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

Lech Jamroż
Politechnika Krakowska

PRZYBLIŻONY ALGORYTM DLA MINIMAKSOWEGO ZAGADNIENIA SZEREGOWANIA OPERACJI

Streszczenie. W artykule rozważane jest minimaksowe zagadnienie kolejnościowe typu gniazdowego. Do jego rozwiązania zaproponowano przybliżony algorytm oparty na schemacie metody podziału i ograniczeń. Reguła ograniczająca oraz reguła wyboru określona zostały na podstawie odpowiednio zdefiniowanego parametru.

1. Wstęp

Dany jest zbiór niepodzielnych operacji $N=\{n_1,n_2,\dots n_n\}$ oraz zbiór maszyn $M=\{m_1,m_2,\dots m_m\}$ różnych typów. Typ maszyn stanowi zbiór $K=\{1,2,\dots q\}$. Oznaczmy przez s_i, c_i oraz t_{ij} odpowiednio czasy rozpoczęcia, zakończenia oraz wykonania i-tej operacji na j-tej maszynie.
W zbiorze maszyn M istnieją rozłączne podzbiory maszyn $M^k \left(\bigcup_{k\in K} M^k = M \right)$ jednakowego typu stanowiące wyposażenia gniazda. W zbiorze operacji N istnieją rozłączne podzbiory operacji $N^k \left(\bigcup_{k\in K} N^k = N \right)$, które mogą być wykonane przy użyciu maszyn k-tego typu. Ponadto wymóg porządku technologicznego wykonywania operacji określony przez relację RT, $(i,j)\in RT$ sprawia, że operacja i-ta musi być zakończona przed rozpoczęciem operacji j-tej. Jeżeli natomiast oznaczymy przez r_i – najwcześniejszy wymagany termin rozpoczęcia realizacji operacji n_i ($s_i \geqslant r_i$), d_i – najpóźniejszy wymagany termin zakończenia wykonywania operacji n_i ($c_i \leqslant d_i$) to $E_i=\max[0, r_i-s_i]$ oraz $L_i=\max[0,c_i-d_i]$ będą odpowiednio przyśpieszeniem oraz opóźnieniem wykonywania operacji $n_i\in N$.

Z każdą operacją $n_i\in N$ oraz maszyną $m_l\in M$ związana jest funkcja $u: m_l=\mu(i)$ zwana przydziałem i-tej operacji do l-tej maszyny. Zbiór $\eta=\{\mu \mid \mu=\mu(1),\mu(2),\dots\mu(n), m_l=\mu(i)\in M^k, n_i\in N^k, k\in K\}$ nazywamy zbiorem możliwych przydziałów operacji do maszyn. Jeżeli $N_l=\{n_i \mid n_i\in N^k,\ \mu(i)=m_l\in M^k, k\in K\}$ będzie zbiorem operacji wykonywanych na maszynie m_l, to kolejność realizacji operacji

ze zbioru N_l na tej maszynie jest określona permutacją $\Pi_l=(\Pi_l(1),\Pi_l(2),\dots\Pi_l(L))$, $\Pi_l(i)\in N_l$, $L=\mathrm{card}(N_l)$, $l=\overline{1,m}$.
Niech dalej wektor permutacji $\Pi=(\Pi_1,\Pi_2,\dots\Pi_m)$ określa kolejność wykonywania operacji na maszynach ze zbioru M, wówczas zbiór $\Gamma=\{\Pi \mid \Pi=(\Pi_1,\Pi_2,\dots\Pi_m)\}$ będzie zbiorem możliwych permutacji Π.

Rozwiązanie problemu szeregowania polega na wyznaczeniu wartości zmiennych $s_{\mu(i)}$, $i=\overline{1,n}$ minimalizujących funkcję celu postaci

$$L_{max} = \min_i L_i \qquad /1.1/$$

przy ograniczeniach:

$$s_i + t_{i\mu(i)} \leqslant s_j \,, \quad (i,j)\in RT \qquad /1.2/$$

$$(\mu(i) = \mu(j)) \Longrightarrow (s_i + t_{i\mu(i)} \leqslant s_j) \text{ lub } (s_j + t_{j\mu(j)} \leqslant s_i) \qquad /1.3/$$

Funkcja /1.1/ przedstawia sobą maksymalne spóźnienie w wykonywaniu operacji. Ograniczenie /1.2/ reprezentuje wymaganą kolejność realizacji operacji zgodnie z porządkiem technologicznym. Warunek /1.3/ natomiast zapewnia, że operacje przeznaczone do wykonania na tej samej maszynie nie mogą być wykonywane jednocześnie, lecz muszą być uszeregowane.

2. Algorytm

2.1. Funkcja kary

Niech $s=(s_1,s_2,\dots s_n)$ będzie ciągiem najwcześniejszych chwil rozpoczęcia wykonywania operacji ze zbioru N. Ciąg $S=(S_1,S_2,\dots S_v,\dots\dots S_n)$, dla którego $S_v=\min\{S_w - t_{w\mu(w)} \mid (n_v,n_w)\in RT\}$, $S_n=s_n$ – jest ciągiem najpóźniejszych chwil rozpoczęcia wykonywania operacji z tego zbioru.
Jeżeli $s_v=S_v$, wówczas każde opóźnienie wykonania operacji n_v powoduje wydłużenie czasu zakończenia wykonania wszystkich operacji, natomiast dla $\mu(v)=\mu(w)$ zachodzi

$$s_v + t_{v\mu(v)} \leqslant S_w \text{ lub } s_w + t_{w\mu(w)} \leqslant S_v \qquad /2.1/$$

Oznaczmy przez $s^k=(s_1^k,s_2^k,\dots s_{n_k}^k)$ oraz przez $S^k=(S_1^k,S_2^k,\dots S_{n_k}^k)$ ciąg odpowiednio najwcześniejszych oraz najpóźniejszych chwil rozpoczęcia wykonywania operacji ze zbioru N^k, $k\in K$.
Zbiór $C^k=\{(n_i,n_j): s_j^k < S_i^k \cap s_i^k < S_j^k,\ \mu(i)=\mu(j)=m_l\in M^k, n_i,n_j\in N^k, k\in K\}$ określony dla k-tego typu maszyn nazywamy zbiorem operacji krytycznych

Warunkiem koniecznym na to, aby kolejne operacje ze zbioru C^k były wykonywane bez przestoju maszyn, jest, aby bezpośrednio przed uszeregowaniem zachodziło $s_j^k < c_i^k$ oraz $s_i^k < c_j^k$ /2.2/

Zdefiniujmy teraz funkcję związaną z uszeregowaniem operacji ze zbioru C^k. Funkcję tę określamy zależnością

$$Q_{ij}^k(s_i^k, S_j^k) = s_i^k + t_{i\mu(i)} - S_j^k \qquad /2.3/$$

i nazywamy funkcją kary za przekroczenie ograniczenia /1.3/.
Funkcja ta jest niesymetryczna (tj. $Q_{ij}^k \neq Q_{ji}^k$) i może przyjmować wartości dodatnie, ujemne oraz zero. Rozpatrzmy przypadek dodatniej wartości. Funkcja Q_{ij}^k jest dodatnia wtedy, gdy dla $\{n_i, n_j\} \in C^k$ zachodzi $s_j^k = S_j^k$. Sprawdźmy to. $Q_{ij}^k = s_i^k + t_{i\mu(i)} - S_j^k = c_i^k - s_j^k$, ponieważ $c_i^k > s_j^k$ więc $Q_{ij}^k > 0$. Podobnie jeżeli $s_i^k = S_i^k$, wówczas $Q_{ji}^k > 0$. Łatwo zauważyć, że jeżeli funkcja kary związana z uszeregowaniem operacji jest dodatnia, wtedy kolejność realizacji tych operacji wpływa na wartość funkcji celu. Ujemna lub równa zero, jeżeli takiego wpływu nie ma.

Otrzymywanie najkorzystniejszych wartości kar wymaga, aby dla każdej pary $\{n_i, n_j\} \in C^k$ zachodziło

$$Q_l^k = \min\{Q_{ij}^k, Q_{ji}^k\} \qquad /2.4/$$

przy czym $n_i = \Pi_l(u)$, $n_j = \Pi_l(u+1)$, $m_l = \mu(i) = \mu(j)$ oraz

$$s^k_{\Pi_l(u)} + t_{\Pi_l(u)\mu(\Pi_l(u))} \leq s^k_{\Pi_l(u+1)}$$

Określmy następującą zależność

$$\mathcal{H}_l^k = \max\{0, Q_l^k\}$$

której wartość jest przyrostem funkcji celu związanym z uszeregowaniem operacji $\{n_i, n_j\} \in C^k$ tak, aby $s_i^k + t_{i\mu(i)} \leq s_j^k$; wówczas

$$H^k(s^k) = \sum_{s>k} \mathcal{H}_l^s \qquad /2.5/$$

jest oszacowaniem funkcji celu przy założeniu nieograniczonej przepustowości maszyn ze zbioru M^s, $s>k$, $k \in K$.
Funkcja postaci /2.5/ jest niemalejącą funkcją swojego argumentu. Załóżmy, że $s^{*k} = (s_1^{*k}, s_2^{*k}, \dots, s_n^{*k})$ oraz $s^k = (s_1^k, s_2^k, \dots, s_n^k)$ - będą ciągiem chwil rozpoczęcia wykonywania operacji odpowiednio przed oraz po uszeregowaniu $\{n_v, n_w\} \in C^k$, zatem $s_v^{*k} + t_{v\mu(v)} > s_w^{*k}$ oraz $s_w^{*k} + t_{w\mu(w)} > s_v^{*k}$.

ze zbioru N_l na tej maszynie jest określona permutacją $\Pi_l=(\Pi_l(1),\Pi_l(2),\ldots\Pi_l(L)),\ \Pi_l(i)\in N_l, L=card(N_l)\ ,\ l=\overline{1,m}$.
Niech dalej wektor permutacji $\Pi=(\Pi_1,\Pi_2,\ldots\Pi_m)$ określa kolejność wykonywania operacji na maszynach ze zbioru M, wówczas zbiór $\Gamma=\{\Pi\,|\,\Pi=(\Pi_1,\Pi_2,\ldots\Pi_m)\}$ będzie zbiorem możliwych permutacji Π.

Rozwiązanie problemu szeregowania polega na wyznaczeniu wartości zmiennych $s_{\mu(i)}, i=\overline{1,n}$ minimalizujących funkcję celu postaci

$$L_{max}=\min_i L_i \qquad /1.1/$$

przy ograniczeniach:

$$s_i+t_{i\mu(i)}\leqslant s_j\ ,\quad (i,j)\in RT \qquad /1.2/$$

$$(\mu(i)=\mu(j))\Longrightarrow(s_i+t_{i\mu(i)}\leqslant s_j)\ \text{lub}\ (s_j+t_{j\mu(j)}\leqslant s_i) \qquad /1.3/$$

Funkcja /1.1/ przedstawia sobą maksymalne spóźnienie w wykonywaniu operacji. Ograniczenie /1.2/ reprezentuje wymaganą kolejność realizacji operacji zgodnie z porządkiem technologicznym. Warunek /1.3/ natomiast zapewnia, że operacje przeznaczone do wykonania na tej samej maszynie nie mogą być wykonywane jednocześnie, lecz muszą być uszeregowane.

2. Algorytm

2.1. Funkcja kary

Niech $s=(s_1,s_2,\ldots s_n)$ będzie ciągiem najwcześniejszych chwil rozpoczęcia wykonywania operacji ze zbioru N. Ciąg $S=(S_1,S_2,\ldots S_v,\ldots\ldots S_n)$, dla którego $S_v=\min\{S_w-t_{w\mu(w)}\,|\,(n_v,n_w)\in RT\}$, $S_n=s_n$ – jest ciągiem najpóźniejszych chwil rozpoczęcia wykonywania operacji z tego zbioru.
Jeżeli $s_v=S_v$, wówczas każde opóźnienie wykonania operacji n_v powoduje wydłużenie czasu zakończenia wykonania wszystkich operacji, natomiast dla $\mu(v)=\mu(w)$ zachodzi

$$s_v+t_{v\mu(v)}\leqslant S_w\quad \text{lub}\quad s_w+t_{w\mu(w)}\leqslant S_v \qquad /2.1/$$

Oznaczmy przez $s^k=(s_1^k,s_2^k,\ldots s_{n_k}^k)$ oraz przez $S^k=(S_1^k,S_2^k,\ldots S_{n_k}^k)$ ciąg odpowiednio najwcześniejszych oraz najpóźniejszych chwil rozpoczęcia wykonywania operacji ze zbioru $N^k, k\in K$.
Zbiór $O^k=\{(n_i,n_j): s_j^k<S_i^k\cap s_i^k<S_j^k,\ \mu(i)=\mu(j)=m_l\in M^k, n_i,n_j\in N^k, k\in K\}$ określony dla k-tego typu maszyn nazywamy zbiorem operacji krytycznych

Warunkiem koniecznym na to, aby kolejne operacje ze zbioru C^k były wykonywane bez przestoju maszyn, jest, aby bezpośrednio przed uszeregowaniem zachodziło $s_j^k < c_i^k$ oraz $s_i^k < c_j^k$ /2.2/

Zdefiniujmy teraz funkcję związaną z uszeregowaniem operacji ze zbioru C^k. Funkcję tę określamy zależnością

$$Q_{ij}^k(s_i^k, S_j^k) = s_i^k + t_{i\mu(i)} - S_j^k \qquad /2.3/$$

i nazywamy funkcją kary za przekroczenie ograniczenia /1.3/.
Funkcja ta jest niesymetryczna (tj. $Q_{ij}^k \neq Q_{ji}^k$) i może przyjmować wartości dodatnie, ujemne oraz zero. Rozpatrzmy przypadek dodatniej wartości. Funkcja Q_{ij}^k jest dodatnia wtedy, gdy dla $\{n_i, n_j\} \in C^k$ zachodzi $s_j^k = S_j^k$. Sprawdźmy to. $Q_{ij}^k = s_i^k + t_{i\mu(i)} - S_j^k = c_i^k - s_j^k$, ponieważ $c_i^k > s_j^k$ więc $Q_{ij}^k > 0$. Podobnie jeżeli $s_i^k = S_i^k$, wówczas $Q_{ji}^k > 0$. Łatwo zauważyć, że jeżeli funkcja kary związana z uszeregowaniem operacji jest dodatnia, wtedy kolejność realizacji tych operacji wpływa na wartość funkcji celu. Ujemna lub równa zero, jeżeli takiego wpływu nie ma.

Otrzymywanie najkorzystniejszych wartości kar wymaga, aby dla każdej pary $\{n_i, n_j\} \in C^k$ zachodziło

$$Q_i^k = \min\{Q_{ij}^k, Q_{ji}^k\} \qquad /2.4/$$

przy czym $n_i = \Pi_l(u)$, $n_j = \Pi_l(u+1)$, $m_l = \mu(i) = \mu(j)$ oraz

$$s_{\Pi_l(u)}^k + t_{\Pi_l(u)\mu(\Pi_l(u))} \leq s_{\Pi_l(u+1)}^k$$

Określmy następującą zależność

$$\mathcal{H}_i^k = \max\{0, Q_i^k\}$$

której wartość jest przyrostem funkcji celu związanym z uszeregowaniem operacji $\{n_i, n_j\} \in C^k$ tak, aby $s_i^k + t_{i\mu(i)} \leq s_j^k$; wówczas

$$H^k(s^k) = \sum_{s>k} \mathcal{H}_i^s \qquad /2.5/$$

jest oszacowaniem funkcji celu przy założeniu nieograniczonej przepustowości maszyn ze zbioru M^s, $s>k$, $k \in K$.
Funkcja postaci /2.5/ jest niemalejącą funkcją swojego argumentu. Załóżmy, że $s^{*k} = (s_1^{*k}, s_2^{*k}, \ldots, s_n^{*k})$ oraz $s^k = (s_1^k, s_2^k, \ldots, s_n^k)$ - będą ciągiem chwil rozpoczęcia wykonywania operacji odpowiednio przed oraz po uszeregowaniu $\{n_v, n_w\} \in C^k$, zatem $s_v^{*k} + t_{v\mu(v)} > s_w^{*k}$ oraz $s_w^{*k} + t_{w\mu(w)} > s_v^{*k}$.

Po uszeregowaniu tych operacji otrzymujemy

$$a/\ s_v^k + t_{v\mu(v)} \leqslant s_w^k \qquad b/\ s_w^k + t_{w\mu(w)} \leqslant s_v^k$$

a zatem $s_w^k > s_w^{\prime k}$ lub $s_v^k > s_v^{\prime k}$.

Rozpatrzmy przypadek a/; wówczas może zachodzić

$$i/\ s_v^k + t_{v\mu(v)} - S_w^k \leqslant 0 \qquad ii/\ s_v^k + t_{v\mu(v)} - S_w^k > 0$$

zarówno dla i/, jak ii/ $\sum_s \gamma_s|_v > 0$.

Własność tę wykorzystamy przy określeniu reguły ograniczającej.

2.2. Oszacowanie wartości funkcji celu.

Wprowadźmy dla każdej pary operacji $(n_i, n_j) \in C^k, k \in K$ liniową względem parametru h^k funkcję postaci

$$F_i^k(h_i^k) = Q_i^k + h_i^k H^k(s^k) \qquad /2.6/$$

i nazwijmy ją parametrycznym oszacowaniem funkcji celu.

Wyznaczmy teraz zakres zmian wartości parametru h_i^k.

Oznaczmy przez LB oraz UB wartości odpowiedniego dolnego oraz górnego oszacowania funkcji celu; wówczas

$$LB \leqslant Q_i^k + h_i^k H^k(s^k) \leqslant UB \qquad /2.7/$$

Ponieważ $H^k(s^k) \neq 0, k \in K$, to

$$\frac{LB - Q_i^k}{H^k(s^k)} \leqslant h_i^k \leqslant \frac{UB - Q_i^k}{H^k(s^k)} \qquad /2.8/$$

Zależność /2.8/ możemy zapisać teraz następująco

$$h_{i,L}^k \leqslant h_i^k \leqslant h_{i,U}^k \qquad /2.8/$$

przy czym

$$h_{i,L}^k = \frac{LB - Q_i^k}{H^k(s^k)}, \qquad h_{i,U}^k = \frac{UB - Q_i^k}{H^k(s^k)}.$$

2.3. Reguła ograniczająca i reguła wyboru.

Reguła ograniczająca określona zostanie na podstawie wartości parametru h^k reprezentującego poszczególne uszeregowania operacji ze zbioru C^k, $k \in K$. Rozważmy ustalenie kolejności operacji n_i, n_j względem operacji n_w $\{n_i, n_j, n_w\} \in C^k$. Powiemy, że uszeregowanie $n_w = \Pi_I(u), n_i = \Pi_I(u+1)$ jest nie lepsze

$n_v = \Pi_I(u)$, $n_j = \Pi_I(u+1)$, jeżeli zachodzi

$$\min\{|h^k_{j,U} - h^k_{i,U}|, |h^k_{j,L} - h^k_{i,L}|\} \geqslant |h^k_j - h^k_i| \qquad /2.10/$$

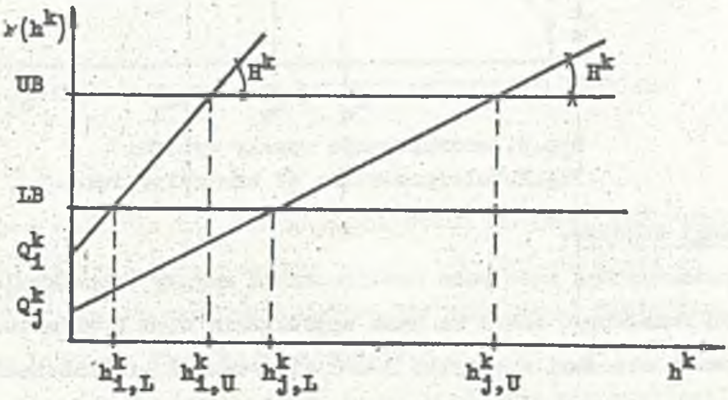
Niech h_H będzie odpowiednią wartością, dla której spełniona jest zależność

$$|h^k_j - h^k_i| \leqslant h_H \quad , \quad n_i, n_j \in N^k, k \in K \qquad /2.11/$$

wówczas regułę ograniczającą możemy zapisać w postaci

$$\min\{|h^k_{j,U} - h^k_{i,U}|, |h^k_{j,L} - h^k_{i,L}|\} \geqslant h_H \qquad /2.12/$$

Wartość h_H związaną z funkcją H^k można zawsze dobrać tak, aby zależność /2.11/ była spełniona. Poniższy rysunek jest geometryczną interpretacją reguły ograniczającej.



Rys.1. Interpretacja reguły ograniczającej.
Fig.1. Interpretation of exclusion rule.

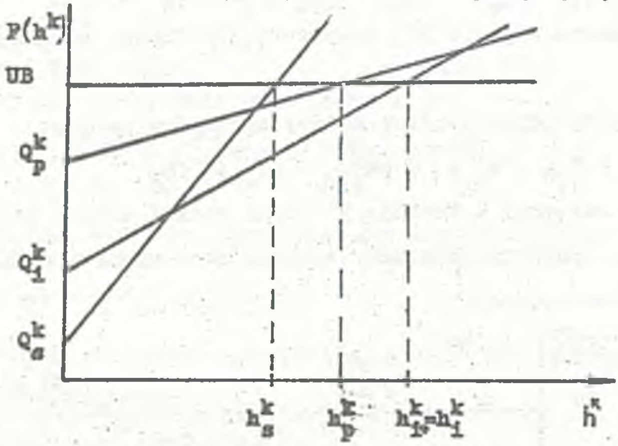
Ogólną zasadą przy określeniu reguły wyboru jest dążenie do kompromisowego wyznaczenia takiego uszeregowania, odpowiadająca któremu prosta związana z funkcją $F(\overline{h^k})$ posiada możliwie najmniejsze wartości Q^k_i oraz H^k, $k \in K$. Kompromis taki zapewnia wyznaczenie h^k wg. zależności

$$h^k_{i*} = \max_i h^k_{i,U} \qquad /2.13/$$

przy czym

$$h^k_{i,U} = \frac{UB - Q^k_i}{H^k(s^k)}$$

Poniższy rysunek jest geometryczną interpretacją reguły wyboru.



Rys.2.Interpretacja reguły wyboru.
Fig.2.Interpretation of branching rule.

3. Uwagi końcowe.

Przedstawione podejście konstruowania reguły ograniczającej i reguł wyboru rozszerzyć można na inne zagadnienia typu kombinatorycznego rozwiązywane metodami nie tylko B-a-B, ale również programowania dynamicznego. Podstawowym problemem byłoby określenie funkcji H^k. Dobierając odpowiednią wartość h_H, możemy tym samym zmieniać liczbę eliminowanych rozwiązań, co jest ważne w przypadku dużych problemów kombinatorycznych.

LITERATURA

[1] Biondi E.,Palermo D.C.:A heuristic approach to combinatorial optimization problems,Lecture Notes in Computer Science,Springer-Verlag 1983.

[2] Lageweg B.J.,Lenstra J.K.,Rinnooy Kan A.H.; Job-shop scheduling by implicit enumeration,Management Science,Vo24,No.4,1977.

[3] Maffioli F.;The complexity of combinatorial optimization algorithm and the chellenge of heuristic,In Christofides N./ed./,Combinatorial Optimization,John Wiley,1979.

Recenzent: Dr inż.G.Płoszajski

Wpłynęło do Redakcji do 1988-04-30.

ЭВРИСТИЧЕСКИЙ АЛГОРИТМ ДЛЯ МИНИМАКСНОЙ ПРОБЛЕМЫ ПОСЛЕДОВАТЕЛЬНОСТИ ЗАДАЧ

Р е з ю м е

В работе представлена математическая модель общего дискретного производственного процесса и алгоритмическо-эвристический метод , базирующий на идее методов ветвей и границ. Рассматриваемая проблема является одной из типичных комбинаторных проблем . Для её решения предложено правило ветвей и границ для множества допустимых решений на основе параметра. Правило ветвей и границ ограничено к множеству , от порядка которого зависит стоимость функции цели. Представлены также часто встречаемые в литературе методы решения этих проблем.

SUBOPTIMAL ALGORITHM FOR MIN-MAX JOBS SCHEDULING PROBLEM

S u m m a r y

This paper concerns the formal description of general discrete processes of the job-shop type and scheduling methods based on the concept of branch-and-bound method.The problem of minimizing the maximum lateness time is NP-complete.
Some properties of the problem are formulated. For computational effectivness the branch-and-bound rules defined within the set of feasible solutions are described.These rules are restricted to the set of operatins whose schedules influence the objective function. Additionaly the paper outlines some other approaches to this problem.