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

Eugeniusz Nowicki
Politechnika Wrocławska
Instytut Cybernetyki Technicznej

ANALIZA ALGORYTMÓW APROKSYMACYJNYCH DLA M-MASZYNOWEGO PROBLEMU PRZEPŁYWOWEGO ZE ZMIENNYMI CZASAMI WYKONYWANIA ZADAŃ

Streszczenie. W pracy rozważa się m-maszynowy problem przepływowy, w którym czasy wykonywania zadań można zmieniać w pewnym przedziale. Zmiana czasu wykonywania zadania powoduje zmianę kosztu wykonywania, o którym zakłada się, że jest liniową, nierosnącą funkcją czasu wykonywania. Stawia się problem minimalizacji łącznego kosztu będącego sumą dwóch składników: kosztu liniowo zależnego od momentu zakończenia wykonania wszystkich zadań oraz sumarycznego kosztu wykonywania poszczególnych zadań. Do rozwiązania problemu proponuje się trzy algorytmy aproksymacyjne. Przeprowadza się szczegółową analizę eksperymentalną tych algorytmów.

1. Wstęp

Przedstawiana praca jest jedną z wielu prób wzbogacania rezultatów otrzymanych w "klasycznej" teorii szeregowania zadań na maszynach. Próby te polegają na rozszerzaniu zbadanych już problemów o różne dodatkowe elementy, rozwiązywaniu ich i w konsekwencji zbliżaniu się do zaatakowania problemów występujących w realnie istniejących dyskretnych systemach produkcyjnych. Przykładowo przyjmuje się, że do wykonania zadania (operacji) potrzebnych jest wiele rodzajów zasobów odnawialnych, nieodnawialnych i podwójnie ograniczonych, zasoby są podzielne w sposób ciągły lub dyskretny, występują różnego rodzaju magazyny o różnej pojemności. Ze względu na bogactwo literatury trudno jest w tym miejscu dokonać nawet krótkiej prezentacji poszczególnych kierunków badań. Należy tu jednak stwierdzić, że jedno z ogólniejszych sformułowań zostało podane po raz pierwszy przez Węglarza np. w [16].

W jednym z kierunków badań problemów szeregowania zakłada się, że czas wykonywania zadania nie jest stały, ale można go zmieniać w sposób ciągły w pewnym przedziale. Konsekwencją tej możliwości jest pojawienie się kosztu wykonywania zadania jako nierosnącej, liniowej funkcji jego czasu wykonywania i dalej, dodatkowego wskaźnika jakości uszeregowania reprezentowanego

przez sumaryczny koszt wykonywania poszczególnych zadań. W połączeniu z istniejącym kryterium rozpatrywanego problemu szeregowania otrzymujemy problem dwukryterialny. Często problem ten rozwiązuje się przez wprowadzenie kryterium globalnego, będącego kombinacją liniową tych kryteriów z odpowiednimi wagami. Reasumując, istotą omawianego kierunku badań jest przyjęcie modelu zadania takiego jak w problemach PERT/koszt i pojawienie się nowych zmiennych decyzyjnych (czasy wykonywania zadań). Badania nad tak sformułowanym modelem zostały zapoczątkowane w pracach [14],[15]. Prace te, jak i szereg następnych, dotyczyły przede wszystkim różnych jednomaszynowych problemów szeregowania. Problemy wielomaszynowe były badane, między innymi, w [5] oraz [11]. W [5] rozpatrywano gniazdowe zagadnienie kolejnościowe (ang. job-shop problem) z kryterium typu: termin zakończenia wykonywania wszystkich zadań. W odróżnieniu od omawianych tu prac, sumarycznego kosztu wykonywania zadań nie wprowadzono do kryterium globalnego, ale potraktowano jako ograniczenie. Zaproponowany algorytm, oparty na metodzie podziału i ograniczeń, znajdował rozwiązanie optymalne w "rozsądnym" czasie tylko dla problemów o niewielkim rozmiarze (liczba zadań i maszyn nie większa niż 6). Z kolei w [11] zaproponowano dwa algorytmy aproksymacyjne łącznie z analizą najgorszego przypadku dla dwumaszynowego problemu przepływowego. Wyczerpujący przegląd rezultatów otrzymanych w literaturze do roku 1987 i dotyczących omawianego kierunku badań, przedstawiono w pracy przeglądowej [12].

Niniejsza praca jest kontynuacją prac [11],[8] i dotyczy m-maszynowego permutacyjnego problemu przepływowego z kryterium typu termin zakończenia wykonywania zadań i z modelem zadania przedstawionym powyżej. Na bazie, przedstawionego w [8], ogólnego schematu algorytmu aproksymacyjnego rozwiązującego ten problem, konstruuje się trzy algorytmy aproksymacyjne. Algorytmy te wykorzystują różne algorytmy heurystyczne wyznaczające przybliżone rozwiązanie w "klasycznych" problemach przepływowych. Następnie przeprowadza się szczegółową analizę eksperymentalną zaproponowanych algorytmów. Wyniki tej analizy potwierdzają ich praktyczną przydatność do rozwiązywania omawianego problemu.

Praca była częściowo finansowana przez RP.I.02 "Teoria sterowania i optymalizacji układów dynamicznych i procesów dyskretnych".

2. Sformułowanie problemu i podstawowe własności

M-maszynowy permutacyjny problem przepływowy można sformułować następująco:

Dany jest zbiór zadań $J=\{1,2,\ldots,n\}$ oraz zbiór maszyn $M=\{1,2,\ldots,m\}$. Wykonanie zadania j polega na zrealizowaniu kolejno pewnych czynności na maszynach $1,2,\ldots,m$. Zadanie j na maszynie i wykonuje się w czasie $p_{ij} \geq 0$, $i\in M, j\in J$. Przyjmujemy, że (i) każda maszyna w dowolnej chwili czasowej może wykonywać nie więcej niż jedno zadanie, (ii) każda maszyna wykonuje zadania bez przerwań, tzn. jeżeli maszyna i rozpocznie wykonywać zadanie j w chwili S_{ij}, to musi je wykonać w chwili $C_{ij}=S_{ij}+p_{ij}$, (iii) wszystkie zadania są gotowe do wykonywania w chwili zerowej, (iv) kolejność wykonywania zadań na wszystkich maszynach jest identyczna. Powyższe założenia, a w szczególności (ii), pozwalają jednoznacznie określić uszeregowanie przez zestaw $S_{ij}, i\in M, j\in J$. Jako kryterium uszeregowania przyjmujemy moment zakończenia wykonywania wszystkich zadań, tzn. wielkość $\max_{j\in J} C_{mj}$. Przedstawiony problem, za [4], będziemy notować przez $F||C_{max}$. Wprowadzimy teraz kilka oznaczeń ułatwiających dalszą prezentację pracy. Niech $P=(p_1,p_2,\ldots,p_m)$, gdzie $p_i=(p_{i1},p_{i2},\ldots,p_{in})^T$, $i\in M$ oraz $\pi=(\pi(1),\pi(2),\ldots,\pi(n))$ określa permutację elementów zbioru J, zaś Π – zbiór wszystkich takich permutacji. Oznaczmy przez $C_{max}(\pi,p_1,\ldots,p_m)$ minimalny czas zakończenia wykonywania wszystkich zadań przy spełnieniu założeń (i)...(iv), dla kolejności wykonywania określonej przez π oraz czasów wykonywania P. Zachodzi

$$C_{max}(\pi,p_1,\ldots,p_m) = \max_{1\leq j_1\leq j_2\leq\ldots\leq j_m\leq j_{m+1}=n} \sum_{i=1}^{m} \sum_{j=j_i}^{j_{i+1}} p_{i\pi(j)}. \qquad (1)$$

Uszeregowaniem określonym przez π oraz P będziemy nazywać uszeregowanie scharakteryzowane zestawem S_{ij}, $i\in M, j\in J$, który definiujemy następującymi zależnościami rekurencyjnymi: $S_{i\pi(j)}=\max(S_{i-1,\pi(j)}+p_{i-1,\pi(j)}, S_{i,\pi(j-1)}+p_{i,\pi(j-1)})$, $i=2,\ldots,m$, $j=2,\ldots,n$; $S_{1\pi(j)}=\Sigma_{l=1}^{j-1} p_{1\pi(l)}$, $j=2,\ldots,n$ oraz $S_{i\pi(1)}=0$, $i\in M$. Uszeregowanie to spełnia warunki (i)...(iv) oraz czas zakończenia wykonywania wszystkich zadań (długość uszeregowania) równa się $C_{max}(\pi,p_1,\ldots,p_m)$.

Jak już wspominaliśmy we wstępie, będziemy dopuszczać możliwość zmiany czasów wykonywania zadań na maszynach. Zakładamy, że p_{ij} może przyjmować dowolną wartość z przedziału $[l_{ij},u_{ij}]$, gdzie $0\leq l_{ij}\leq u_{ij}$. Przez

$$\mathcal{P} = \{P=(p_1,\ldots,p_m) : p_i=(p_{i1},\ldots,p_{in})^T,\ l_{ij}\leq p_{ij}\leq u_{ij},\ i\in M, j\in J\} \qquad (2)$$

oznaczmy zbiór wszystkich dopuszczalnych czasów wykonywania zadań P na maszynach. Dalej zakładamy, iż koszt wykonywania zadania j na maszynach w czasie p_{ij}, $i\in M$ jest równy $\Sigma_{i\in M} c_{ij}*(u_{ij}-p_{ij})$, gdzie $c_{ij}>0$ określa jedno-

stkowy koszt skrócenia czasu wykonywania tego zadania na maszynie i. Stąd, sumaryczny koszt ma postać

$$K_2(p_1,\dots,p_m) = \sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij}*(u_{ij}-p_{ij}) \qquad (3$$

i w konsekwencji uszeregowanie określone przez $\pi\in\Pi$ oraz $P\in\mathcal{P}$ może być oceniane przez dwa kryteria [1],[3]. Przyjmując, że $w>0$ jest jednostkowym kosztem pracy systemu, otrzymujemy jedno kryterium kosztowe postaci: $K(\pi,p_1,\dots,p_m)=K_1(\pi,p_1,\dots,p_m)+K_2(p_1,\dots,p_m)$, gdzie $K_1(\pi,p_1,\dots,p_m)=w*C_{max}(\pi,p_1,\dots,p_m)$. Ostatecznie rozważany problem formułujemy następująco

Znaleźć kolejność wykonywania zadań $\pi^*\in\Pi$ oraz czasy wykonywania zadań $P^*=(p_1^*,\dots,p_m^*)\in\mathcal{P}$ minimalizujące

$$K(\pi,p_1,\dots,p_m) = w*C_{max}(\pi,p_1,\dots,p_m) + \sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij}*(u_{ij}-p_{ij}), \qquad (4$$

przy ograniczeniach

$$\pi\in\Pi,\ P=(p_1,\dots,p_m)\in\mathcal{P}. \qquad (5$$

Pierwszy składnik $K_1(\pi,p_1,\dots p_m)$ minimalizowanego kosztu $K(\pi,p_1,\dots,p_m)$ jest funkcją niemalejącą ze względu na wybraną zmienną p_{ij} przy ustalonych pozostałych, a drugi składnik $K(\pi,p_1,\dots,p_m)$ odpowiednio funkcją nierosnącą. Wynika stąd, iż nie zmniejszając ogólności rozważań możemy ograniczyć się do $c_{ij}<w$, $i\in M, j\in J$. Rzeczywiście, jeżeli pewne c_{ij} jest większe lub równe w, to istnieje rozwiązanie optymalne (π^*,P^*) z $p_{ij}^*=u_{ij}$. Stąd, dla tych i,j dla których $c_{ij}\geq w$ możemy określić nową trójkę l'_{ij}, u'_{ij}, c'_{ij} taką, że $l'_{ij}=u'_{ij}=u_{ij}$, a c'_{ij} jest dowolną liczbą z przedziału $(0,w)$.

Tak jak już zauważono w [8], problem (4),(5) jest silnie NP-trudny. Wynika to z faktu, iż szczególny przypadek tego problemu z $l_{ij}=u_{ij}, i\in M, j\in J$ jest równoważny silnie NP-trudnemu problemowi $F||C_{max}$.

Łatwo pokazać, że

$$\min_{(p_1,\dots,p_m)\in\mathcal{P}} K(\pi,p_1,\dots,p_m) \geq C_{max}(\pi,p'_1,\dots,p'_m)\ ,\ \pi\in\Pi\ , \qquad (6$$

gdzie

$$p'_i = (p'_{i1},\dots,p'_{in})^T\ ;\ p'_{ij} = w*l_{ij} + c_{ij}*(u_{ij}-l_{ij})\ ,\ i\in M, j\in J. \qquad (7$$

Stąd

$$LB1 = \min_{\pi\in\Pi} C_{max}(\pi,p'_1,\dots,p'_m) \qquad (8$$

jest dolnym ograniczeniem minimalnego kosztu $K(\pi^*,p_1^*,\dots p_m^*)$.

Uwzględniając powyższe uwagi oraz potrzebę skonstruowania algorytmu rozwiązującego w "rozsądnym" czasie problem (4),(5) dla dużych rozmiarów ($n \geq 100$), we wspomnianej już pracy [8] zaproponowano następujący schemat algorytmu aproksymacyjnego.

Algorytm H

Krok 1. Wyznacz $\pi^H \in \Pi$ minimalizujące $C_{max}(\pi, p'_1, \ldots, p'_m)$, stosując pewien algorytm aproksymacyjny H' o złożoności wielomianowej rozwiązujący problem $F||C_{max}$ dla czasów wykonywania zadań określonych zestawem $(p'_1, \ldots, p'_m)$ wg. wzoru (7).

Krok 2. Wyznacz $(p_1^H, \ldots, p_m^H) \in \mathcal{P}$ minimalizujące $K(\pi^H, p_1, \ldots p_m)$ przy ograniczeniach $(p_1, \ldots p_m) \in \mathcal{P}$.

Otrzymany zestaw $\pi^H, p_1^H, \ldots, p_m^H$ należy potraktować jako przybliżone rozwiązanie problemu (4),(5). W następnym rozdziale dokonamy przeglądu istniejących algorytmów aproksymacyjnych dla m-maszynowego problemu przepływowego i zaproponujemy trzy konkretne algorytmy dla badanego zagadnienia (4),(5).

3. Algorytmy aproksymacyjne i dolne ograniczenia

W literaturze podano wiele algorytmów aproksymacyjnych o złożoności wielomianowej do rozwiązania problemów typu: $F||C_{max}$. Jednakże, bazując na wynikach badań eksperymentalnych, przedstawionych między innymi w [2],[13], tylko trzy z nich; algorytm Campbella i in. [1], Dannenbringa [2] oraz Nawaza i in. [7], wydają się być interesujące do naszych celów. Algorytmy te charakteryzują się stosunkowo niewielkim średnim błędem względnym między produkowanym rozwiązaniem a rozwiązaniem optymalnym – w sensie wartości funkcji celu (C_{max}). Umożliwia to przyjęcie jako permutacji π^H permutacji, która jest "dobrym" przybliżonym rozwiązaniem problemu: $\min_{\pi \in \Pi} C_{max}(\pi, p'_1, \ldots, p'_m)$. Omówimy teraz szczegółowo wspomniane algorytmy rozwiązujące problem $F||C_{max}$ z czasami wykonywania zadań określonymi zestawem $(p_1, \ldots, p_m)$.

Algorytm Campbella i in. bazuje na m-1 dwumaszynowych problemach przepływowych typu: $F2||C_{max}$ z czasami wykonywania zadań $a^k = (a_1^k, \ldots, a_n^k)^T$ na maszynie 1 oraz czasami $b^k = (b_1^k, \ldots, b_n^k)^T$ na maszynie 2, gdzie $a^k = \sum_{i=1}^{k} p_i$, $b^k = \sum_{i=m-k+1}^{m} p_i$, $k = 1, 2, \ldots, m-1$. Produkowana przez niego permutacja $\pi^C \in \Pi$ jest rozwiązaniem następującego zadania

$$\min_{\pi \in \{\pi^1, \pi^2, \ldots, \pi^{m-1}\}} C_{max}(\pi, p_1, \ldots, p_m), \tag{9}$$

gdzie

$$\pi^k \in \Pi,\ C_{max}(\pi^k, a^k, b^k) = \min_{\pi \in \Pi} C_{max}(\pi, a^k, b^k)\ ,\ k=1,2,\ldots,m-1. \qquad (11)$$

Do rozwiązania problemu $F2||C_{max}$ można zastosować znany algorytm Johnsona złożoności $O(n\log n)$, [6]. Stąd złożoność algorytmu Campbella i in. jest rzędu $O(m^2n+mn\log n)$.

Algorytm Dannenbringa bazuje tylko na jednym dwumaszynowym problemie $F2||C_{max}$ z czasami wykonywania zadań $a=(a_1,\ldots,a_n)^T$ na pierwszej i $b=(b_1,\ldots,b_n)^T$ na drugiej maszynie, gdzie $a=\Sigma_{i=1}^m (m-i+1)p_i$, $b=\Sigma_{i=1}^m ip_i$. Produkowana przez niego permutacja $\pi^D \in \Pi$ spełnia $C_{max}(\pi^D,a,b) = \min_{\pi \in \Pi} C_{max}(\pi,a,b)$. Stąd złożoność algorytmu Dannenbringa jest rzędu $O(mn+n\log n)$. Istnieją dwie modyfikacje tego algorytmu (patrz [2]), których nie omawiamy.

Algorytm Nawaza i in. w pierwszym kroku wyznacza permutację $\pi'=(\pi'(1), \pi'(2),\ldots,(\pi'(n))$ zbioru J spełniającą $a_{\pi'(1)} \geq a_{\pi'(2)} \geq \ldots \geq a_{\pi'(n)}$, gdzie $a_j = \Sigma_{i=1}^m p_{ij}$, $j \in J$ oraz jako jednoelementową permutację π przyjmuje $(\pi'(1))$. Następnie wykonuje n-1 dalszych kroków. W k-tym kroku, k=2,3,...,n, bazując na znalezionej w poprzednim kroku k-1 elementowej permutacji $\pi=(\pi(1), \pi(2),\ldots,\pi(k-1))$ oraz elemencie $\pi'(k)$ przegląda k następujących permutacji k-elementowych: $(\pi(1),\ldots,\pi(k-1),\pi'(k))$, $(\pi(1),\ldots,\pi(k-2),\pi'(k),\pi(k-1))$, ..., $(\pi'(k),\pi(1),\ldots,\pi(k-1))$. Wśród tych permutacji znajduje permutację, dla której czas zakończenia wykonywania zadań ze zbioru $\{\pi'(1),\ldots,\pi'(k)\}$ jest minimalny i oznacza ją przez π. Otrzymana po wykonaniu n-tego kroku permutacja π jest traktowana jako wyjściowa permutacja π^N produkowana przez algorytm Nawaza i in.. Złożoność obliczeniowa jest rzędu $O(mn^3)$.
Dla przedstawianych algorytmów nie były znane, do niedawna, żadne rezultaty dotyczące analizy najgorszego przypadku. Ostatnio w [10] pokazano, że współczynnik najgorszego przypadku dla algorytmu Campbella i in. jest równy $\lceil m/2 \rceil$, a dla algorytmu Dannenbringa nie mniejszy niż $\lceil m/2 \rceil$. Analiza najgorszego przypadku algorytmu Nawaza i in. jest dalej otwartym problemem.

Uwzględniając wcześniejsze uwagi, proponujemy trzy następujące algorytmy aproksymacyjne do rozwiązania problemu (4),(5)

<u>Algorytm HC</u> : algorytm H; jako H' przyjąć algorytm Campbella i in..

<u>Algorytm HD</u> : algorytm H; jako H' przyjąć algorytm Dannenbringa.

<u>Algorytm HN</u> : algorytm H; jako H' przyjąć algorytm Nawaza i in..

Realizację kroku 2 powyższych algorytmów oparto na standardowym algorytmie, opisanym w [3], przeznaczonym do wyznaczania tzw. krzywej kosztu projektu w problemach PERT/koszt. W algorytmie tym zmodyfikowano warunek stopu w ten sposób, że wyznaczanie krzywej kosztu projektu kończy się, gdy jej

nachylenie jest większe lub równe w. Dokonano także innych modyfikacji wynikających ze specyfiki ograniczeń kolejnościowych. Dokładny opis wszystkich modyfikacji pomijamy. Z powyższego wynika, iż w odróżnieniu od kroku 1 realizacja kroku 2 nie ma złożoności wielomianowej. Jednakże, istniejące programy, wyznaczające krzywą kosztu projektu wg. algorytmu z [3], rozwiązują w stosunkowo krótkim czasie problemy o bardzo wielkich rozmiarach. Dlatego też wydaje się, że proponowany sposób wykonania kroku 2 będzie dość efektywny - tym bardziej, że wprowadzone modyfikacje zmniejszają złożoność obliczeniową wykorzystywanego algorytmu.

Przejdziemy teraz do omówienia różnych postaci dolnych ograniczeń minimalnego kosztu $K(n^*,p_1^*,\ldots,p_m^*)$. Te dolne ograniczenia będą wykorzystane do analizy eksperymentalnej algorytmów HC,HD oraz HN. Ze względu na ograniczenia czasowe będą nas interesowały tylko takie, które mają złożoność wielomianową. Dlatego też, proponujemy relaksację pojemności maszyn przy wyznaczaniu LB1 wg. (8) i w konsekwencji następujące dolne ograniczenie

$$LB2 = \max_{1\le k\le m} c_{max}(k) , \tag{10}$$

gdzie przez $c_{max}(k)$ oznaczamy minimalną wartość funkcji celu dla problemu $1|r_j^k,q_j^k,prec.|C_{max}$ z danymi $r_j^k=\Sigma_{i=1}^{k-1} p'_{ij}$, $p_j^k=p'_{kj}$, $q_j^k=\Sigma_{i=k+1}^{m} p'_{ij}$, $j\in J$, $k\in M$. Problem notowany jako $1|r_j,q_j,prec.|C_{max}$ jest jednomaszynowym problemem szeregowania z podzielnymi zadaniami, z niezerowymi terminami gotowości r_j zadań do wykonywania, z tzw. końcówkami zadań q_j oraz z kryterium typu: czas zakończenia wykonywania wszystkich zadań. Do jego rozwiązania wykorzystano algorytmy z [9] o złożoności $O(n^2)$. Stąd, złożoność algorytmu wyznaczającego LB2 jest rzędu $O(n^2m)$.

Wykorzystując nierówność $C_{max}(\pi,p_1,\ldots,p_m)\ge \max_{1\le i\le m} \Sigma_{j=1}^{n} p_{ij}$ prawdziwą dla dowolnych $\pi\in\Pi$, $(p_1,\ldots,p_m)\in\mathcal{P}$ dostaniemy jeszcze inne dolne ograniczenie

$$LB3 = \min_{(p_1,\ldots,p_m)\in\mathcal{P}} [\, w \max_{1\le k\le m} (\sum_{j=1}^{n} p_{kj}) + \sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij}\cdot(u_{ij}-p_{ij}) \,]. \tag{11}$$

Łatwo podać algorytm o złożoności $O(mn\log n+m^2n)$ wyznaczający (11). Jego opis pomijamy.

4. Analiza eksperymentalna algorytmów

Algorytmy HC,HD oraz HN zostały zaimplementowane w języku FORTRAN i przetestowane na minikomputerze IBM PC/XT. Dla n=20,30,50,80,100 oraz m=

2,3,4,5 wygenerowano 10 przykładów konkretnych problemów (4),(5). Łącznie przeliczono 200 przykładów. Dla każdego przykładu oraz ustalonej pary (i,j), $i\in M, j\in J$, wylosowano trzy liczby α, β, γ przez generator o rozkładzie równomiernym, ze zbioru {1,2,...,50} i położono $l_{ij}=\alpha$, $u_{ij}=\alpha+\beta$, $c_{ij}=\gamma$. Przyjęto $w=25*m$. Wylosowane przykłady są bardzo niekorzystne dla działania testowanych algorytmów, ponieważ średnio czas wykonywania każdego zadania na każdej maszynie może być skrócony o połowę. Wybór takich właśnie danych, pozwolił zbadać zachowanie się algorytmów w najbardziej "niesprzyjających" warunkach. Dla konkretnego przykładu wyznaczano wg. algorytmów HC,HD,HN π^{α}, $p_1^{\alpha},\ldots,p_m^{\alpha}$ oraz $K(\pi^{\alpha},p_1^{\alpha},\ldots,p_m^{\alpha})$, α=HC,HD,HN. Następnie wyliczano LB2, LB3, LB=max(LB2,LB3) i $r^{\alpha}=((K(\pi^{\alpha},p_1^{\alpha},\ldots,p_m^{\alpha})-LB)/LB)*100\%$, α=HC,HD,HN. Zauważmy, że r^{α} jest górnym ograniczeniem błędu względnego ρ^{α} określonego wzorem: $\rho^{\alpha}=((K(\pi^{\alpha},p_1^{\alpha},\ldots,p_m^{\alpha})-K(\pi^{*},p_1^{*},\ldots,p_m^{*}))/K(\pi^{*},p_1^{*},\ldots,p_m^{*}))*100\%$. Wyniki obliczeń przedstawiono w Tabeli 1.

Tabela 1.

Wyniki eksperymentów numerycznych

n|m	RSC	RSD	RSN	CPUC	CPUD	CPUN
20|2	0.38	0.58	1.60	0.9	1.0	1.7
20|3	3.11	3.35	4.84	3.0	3.2	4.1
20|4	6.74	6.21	8.07	6.0	6.3	7.5
20|5	8.81	8.36	9.96	10.9	10.5	12.2
30|2	0.26	0.32	0.98	2.0	2.0	4.4
30|3	1.74	1.83	4.23	5.7	5.9	9.4
30|4	5.81	5.26	6.61	12.9	13.0	17.1
30|5	6.11	6.31	8.67	21.1	20.8	27.3
50|2	0.11	0.20	0.83	4.6	4.6	14.8
50|3	1.21	1.24	2.71	10.4	10.1	21.8
50|4	3.65	3.73	4.81	21.4	20.2	51.3
50|5	5.67	5.71	7.43	53.2	51.2	102.3
80|2	0.06	0.06	0.45	8.3	8.1	30.2
80|3	0.70	0.73	0.91	14.8	13.3	42.1
80|4	2.11	2.17	3.91	32.1	31.1	70.6
80|5	4.32	4.21	5.35	80.8	79.8	199.3
100|2	0.04	0.05	0.16	14.2	15.2	90.5
100|3	0.41	0.43	0.81	21.4	22.1	101.9
100|4	1.98	1.84	3.52	50.7	53.2	148.4
100|5	2.30	2.23	3.56	168.2	170.2	371.0

RSC,RSD,RSN - średnia arytmetyczna dziesięciu górnych ograniczeń odpowiednio r^{HC}, r^{HD}, r^{HN}; wyrażona w procentach.

CPUC,CPUD,CPUN - średnia arytmetyczna dziesięciu czasów obliczeń odpowiednio dla algorytmu HC,HD,HN; wyrażona w sec..

Z analizy Tab. 1. wynika, że:

(i) Średnia arytmetyczna RSC niewiele różni się od średniej RSD dla tych samych n i m.

(ii) Średnia arytmetyczna RSN jest prawie zawsze większa od średnich RSC i RSD dla tych samych n i m.

(iii) Przy wzroście n a ustalonym m średnie RSC, RSD, RSN maleją.

(iv) Przy wzroście m a ustalonym n średnie RSC, RSD, RSN rosną.

(v) Średnie czasy obliczeń CPUC, CPUD, dla tych samych n,m, są porównywalne i mniejsze niż CPUN.

Powyższe obserwacje wskazują na to, iż w grupie testowanych przykładów algoryt HN ma gorsze własności, ze względu na czas obliczeń, jak i na dokładność produkowanego rozwiązania, niż algorytmy HC i HD. Zauważmy dalej, że algorytmy HC i HD dla przykładów o dużych rozmiarach n=100 ,m=5, dostarczają rozwiązań, których błąd względny jest nie większy od 2.3%. Czas obliczeń jest w tym wypadku rzędu trzech minut. Wszystkie te uwagi wskazują na to, że algorytmy HN i HD mogą być przydatne w praktyce.

LITERATURA

[1] Campbell H.G., Dudek R.A., Smith M.L.: A heuristic algorithm for the n job m machine sequencing problem, Mgmt. Sci. 16, 1970, B630-637.

[2] Dannenbring D.G.: An evaluation of flow-shop sequencing heuristics, Mgmt. Sci. 23, 1977, 1174-1182.

[3] Elmaghraby S.E.: Activity networks, J. Wiley and Sons New York 1977.

[4] Graham R.L. et al.: Optimization and approximation in deterministic sequencing and scheduling: a survey, Ann. Discrete Math. 5, 1979, 287-326.

[5] Grabowski J., Janiak A.: Job-shop scheduling with resource-time models of operations, European J. Oper. Res. 28, 1986, 58-73.

[6] Johnson S.M.: Optimal two- and three-stage production schedules with setup time included, Naval Res. Log. Quart. 1, 1954, 61-68.

[7] Nawaz M., Enscore Jr.E.E., Ham I.: A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, OMEGA Int. J. of Mgmt. Sci. 11, 1983, 91-95.

[8] Nowicki E.: Minimalizacja kosztu w m-maszynowym problemie przepływowym ze zmiennymi czasami wykonywania zadań, Zeszyty Naukowe AGH Seria Automatyka 42, 1987, 161-168.

[9] Nowicki E., Smutnicki Cz.: On lower bounds on the minimum maximum lateness on one machine subject to release date, OPSEARCH 24, 1987, 106-110.

[10] Nowicki E., Smutnicki Cz.: Worst-case analysis of an approximation algorithm for flow-shop scheduling, Raport serii: Preprinty nr 6/88, Wydawnictwo Politechniki Wrocławskiej, 1988 (przesłane do Oper. Res. Lett.).

[11] Nowicki E., Zdrzałka S.: Two-machine flow shop scheduling problem wit controllable job processing times, European J. Oper. Res. 34, 1988, 208-220.
[12] Nowicki E., Zdrzałka S.: Sequencing problems with controllable job processing times, Raport serii: Preprinty nr. 36/87, Wydawnictwo Politechniki Wrocławskiej, 1987 (zaakceptowane do druku w Discrete Appl. Math.).
[13] Turner S., Booth D.: Comparison of heuristics for flow shop sequencing OMEGA Int. J. of Mgmt. Sci. 15, 1987, 75-78.
[14] Vickson R.G.: Choosing the job sequence and processing times to minimize total processing plus flow cost on a single machine, Oper. Res. 28, 1980, 1155-1167.
[15] Van Wassenhove L.N., Baker K.R.: A bicriterion approach to time/cost trade-offs in sequencing, European J. Oper. Res. 11, 1982, 48-54.
[16] Węglarz J.: New models and procedures for resource allocation problem Proc. 6th INTERNRT Congres, VDI Verlag, Dusseldorf, Vol. 2, 1979, 521-530.

Recenzent: Doc.dr h.inż.J.Kłamka

Wpłynęło do Redakcji do 1988-04-30

АНАЛИЗ АППРОКСИМАЦИОННЫХ АЛГОРИТМОВ В СИСТЕМЕ ПОТОЧНОГО ТИПА С ПОСЛЕДОВАТЕЛЬНЫМИ М - МАШИНАМИ И С УЧЁТОМ ИЗМЕНЕНИЯ ДЛИТЕЛЬНОСТИ ОБСЛУЖИВАНИЯ

Р е з ю м е

В статье представляется система поточного типа с последовательными - машинами , в которых очерёдность выполнения задач и их время выполнения являются переменными подлежащими выбору. Принимая , что стоимость выполнения задачи является линейной функцией её времени выполнения, ставится проблема минимизации : суммарной стоимости выполнения всех задач или стоимость , которая зависит от времени окончания всех задач. Для этой проблемы формулируются аппроксимационные алгоритмы и предоставляются результаты вычислительных экспериментов , проведённых для случайно избранных примеров.

ANALYSIS OF APPROXIMATION ALGORITHMS FOR M-MACHINE FLOW SHOP PROBLEM WITH VARIABLE JOB PROCESSING TIMES

S u m m a r y

The paper deals with a m-machine flow shop scheduling problem in which both the sequence of jobs and their processing times are decision variables. It is assumed that the cost of performing a job is a linear function of its processing time, and schedule cost to be minimized is the total processing cost plus maximum completion time cost. A heuristic methods for solving this problem are proposed. The results of numerical simulations for approximation algorithms are also given.