, No 2, 1988, ss. 1-11 /w druku/.

[21] Janiak A.: Time-Optimal Control of a Sequence of Projects of Activities. Syst. Anal. Model. Simul. Vol. 4, No 1, 1987, ss. 43-52.

[22] Janiak A.: Minimization of the Maximum Tardiness in One-Machine Scheduling Problem Subject to Precedence and Resource Constraints. Syst. Anal. Model. Simul. vol. 4, No 6, 1987, ss. 549-556.

[23] Janiak A.: On a single machine sequencing to minimize the maximum job cost subject to resource and precedence constraints. Archiwum Aut. i Telem. Tom XXXI, Zeszyt 4, 1986, ss. 415-417.

[24] Janiak A.: Single machine sequencing with linears models of jobs subject to precedence constraints. Archiwum Aut. i Telem. 1988 /w druku

[25] Janiak A., Grabowski J.: Optimizacja problemy posledowatielnosti opieracij s raspriedielenijem riesursow w diskrietnych proizwodstwiennych sistiemach, Prace IBS PAN, 59, 1980, ss. 129-138.

[26] Johnson S.M.: Optimal two - and three - stage production schedules with setup times included, Nav. Res. Log. Quart. No 1, 1954.

[27] Karp R.M.: Reducibility among combinatorial problems, w: Complexity of Computer Computations. Plenum Press. 1972. ss. 85-103.

[28] Kelley, Jr., J.E., Critical path planning and scheduling; Mathematical basis, Oper. Res. 9, 1961, ss. 296-320.

[29] Kowalowski H./praca zbiorowa pod kierunkiem/: Automatyzacja dyskretnych procesów przemysłowych, WNT, Warszawa 1984.

[30] Lageweg B.J., Lenstra J.K., Rinnooy Kan A.H.G.: A general baunding Scheme for the permutation flow-shop problem. Oper.Res. vol. 26, No 1, 1978, ss. 53-67.

[31] Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G.: Recent developments in deterministic sequencing and scheduling: a survey. [w:] M.A.H. Depster, J.K. Lenstra, A.H.G. Rinnooy Kan /eds/, Deterministic and Stochastic Scheduling, Reidel, Dordrecht, 1982, ss. 35-73.

[32] Nowicki E., Zdrzałka S.: Dwumaszynowy problem przepływowy ze zmiennymi czasami wykonywania zadań. Zeszyty Nauk.AGH w Krakowie, Automatyka 39, 1985, ss. 161-169.

[33] Popadimitriou C.H., Kanellakis P.C.: Flowshop scheduling with limited temporary storage. J. Assoc. Comput. Mach. 27, 1980, ss.533-549.

[34] Phillips S., Dessouky M.I.: Solving the project time/cost tradeoff problem using the minimal cut concept. Management Science vol. 24, No 4, 1977, ss. 393-400.

[35] Piechler J. Ein Beitrag zum Reihenfolgeproblem. Unternehmensforschung 4, 1960, ss. 138-142.

[36] Reddi S.S., Ramamoorthy C.V.: On the flow-shop sequencing problem wait in process. Oper. Res. Quart. 23, 1972, ss. 323-331.

[37] Słowiński R.: Algorytmy sterowania rozdziałem zasobów różnych kategorii w kompleksie operacji. Rozprawy Nr 114, Wyd.Uczel.Polit.Pozn. Poznań, 1980.

[38] Turner S., Booth D.: Comparision of heuristics for flow shop sequencing. OMEGA The Int. Jl of Mgmt Sci. vol. 15, No 1, 1987, ss.75-85.

[39] Tuzikow A.W.: O dwuchkritierialnoj zadacze tieorii rospisanij s ucztom izmienienija dlitielnostiej obsłużiwanija. Żurnał wyczislitielnoj matiematiki i matiematiczeskoj fiziki,24,1984, ss. 1585-1590.

[40] Van Wassenhove L.N., Baker K.R.: A bicriterion approach to time/cost trade-offs in sequencing. Europ. J. Oper.Res.1,1982, ss. 48-54.

[41] Vickson R.G.: Choosing the job sequence and processing time to minimize total processing plus flow cost on a single machine. Oper. Res. 28, 1980, str. 1155-1167.

[42] Węglarz J.: Minimalno-czasowe sterowanie rozdziałem zadań i zasobów w kompleksie operacji w warunkach deterministycznych. Rozprawy Nr 78, Wyd. Uczel. Polit. Pozn., Poznań 1976.

[43] Węglarz J.: Sterowanie w systemach typu kompleks operacji, PWN, Warszawa, 1981.

[44] Węglarz J.: Project scheduling with discrete and continuous resources. IEEE Trans. SMC-9, 1979, ss. 644-650.

Recenzent: Prof.dr hab.inż.J.Węglarz

Wpłynęło do Redakcji do 1988-04-30.

ПРОБЛЕМА КОНВЕЙЕРНОГО ТИПА С ЛИНЕЙНЫМИ МОДЕЛЯМИ ОПЕРАЦИЙ

Р е з ю м е

В статье дан обзор проблем распределения ресурсов и проблем распис[...] ний . Детально изложены результаты полученные для класоической проблем[...] конвейерного типа. Затем эта проблема обобщена на случай с линейными мо[...] дями операций. Подробно расмотрена двухмашинная проблема конвейерного т[...] Полученные результаты обобщены на многомашинный случай.

PERMUTATION FLOW-SHOP PROBLEM WITH LINEAR OPERATION MODELS

S u m m a r y

In the paper the actual state of the resource allocation problems and sequencing ones is presented. The results obtained for the classical permutation flow-shop problem are recapitulated in detail. Next this problem is generalized on the case with linear operation models. Two-machine flow-shop problem is considered in detail and results obtained are generalized to the multimachine case.