Józef Grabowski, Eugeniusz Nowicki,
Czesław Smutnicki

Instytut Cybernetyki Technicznej
Politechniki Wrocławskiej

METODA BLOKOWA W GNIAZDOWYCH ZAGADNIENIACH KOLEJNOŚCIOWYCH*

Streszczenie: Praca zawiera nowy algorytm oparty na metodzie podziału i ograniczeń dla zagadnienia kolejnościowego gniazdowego. Algorytm ten wykorzystuje podejście blokowe.

1. Wstęp

W pracy rozważane jest gniazdowe zagadnienie kolejnościowe /job-shop problem/. Zagadnienie to w ogólnym przypadku jest silnie NP-zupełne i zdaniem wielu autorów jest jednym z najtrudniejszych zagadnień teorii szeregowania. Do rozwiązania tego zagadnienia proponowane są najczęściej algorytmy przybliżone lub algorytmy dokładne /wyznaczające rozwiązania optymalne/ oparte na schemacie metody podziału i ograniczeń. Te ostatnie, ze względu na złożoność zagadnienia, umożliwiają wyznaczenie rozwiązania w "rozsądnym" czasie tylko dla zagadnień o niewielkich rozmiarach ≤ 50.

W prezentowanej pracy przedstawiono własności rozważanego zagadnienia, schemat algorytmu dokładnego opartego na tzw. podejściu blokowym oraz wyniki badań testowych. Podejście blokowe było z powodzeniem stosowane do rozwiązywania innych /prostszych/ silnie NP-zupełnych zagadnień szeregowania takich, jak: problemy jednomaszynowe, wielomaszynowe problemy przepływowe [4] itp. Zastosowanie tego podejścia do gniazdowego zagadnienia kolejnościowego umożliwia rozszerzenie zakresu stosowalności algorytmów dokładnych.

2. Sformułowanie problemu i podstawowe własności

Sformułujemy zagadnienie $J|\prec, m_j=1|C_{max}$ rozpatrywane w tej pracy. Dany jest zbiór zadań $J = \{1,2,\dots,n\}$ przeznaczonych do realizacji przy użyciu zbioru maszyn $M = \{1,2,\dots,m\}$. Każde zadanie składa się tylko z jednej operacji. Dlatego też dalej operację będziemy utożsamiać z zadaniem. Zadanie j jest wykonywane na maszynie $\mu_j \in M$ w czasie $p_j > 0, j \in J$. Każda maszyna może wykonywać co najwyżej jedno zadanie w dowolnej chwili czasowej. Graf relacji $R_0 = \prec$ jest acykliczny. Należy znaleźć terminy

* praca była częściowo finansowana przez RP.I.02 "Teoria sterowania i optymalizacja ciągłych układów dynamicznych i procesów dyskretnych"

$S_j \geq 0$ rozpoczęcia wykonywania zadań na poszczególnych maszynach przy spełnieniu powyższych ograniczeń tak, by minimalizować funkcję celu postaci $\min_{j \in J} (S_j + p_j)$.

Zauważmy, że zagadnienie $J||C_{max}$ jest szczególnym przypadkiem wyżej sformułowanego zagadnienia, takim że relacja R_o jest zbiorem rozłącznych łańcuchów reprezentujących zadania.

W dalszym ciągu wprowadzimy szereg pojęć i definicji, które umożliwią nam sformułowanie zagadnienia $J|\prec, m_j=1|C_{max}$ w postaci wygodnej do dalszych rozważań. Oznaczmy przez $J^k = \{j \in J: \mu_j = k\}$ zbiór zadań przeznaczonych do wykonywania na maszynie k-tej oraz przez n_k liczność tego zbioru, $k \in M$. Niech $\pi_k = (\pi_k(1), \pi_k(2), \ldots, \pi_k(n_k))$ będzie permutacją elementów zbioru J^k a Π^k niech będzie zbiorem wszystkich takich permutacji π_k, $k \in M$; permutacja π_k określa kolejność wykonywania zadań na maszynie k-tej. Dalej niech $\Pi = \Pi^1 \times \Pi^2 \times \ldots \times \Pi^m$ a $\pi = (\pi_1, \pi_2, \ldots, \pi_m)$ niech będzie dowolnym elementem tego zbioru; zauważmy, że π określa kolejność wykonywania zadań na wszystkich maszynach /niekoniecznie dopuszczalną ze względu na R_o/. Przyjmijmy, że $R, R_o \subset R \subset J \times J$ jest relacją acykliczną oraz $G_R = (J, R)$ jest grafem tej relacji. Dalej przez $G_{\pi,R} = (J, E_\pi \cup R)$ oznaczmy graf utworzony dla relacji R oraz $\pi \in \Pi$, gdzie: $E_\pi = \bigcup_{k=1}^{m} \bigcup_{i=1}^{n_k - 1} \{(\pi_k(i), \pi_k(i+1))\}$. Grafy G_R i $G_{\pi,R}$ posiadają obciążone wierzchołki, przy czym obciążenie wierzchołka j wynosi p_j, $j \in J$. Niech $\Pi_R = \{\pi \in \Pi : G_{\pi,R}$ - acykliczny$\}$ będzie zbiorem dopuszczalnych kolejności wykonywania zadań na wszystkich maszynach dla danej relacji R. Teraz rozważane w pracy zagadnienie można sformułować w następującej równoważnej postaci:

Znaleźć kolejność $\pi^* \in \Pi_R$ wykonywania zadań na wszystkich maszynach taką, że

$$C_{max}(\pi^*, R) = \min_{\pi \in \Pi_R} C_{max}(\pi, R), \qquad /1/$$

gdzie: $C_{max}(\pi, R)$ jest najdłuższą drogą w grafie $G_{\pi,R}$, zaś $R = R_o$.

W dalszym ciągu zdefiniujemy kluczowe pojęcie bloku zadań wykorzystywane w konstrukcji algorytmu. Niech π będzie dowolnym elementem Π_R. Przez $u = (u_1, u_2, \ldots, u_r)$ oznaczmy drogę krytyczną w grafie $G_{\pi,R}$, gdzie $u_j \in J$, $1 \leq j \leq r$, a $1 \leq r \leq n$ jest liczbą wierzchołków na tej drodze. Formalnie droga krytyczna u oraz inne pojęcia wprowadzone dalej będą zależeć od π oraz R, ale dla uproszczenia nie będziemy tego uwzględniać w notacji. Niech a_k, $k=1,\ldots,s+1$; b_k, $k=1,\ldots,s$ będą podciągami ciągu u takimi, że: /i/ $(u_1, u_2, \ldots, u_r) = (a_1, b_1, a_2, \ldots, a_s, b_s, a_{s+1})$, $|b_k| > 1$, $1 \leq k \leq s$, /ii/ wszystkie zadania z ciągu b_k są wykonywane na tej samej maszynie /oznaczmy ją przez l_k/, $1 \leq k \leq s$, /iii/ każdy podciąg b_k jest maksymalnym ciągiem spełniającym /ii/; podciąg $b_k = (u_p, u_{p+1}, \ldots, u_q)$ jest maksymalny jeśli $\mu_{u_{p-1}} \neq 1_k \neq \mu_{u_{q+1}}$ $(\mu_{u_0} \equiv 0, \mu_{u_{r+1}} \equiv 0)$. Jeżeli $s \geq 1$ to podciąg b_k będziemy nazywać k-tym blokiem w π. Zgodnie z powyższą definicją pewne podciągi a_k mogą być puste.

Zachodzi następująca własność:

Własność 1.

/i/ Każda para sąsiednich elementów podciągu a_k jest w relacji R, $1 \leq k \leq s+1$.

/ii/ Jeżeli podciąg a_k jest niepusty to ostatni element podciągu a_k i pierwszy element podciągu b_k są w relacji R, $1 \leq k \leq s$, oraz ostatni element podciągu b_{k-1} i pierwszy element podciągu a_k są w relacji R, $2 \leq k \leq s+1$.

/iii/ Jeżeli podciąg a_k jest pusty to ostatni element podciągu b_{k-1} i pierwszy element podciągu b_k są w relacji R, $2 \leq k \leq s$.

Prawdziwość własności 1 wynika z faktu, że jeżeli dwa kolejne elementy u_i, u_{i+1} ciągu $u = (u_1,\ldots,u_r)$ są takie, że $\mu_{u_i} \neq \mu_{u_{i+1}}$ to $(u_i,u_{i+1}) \in R$. Przejdziemy dalej do sformułowania algorytmu.

3. Algorytm

Algorytm rozwiązujący problem /1/ został oparty na metodzie podziału i ograniczeń. Dlatego też w kolejnych punktach omówimy istotne elementy tego algorytmu takie, jak: charakterystyka i sposób podziału węzła w drzewie rozwiązań, strategia przeglądania drzewa rozwiązań, sposób wyznaczania górnych i dolnych ograniczeń.

3.1. Charakterystyka i sposób podziału węzła

Przyjmujemy, że każdy węzeł w drzewie rozwiązań jest całkowicie scharakteryzowany przez relację R. W korzeniu $R = R_0$. Relacja R określa zbiór Π_R rozwiązań dopuszczalnych dla problemu /1/. W związku z tym możemy powiedzieć, że każdy węzeł określony przez R jest związany z zadaniem /1/ dla relacji R. Przedstawimy teraz zasadę podziału węzła R /tzn. węzła z relacją R/ na węzły będące bezpośrednimi jego następnikami w drzewie rozwiązań. Niech π będzie pewnym elementem ze zbioru Π_R a $u = (u_1,\ldots,u_r)$ odpowiadającą mu drogą krytyczną w grafie $G_{\pi,R}$ oraz podciąg $b_k = (b_{k,1},\ldots,b_{k,v_k})$ k-tym blokiem w π, $k=1,\ldots,s$. Jeżeli $s=0$ to węzeł R nie jest dzielony. W przeciwnym wypadku dzielimy węzeł R na $2\cdot\sum_{k=1}^{s} v_k - 3s$ węzłów scharakteryzowanych przez relację $R_{k,l}^{\alpha}$, gdzie

$$R^{-}_{1,l} = R \cup \bigcup_{\substack{j=1\\ j\neq l}}^{v_1-1} \{(b_{1,l},b_{1,j})\} \cup F^{-}_{1} \cup \bigcup_{j=2}^{s} (F^{+}_{j} \cup F^{-}_{j}), \quad l=2,\ldots,v_1-1,$$

$$R^{+}_{1,l} = R \cup \bigcup_{\substack{j=1\\ j\neq l}}^{v_1} \{(b_{1,j},b_{1,l})\}, \quad l=1,2,\ldots,v_1-1$$

$$R^{-}_{k,l} = R \cup F^{-}_{1} \cup \bigcup_{j=2}^{k-1} (F^{+}_{j} \cup F^{-}_{j}) \cup \bigcup_{\substack{j=1\\ j\neq l}}^{v_k} \{(b_{k,l},b_{k,j})\}, \quad l=2,\ldots,v_k, \; k=2,\ldots,s,$$

$$R^{+}_{k,l} = R \cup F^{-}_{1} \cup \bigcup_{j=2}^{k-1} (F^{+}_{j} \cup F^{-}_{j}) \cup F^{+}_{k} \cup \bigcup_{\substack{j=2\\ j\neq l}}^{v_k} \{(b_{k,j},b_{k,l})\}, \quad l=2,\ldots,v_k-1, \; k=2,\ldots,s,$$

oraz

$$F^{+}_{k} = \bigcup_{j=2}^{v_k} \{(b_{k,1},b_{k,j})\}, \quad F^{-}_{k} = \bigcup_{j=1}^{v_k-1} \{(b_{k,j},b_{k,v_k})\} \quad k=1,\ldots,s.$$

Zauważmy, że jeśli relacja $R^{\alpha}_{k,l}$ jest cykliczna to $\Pi_{R^{\alpha}_{k,l}} = \emptyset$, zatem węzeł odpowiadający tej relacji nie jest tworzony. Problem czy relacja $R^{\alpha}_{k,l}$ jest cykliczna, czy acykliczna można stosunkowo prosto rozstrzygnąć. W tym celu zaobserwujmy, że jeżeli relacja R jest acykliczna to relacja $R \cup F^{\alpha}_{k}$ jest też acykliczna dla dowolnego k oraz $\alpha \in \{+,-\}$ /na podstawie własności drogi w grafie $G_{\pi,R}$/. W związku z tym relacja $R^{+}_{k,l}$ jest acykliczna wtedy i tylko wtedy, gdy nie istnieje droga w grafie G_R pomiędzy wierzchołkiem $b_{k,l}$ a dowolnym z wierzchołków $b_{k,l+1},\ldots,b_{k,v_k}$. Z kolei relacja $R^{-}_{k,l}$ jest acykliczna wtedy i tylko wtedy, gdy nie istnieje droga w grafie G_R pomiędzy dowolnym z wierzchołków $b_{k,1},\ldots,b_{k,l-1}$ a wierzchołkiem $b_{k,l}$.

Twierdzenie 1.

Niech $R \subset J \times J$ będzie relacją acykliczną a $\pi \in \Pi_R$. Wtedy dla każdego β należącego do zbioru

$$\Pi_R \setminus \bigcup_{k=1}^{s} \Big(\bigcup_{l=z_k}^{v_k-1} \Pi_{R^{+}_{k,l}} \cup \bigcup_{l=2}^{\bar{v}_k} \Pi_{R^{-}_{k,l}} \Big) \qquad /2/$$

zachodzi $C_{max}(\beta,R) \geq C_{max}(\pi,R)$, gdzie $z_1=1$, $z_k=2$, $k>1$ oraz $\bar{v}_1=v_1-1$, $\bar{v}_k=v_k$, $k>1$.

Szkic dowodu: Zauważmy, że zbiór /2/ jest równy zbiorowi $\Pi_{R'}$, $R' = R \cup \bigcup_{k=1}^{s} (F^{+}_{k} \cup F^{-}_{k})$. Dowód tego faktu pomijamy. Dalej niech $\beta \in \Pi_{R'}$. W grafie $G_{\beta,R'}$ istnieje droga $u' = (u'_1,\ldots,u'_{s'})$ taka, że $u' = (a_1,b'_1,a_2,\ldots,a_s,b'_s,a_{s+1})$, gdzie a_k, b_k podciągi zdefiniowane dla drogi krytycznej u w grafie $G_{\pi,R}$ oraz podciąg b'_k zawiera co najmniej wszystkie elementy ze

zbioru $\{b_{k,1},\dots,b_{k,v_k}\}$. Fakt ten wynika z własności 1. Droga u' istnieje również w grafie $G_{\beta,R}$. Stąd $C_{max}(\beta,R) \geqslant \sum_{i=1}^{s'} p_{u'_i} \geqslant \sum_{i=1}^{s} p_{u_i} =$ $= C_{max}(\pi,R)$, co kończy dowód.
Powyższe twierdzenie stanowi podstawę przy wykazaniu zbieżności przedstawionego algorytmu. Mówi ono,że zbiór określony przez /2/ może być pominięty w podziale węzła R bowiem kolejności β należące do tego zbioru są "nie lepsze" niż kolejność π /w sensie wartości $C_{max}(\pi,R)$/. Z twierdzenia 1 wynika również, że jeśli s=0 to dla każdego $\beta \in \Pi_R$ zachodzi $C_{max}(\beta,R) \geqslant C_{max}(\pi,R)$, co uzasadnia niedzielenie węzła R. Z definicji $R^{\alpha}_{k,l}$ wynika, że zbiory $\Pi_{R^{\alpha}_{k,l}}$ są parami rozłączne. Natomiast z faktu, że zbiór /2/ jest niepusty /zawiera co najmniej π / wynika, że suma elementów wszystkich zbiorów $\Pi_{R^{\alpha}_{k,l}}$ jest mniejsza niż liczba elementów zbioru Π_R. Dodatkowo, jeśli węzeł $\bar{R}$ jest następnikiem węzła R /niekoniecznie bezpośrednim/ w drzewie rozwiązań to dla każdego $\gamma \in \Pi_{\bar{R}}$ $C_{max}(\gamma,\bar{R}) =$ $= C_{max}(\gamma,R)$, a w szczególności $C_{max}(\gamma,R) = C_{max}(\gamma,R_0)$.
Podane własności uzasadniają, że drzewo rozwiązań generowane przy użyciu opisanej zasady podziału węzła będzie skończone. Ponadto $C_{max}(\pi^*,R_0) =$ $= \min_{R \in \mathcal{R}} C_{max}(\pi^R,R_0)$, gdzie $\pi^R \in \Pi_R$ jest kolejnością na bazie której został podzielony węzeł R zaś $\mathcal{R}$ - zbiór wszystkich węzłów drzewa rozwiązań. Zmniejszenie liczby węzłów w drzewie rozwiązań jest możliwe przez odpowiedni wybór kolejności π^R oraz postaci dolnego ograniczenia w każdym węźle. W algorytmie przyjmujemy, że π^R jest wybierane wg pewnego algorytmu heurystycznego dla zagadnienia $J|\prec',m_j=1|C_{max}$, gdzie $\prec' = R$.

3.2. Strategia przeglądania drzewa rozwiązań

Uważamy, że węzeł R jest przeglądnięty /zamknięty/ jeśli węzeł R nie jest dzielony lub wartość dolnego ograniczenia jest nie mniejsza niż aktualna wartość górnego ograniczenia lub wszystkie bezpośrednie następniki tego węzła zostały przeglądnięte /zamknięte/. Z węzła R przechodzimy do takiego bezpośredniego, nie zamkniętego następnika tego węzła, dla którego wartość dolnego ograniczenia jest najmniejsza. Po zamknięciu węzła R przechodzimy do jego bezpośredniego poprzednika; algorytm kończy się po zamknięciu węzła R_0.

3.3. Górne ograniczenia

Wartość górnego ograniczenia UB jest wyznaczana w każdym węźle R drzewa rozwiązań wg zasady UB := min $[UB, C_{max}(\pi^R,R)]$. Jako początkową wartość UB przyjmujemy UB = ∞.

3.4. Dolne ograniczenia

W węźle scharakteryzowanym przez relację R wyznaczamy wielkości r_j, q_j, $j \in J$ spełniające

$$
\begin{aligned}
r_j &= \max\left\{c'_{max}(B_j \cap J^{\mu_j}),\ \max_{i \in B_j}(r_i + p_i)\right\} \\
q_j &= \max\left\{c_{max}(A_j \cap J^{\mu_j}),\ \max_{i \in A_j}(p_i + q_i)\right\}
\end{aligned}
\qquad /3/
$$

gdzie: $B_j = \{i \in J : (i,j) \in R\}$, $A_j = \{i \in J : (j,i) \in R\}$, $j \in J$; $c'_{max}(I)$, $c_{max}(I)$ - minimalna wartość funkcji celu dla konkretnych problemów zagadnień $1|r_j|C_{max}$ oraz odpowiednio $1|q_j|C_{max}$ z danymi $I\ (I \subset J)$, p_j, r_j, q_j, $j \in I$; $c'_{max}(\emptyset) = c_{max}(\emptyset) = 0$. Na bazie tych wielkości można skonstruować różne postacie dolnych ograniczeń LB(R) dla węzła R [5]. Szczegółowy przegląd różnych możliwych postaci dolnych ograniczeń dla tego problemu został zawarty w pracy [3]. Warto jeszcze zauważyć, że można podać przepis na wyznaczenie wartości r_j, q_j w węźle R na podstawie wartości r_j, q_j dla bezpośredniego poprzednika tego węzła. Złożoność obliczeniowa tego przepisu jest znacznie mniejsza od złożoności obliczeniowej algorytmu wyznaczającego wartości r_j, q_j bezpośrednio z zależności /3/.

Jeżeli węzeł $\tilde{R}$ będący bezpośrednim następnikiem węzła R w drzewie rozwiązań jest taki, że $\tilde{R} = R^-_{1,l}$ lub $\tilde{R} = R^+_{s,l}$, to można dla niego skonstruować dodatkowe dolne ograniczenie $LB(\tilde{R})$ w postaci:

$$LB(R^-_{1,l}) = C_{max}(\pi^R, R) + r_{b_{1,l}} - r_{b_{1,1}}, \quad l=2,\dots,v_1-1,$$

$$LB(R^+_{s,l}) = C_{max}(\pi^R, R) - r_{b_{1,1}} + \min_{1 \le j < v_1}\{r_{b_{1,j}}\} + q_{b_{s,l}} - q_{b_{s,v_s}}, \quad l=2,\dots,v_s-1$$

gdzie: $r_j, q_j, j \in J$ są wyznaczone dla węzła R. Dowód pomijamy.

4. Wyniki obliczeniowe

Algorytm został zaimplementowany w języku FORTRAN na m.c. ODRA 1325. Przy testowaniu algorytmu posłużyliśmy się znanymi przykładami z literatury, dla których uzyskiwane wyniki od lat stanowią pewien miernik jakości i efektywności algorytmów rozwiązywania problemów gniazdowych. Przykłady zaczerpnięto z pracy [5].

Na podanych przykładach były /i są/ testowane wszystkie nowo pojawiające się w literaturze algorytmy dotyczące zagadnień gniazdowych. Do testowania algorytmu wybrano z wm. pracy 3 przykłady oznaczone odpowiednio 13-04-04, 13-05-04, 36-06-06, gdzie x-y-z, x - liczba operacji, y - liczba zadań, z - liczba maszyn. Zrealizowany algorytm /G/ został porównany z 3 najlepszymi znanymi w literaturze algorytmami: /B/ z pracy [2], /R/ z pracy [5], /A/ z pracy [1]. Wyniki przedstawiono w tabeli.

Przykład	Algorytm	CPU	L	LB
13-04-04	/G/	< 1	3	4
	/B/	.012	2	8
	/R/	.27	8	-
	/A/	-	4	11
13-05-04	/G/	< 1	1	1
	/B/	.02	1	6
	/R/	.21	5	-
	/A/	-	1	3
36-06-06	/G/	16	8	8
	/B/	1.96	20	100
	/R/	2.83	62	-
	/A/	-	15	83

L - liczba węzłów wygenerowanych,
LB - liczba węzłów, w których wyliczano dolne ograniczenia,
\- - brak danych z literatury,
CPU - czas obliczeń /sec/ dla poszczególnych algorytmów na maszynach: ODRA 1325 - /G/, DEC 2060 - /B/, CYBER 73-28 - /R/, ODRA 1305 - /A/.

Z analizy tabeli wynika, że przedstawiony algorytm /G/ przegląda znacznie mniejszą ilość węzłów w drzewie rozwiązań niż wzięte do porównania algorytmy /B/, /R/, /A/. Wydaje się także, że porównując szybkości obliczeń m.c. ODRA 1325 z m.c. DEC 2060 i CYBER 73-28 algorytm /G/ znajduje rozwiązanie szybciej niż algorytmy /B/ i /R/. Badania algorytmu /G/ przeprowadzono również na dużej grupie przykładów o wielkości n = 100, dla których parametry wylosowano. Średnio rozwiązanie optymalne otrzymano po wygenerowaniu ok. 50 węzłów w drzewie rozwiązań. Na wygenerowanie jednego węzła potrzeba było średnio ok. 3s.

LITERATURA

[1] Adrabiński A.: Algorytmy szeregowania operacji w dyskretnych systemach produkcyjnych z maszynami równoległymi, Praca doktorska, Raport Serii Preprinty, nr 32/83, Instytut Cybernetyki Technicznej PWr, Wrocław 1983.

[2] Bouma P.W.; Job-Shop Scheduling: A Comparison of Three Enumeration Schemes in a Branch-and-Bound Approach , Master thesis, Erasmus University Rotterdam, Faculty of Economics, Department of Econometrics/Operations Research, 1982.

[3] Grabowski J., Nowicki E., Smutnicki C.; Nowe metody wyznaczania dolnych ograniczeń dla zagadnienia kolejnościowego gniazdowego , Zeszyty Naukowe AGH, Automatyka z. 39, nr 1064, 1985.

[4] Grabowski J., Skubalska E., Smutnicki C., On flow-shop scheduling with release and due dates to minimize maximum lateness , J. Opl. Res. Soc., vol. 34, nr 7, 1983.

[5] Rinnooy Kan A.H.G.; Machine Scheduling Problems. Classifications, Complexity and Computations, Nijhoff, The Hague, 1976.

Recenzent: Doc.dr h.inż.Jerzy Klamka

Wpłynęło do Redakcji do 1986.04.30

БЛОЧНЫЙ МЕТОД В ГНЕЗДОВЫХ ЗАДАЧАХ ЧЕРЕДОВАНИЯ

Р е з ю м е

В рассматриваемой статье предоставлена задача гнездовая чередования (job-shop problem). Задача эта в общем случае является сильно P - трудной. В работе представлены свойства рассматриваемой задачи, схему точного алгоритма, основанного на т.н. блочном подходе , методы построения нижних ограничений а также результаты вычислений. Блочный подход был с успехом применён к другим (более простым) сильно P- трудным задачам чередования , таким как однoприборные задачи. многомашинные потоковые задачи и др.

BLOCK METHOD IN JOB-SHOP PROBLEMS

S u m m a r y

In the paper the job-shop problem is concidered.The problem is strongly NP-complete in general case.The paper contains: some properties of the problem,scheme of enumerative algorithm based on block approach,various lover bounds and computational results.The block approach has been successfully applied to other /simpler/ storngly NP-complete scheduling problems for instance one-machine problems,flow-shop problems and so on. The use of above approach to the job -shop problem extends the range of applicability of the enumerative algorithmus to difficult scheduling problems.