ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1984
Seria: AUTOMATYKA z. 74 Nr kol. 810

Marek Madejski
Eugeniusz Toczyłowski

Politechnika Warszawska

ALGORYTM DLA WIELOETAPOWEGO PROBLEMU PRZEZBRAJANIA LINII PRODUKCYJNEJ

Streszczenie. W pracy przedstawiono algorytm kolejnościowy dla wieloetapowego problemu minimalizacji kosztów przezbrajania linii produkcyjnej oparty na połączeniu metody podziału i ograniczeń z metodą programowania dynamicznego.

1. Model zadania przezbrajania

Podczas harmonogramowania produkcji wieloasortymentowej realizowanej partiami na pojedynczej linii produkcyjnej pojawia się problem ustalenia kolejności wykonywania poszczególnych partii produktów w każdym okresie objętym horyzontem harmonogramowania. Zmiana wersji produkowanych wyrobów powoduje powstawanie pewnych dodatkowych kosztów nazywanych kosztami przezbrajania, których wielkość zależy w ogólnym przypadku od wersji wyrobów. Problem minimalizacji kosztów przezbrajania w ustalonym przedziale czasu objętym harmonogramowaniem można sformułować następująco. Załóżmy, że na linii produkcyjnej można produkować wyroby o numerach wersji ze zbioru $N = \{1,2,\ldots,k\}$. Przypuśćmy, że horyzont harmonogramowania można podzielić na K etapów /np. tygodnie/ przy czym dla każdego etapu k, $1 \leq k \leq K$ znany jest zbiór wersji $N_k \subseteq N$ wyrobów jakie mają być produkowane na tym etapie. Możliwości przezbrajania linii opisane są za pomocą sieci $S = (N,E,P)$, gdzie N - jest zbiorem numerów wszystkich produkowanych wersji; $E \subseteq N \times N$ jest zbiorem łuków grafu (N,E) określającego relację dopuszczalności bezpośredniego przezbrojenia linii, tzn. $(i,j) \in E$ oznacza, że możliwa jest produkcja j-tej wersji po wersji i-tej po dokonaniu przezbrojenia; $P = [p_{ij}]_{n \times n}$ - macierz kosztów przezbrojenia, $p_{ij} \geq 0$ jest średnim kosztem przezbrajania z wersji i-tej na wersję j-tą. Oczywiście $p_{ij} < \infty$ tylko wtedy, gdy $(i,j) \in E$. Zakładamy, że koszty p_{ij} spełniają nierówność trójkąta $p_{il} \leq p_{ij} + p_{jl}$ dla każdego i,j,l.

Zazwyczaj graf (N,E) jest grafem pełnym. Dla ustalonego etapu k możliwości przezbrajania linii opisane są podsiecią $S_k = (N_k, E_k, P_k)$ sieci S, gdzie $E_k = (E \cap (N_k \times N_k))$, natomiast P_k jest macierzą powstałą z P przez wykreślenie wierszy i kolumn ze zbioru $N \setminus N_k$. Stan początkowy i końcowy linii może być swobodny lub ustalony. W ogólności zakładamy, że dane są zbiory wersji początkowych N_0 oraz zbiór wersji końcowych N_{K+1} /w przypadku ustalonych stanów początkowego i końcowego, N_0 i N_{K+1} są zbiorami jednoelementowymi/. Bez straty ogólności możemy założyć, że w każdym etapie wersja końcowa jest zarazem wersją początkową dla następnego etapu. Wprowadźmy zmienne decyzyjne

$$x_{ij}(k) = \begin{cases} 1 & \text{gdy w } k\text{-tym etapie } j\text{-ta wersja jest produkowana} \\ & \text{bezpośrednio po } i\text{-tej wersji;} \\ 0 & \text{w przeciwnym przypadku} \end{cases}$$

v_k - numer wersji produkowanej w k-tym etapie jako ostatnia /i zarazem jako pierwsza w (k+1)-szym etapie/.

Zadanie przezbrajania można sformułować jako problem minimalizacji

$$F = \sum_{k=1}^{K} \sum_{i \in N_k} \sum_{j \in N_k} p_{ij}\, x_{ij}(k) \qquad /1/$$

względem zmiennych $x_{ij}(k)$ oraz v_k, przy ograniczeniach

$$\sum_{j \in N_k} x_{ij}(k) = 1 \qquad i \in N_k \setminus \{v_k\} \cup \{v_{k-1}\}, \quad 1 \le k \le K \qquad /2/$$

$$\sum_{i \in N_k} x_{ij}(k) = 1 \qquad j \in N_k \setminus \{v_{k-1}\} \cup \{v_k\}; \quad 1 \le k \le K \qquad /3/$$

$$x_{ij}(k) = 0,1 \qquad \forall\, i,j,k \qquad /4/$$

$x(k)$ jest drogą w grafie (N_k, E_k) rozpoczynającą się od wierzchołka $v_{k-1} \in N_k$ i kończącą się w wierzchołku v_k. /5/

$$v_k \in N_k, \qquad 0 \le k \le K, \qquad /6/$$

$$v_K \in N_K \cap N_{K+1}$$

Funkcja celu /1/ jest sumaryczną wartością kosztów przezbrojeń w całym okresie harmonogramowania, ograniczenia /2/, /3/ zezwalają na produkcję każdej wersji tylko jeden raz w czasie jednego etapu, warunek /5/ eliminuje możliwość pojawienia się podcykli w rozwiązaniach spełniających /2/ i /3/. Powyższe zadanie jest NP-zupełne. W szczególnym przypadku, gdy K=1 oraz $v_0 = v_K$ zadanie to jest redukowane do zwykłego problemu komiwojażera a tym samym w ogólnym przypadku jest ono trudniejsze do rozwiązywania.

2. Dekompozycja problemu przezbrajania

Dla ustalonego etapu $1 \le k \le K$, oraz dla ustalonych wartości zmiennych v_{k-1} oraz v_k rozważmy następujące jednoetapowe zadanie kolejnościowe sparametryzowane przez v_{k-1} oraz v_k:

Minimalizuj

$$z_k = \sum_{i,j \in N_k} p_{ij}\, x_{ij}(k) \qquad /7/$$

$$\sum_{j \in N_k} x_{ij}(k) = 1 \qquad i \in N_k \setminus \{v_k\} \cup \{v_{k-1}\} \qquad /8/$$

$$\sum_{i \in N_k} x_{ij}(k) = 1 \qquad j \in N_k \setminus \{v_{k-1}\} \cup \{v_k\} \qquad /9/$$

$$x_{ij}(k) = 0,1 \qquad /10/$$

$x(k)$ jest drogą w grafie (N_k, E_k) od v_{k-1} do v_k /11/

Zadanie powyższe oznaczane będzie symbolem $T_k(v_{k-1}, v_k)$. Łatwo zauważyć, że zadanie $T_k(y,v)$ jest równoważne zadaniu komiwojażera. Wszystkie efektywne techniki algorytmu podziału i ograniczeń opracowano dla ogólnego problemu komiwojażera mogą więc być zastosowane do rozwiązywania zadania $T_k(y,v)$.

Problem przezbrajania można przedstawić równoważnie jako dwupoziomowy problem optymalizacji, przy czym nadrzędne zadanie jest postaci

$$\min_{v_1,\ldots,v_k} F = \sum_{k=1}^{K} z_k(v_{k-1}, v_k) \qquad /12/$$

przy ograniczeniach

$$v_{k-1},\ v_k \in N_k \qquad 1 \leq k \leq K$$

$$v_o \in N_o, \qquad v_k \in N_{K+1},$$

gdzie $z_k(v_{k-1}, v_k)$ jest optymalną wartością funkcji celu zadania $T_k(v_{k-1}, v_k)$. Liczność zmiennej stanu $v_k \in N_k$ jest zazwyczaj niewielka /zwykle w każdym okresie harmonogramowania produkuje się co najwyżej kilka wersji wyrobów/, stąd celowe jest rozwiązywanie problemu nadrzędnego /12/ metodą programowania dynamicznego. W ramach każdego etapu zadania parametryczne $T_k(v_{k-1}, v_k)$ rozwiązywane być mogą metodą podziału i ograniczeń.

Niezależne rozwiązywanie podproblemów /12/ i $T_k(y,v)$ nie jest najbardziej efektywne. Łączne wykorzystanie technik programowania dynamicznego i metody podziału i ograniczeń pozwala na efektywniejszy sondaż wierzchołków drzewa poszukiwań podczas rozwiązywania zadań $T_k(y,v)$ metodą podziału i ograniczeń a zarazem pozwala na rezygnację z obliczania wartości trajektorii optymalnych przechodzących przez niektóre stany pośrednie w metodzie programowania dynamicznego. Opisany niżej algorytm wykorzystuje powyższe uwagi.

3. Opis algorytmu

Algorytm dla wieloetapowego problemu przezbrajania linii produkcyjnej złożony jest z dwóch podstawowych części. Część pierwsza jest realizacją ograniczonej wersji programowania dynamicznego, w której wykorzystuje się oszacowania optymalnej wartości funkcji celu otrzymywane przez metodę podziału i ograniczeń. Część druga podrzędna, jest realizacją algorytmu dla zadania komiwojażera.

3.1. Ograniczona wersja programowania dynamicznego

Zadanie wieloetapowe /12/ może być rozwiązywane za pomocą programowania dynamicznego. W kolejnych krokach $k = K, K-1, \ldots, 1$ rozwiązywane jest zadanie od etapu k-tego do końca.

Załóżmy, że w k-tym etapie znane są wartości funkcji celu $F_k(v_k)$ dla optymalnych "ogonów" trajektorii rozpoczynających się od stanu v_k

$$F_k(v_k) = \min_{v_{k+1},\ldots,v_K} \sum_{t=k+1}^{K} z_t(v_{t-1}, v_t) \qquad /13/$$

W następnym kroku obliczane są wartości $F_{k-1}(v_{k-1})$ w funkcji stanu v_{k-1}. Dla ustalonego v_{k-1} przy pełnej wersji programowania dynamicznego obliczenie wartości $F_{k-1}(v_{k-1})$ wymaga rozwiązania

$$F_{k-1}(v_{k-1}) = \min_{v_k \in N_k} \left[z_k(v_{k-1}, v_k) + F(v_k) \right] \qquad /14/$$

co prowadzi do $|N_k|$ -krotnego rozwiązania zadania komiwojażera $T_k(v_{k-1}, v_k)$.

Przypuśćmy, że znane są oszacowania od dołu $\underline{z}_k(v_{k-1}, v_k)$ optymalnych wartości $z_k(v_{k-1}, v_k)$ /$\underline{z}_k$ wyliczane jest przez rozwiązywanie problemu złagodzonego/ oraz oszacowanie od góry $\bar{z}_k(v_{k-1}, v_k)$ optymalnej wartości $z_k(v_{k-1}, v_k)$ /wyliczane jako wartość funkcji celu dla dotychczas najlepszego dopuszczalnego rozwiązania zadania $T(v_{k-1}, v_k)$/. Dla ustalonej wartości v_{k-1} wprowadźmy oznaczenia oszacowań od dołu $\underline{Z}_k(v_{k-1}, v_k) = \underline{z}_k(v_{k-1}, v_k) + F(v_k)$ oraz od góry $\bar{F}_{k-1}(v_{k-1}) = \min_{v_k} [\bar{z}_k(v_{k-1}, v_k) + F(v_k)]$.

Wartości powyższych oszacowań można wykorzystać do obliczenia /14/ za pomocą następującej wersji programowania dynamicznego połączonego z metodą podziału i ograniczeń:

Schemat algorytmu

Krok 1. Oblicz $\underline{Z}_k(v_{k-1}, v_k) = \underline{z}_k(v_{k-1}, v_k) + F(v_k)$ dla wszystkich $v_k \in V_k$.

Krok 2. Uporządkuj strony pośrednie $v_k \in V_k$ według wzrastających wartości wskaźnika $\underline{Z}_k(v_{k-1}, v_k)$.

Krok 3. Traktując v_k jako parametr, rozwiązuj zadania $T_k(v_{k-1},v_k)$, $v_k \in V_k$ za pomocą metody podziału i ograniczeń w kolejności ustalonej w kroku 2. Aktualizowane wartości $\underline{Z}_k(v_{k-1},v_k)$ oraz najlepszego oszacowania od góry $\bar{F}_{k-1}(v_{k-1})$ wykorzystaj do sondażu /odrzucania/ wierzchołków w drzewie poszukiwań metody podziału i ograniczeń.

Łatwo zauważyć, że opisany wyżej algorytm pozwala na odrzucanie w metodzie podziału i ograniczeń nieobiecujących kierunków poszukiwań przez wykorzystanie stosunkowo silnych ograniczeń od dołu i góry uwzględniających wartości funkcji celu dla całych "ogonów" trajektorii optymalnych, a nie tylko dla zadań jednoetapowych $T_k(v_{k-1},v_k)$.

3.2. Rozwiązywanie zadania $T(v_{k-1},v_k)$

Zadanie komiwojażera, jako NP - zupełne, może być rozwiązywane za pomocą metody podziału i ograniczeń. Dla powyższego zadania opracowano wiele efektywnych technik wyznaczania oszacowań dolnych i górnych oraz podziału, wykorzystujących specyficzne właściwości problemu komiwojażera [1] - [5]. Najbardziej typowa relaksacja polega na sprowadzeniu zadania komiwojażera do zadania przydziału. Dla zadań o większym wymiarze relaksacja za pomocą zadania przydziału okazuje się niewystarczająco skuteczna. Szczególnie mocne oszacowania od dołu można uzyskać przez ograniczone relaksacje Lagrange'a wybranych ograniczeń problemu komiwojażera, nie spełnionych w rozwiązaniu zadania przydziału [1], [2]. Siła tych oszacowań jest taka, że z reguły liczba sondowanych wierzchołków drzewa poszukiwań jest znacznie mniejsza od wymiaru zadania. Koszty obliczania tych oszacowań są jednak stosunkowo duże i wynoszą $O(n^3)$ oraz $O(n^4)$. Dla zadań komiwojażera o niewielkim wymiarze /rzędu kilku lub kilkunastu/ stosowanie wyżej opisanych wyrafinowanych technik nie wydaje się być celowe ze względu na niewielkie ewentualne zyski pamięci i czasu wykonania, przy stosunkowo rozbudowanym oprogramowaniu.

4. Wyniki obliczeń

Opisany wyżej algorytm został opracowany w języku Fortran i uruchomiony na maszynie Odra 1325. Dokładniejszy opis realizacji tego algorytmu znajduje się w [6]. Procedury programu zapisano są na taśmie magnetycznej w wersji źródłowej. Taśma ta jest dostępna dla użytkowników i znajduje się w Instytucie Automatyki Politechniki Warszawskiej. Algorytm został przetestowany na serii 36 zestawów danych o współczynnikach generowanych losowo. Wartości współczynników kosztów przezbrojeń wyznaczano są według rozkładu jednostajnego z przedziału (0,100). Opracowana wersja algorytmu pozwala na uzyskanie rozwiązania dokładnego lub suboptymalnego z określoną przez użytkownika dokładnością [6]. Użytkownik może wybrać jedno z dwóch kryteriów suboptymalności:

/i/ ε_1 - względny błąd optymalności na każdym etapie nie przekracza wartości ε_1,

/ii/ ε_2 - bezwzględny błąd optymalności na każdym etapie nie przekracza wartości $\varepsilon_2 \cdot |N_k|\cdot$ średni koszt przezbrojenia.

Z przeprowadzonych obliczeń wybrano kilka charakteryzujących istotne właściwości algorytmu. W tablicy 1 przedstawiono przykładowe wyniki obliczeń algorytmu dla jednoetapowych żadań $T_k(v_{k-1}, v_k)$ przy zmiennym wymiarze N_k dla wariantu optymalnego oraz wariantów suboptymalnych $\varepsilon_1 = 0.1$ i $\varepsilon_2 = 0.02$.

Tablica 1

N_k	Wariant optymalny		Wariant suboptymalny			
	$\varepsilon = 0$		$\varepsilon_1 = 0,1$		$\varepsilon_2 = 0,02$	
	Czas	Funkcja celu	Czas	Funkcja celu	Czas	Funkcja celu
4	2	45,4	2	45,4	2	45,4
7	3	165,0	3	165,0	3	165,0
10	5	114,5	3	114,5	3	114,5
15	96	119,1	35	119,1	35	132,3
20	247	159,8	79	171,5	45	171,5

W tablicach 2 i 3 przedstawiono czasy obliczeń dla wieloetapowego problemu przezbrajania /ze swobodnym warunkiem końcowym/ w zależności od wariantu połączenia algorytmu programowania dynamicznego z algorytmem podziału i ograniczeń:

/i/ wersja N^* - niezależne stosowanie algorytmu programowania dynamicznego i metody podziału i ograniczeń /bez wykorzystywania oszacowań $\underline{Z}_k$ i $\overline{F}_{k-1}$/;

/ii/ wersja OPD - ograniczona wersja programowania dynamicznego, w której wykorzystuje się oszacowania $\underline{Z}_k$ i $\overline{F}_{k-1}$ do odrzucania we wzorze /14/ obliczeń dla zbędnych wariantów stanów pośrednich v_k /nie dających rozwiązania optymalnego/;

/iii/ wersja P - pełna wersja połączonego algorytmu programowania dynamicznego oraz metody podziału i ograniczeń /opisanego w pkt. 3/, w którym oszacowania $\underline{Z}_k$ i $\overline{F}_{k-1}$ wykorzystuje się do sondowania wszystkich wierzchołków drzewa rozwiązań metody podziału i ograni-

W tablicy 2 zamieszczono czasy obliczeń optymalnej wersji algorytmu dla problemu 7-etapowego z liczbą produkowanych wersji na każdym etapie $|N_k| = 7$.

Tablica 2

Wariant algorytmu	Czas dla kolejnych etapów [s]							Czas łączny
	1	2	3	4	5	6	7	
N	5	23	23	24	22	23	23	143
OPD	5	22	20	17	20	16	19	119
P	5	17	16	15	18	16	14	101

W tablicy 3 zamieszczono czasy obliczeń suboptymalnej wersji algorytmu / ε_1 = 0.1/ dla problemu 3-etapowego z liczbą wersji na każdym etapie $|N_k| = 10$.

Tablica 3

Wariant algorytmu	Czas dla kolejnych etapów [s]			Czas łączny
	1	2	3	
N	16	158	158	332
OPD	9	64	79	152
P	9	64	71	144

Dla problemów wieloetapowych krótki czas obliczeń dla pierwszego etapu wynika z wykorzystania informacji o stanie początkowym /numer wersji produkowanej jako pierwsza/. Połączony algorytm programowania dynamicznego oraz metody podzialu i ograniczeń daje niewielki zysk na czasie obliczeń w porównaniu z ograniczoną wersją programowania dynamicznego.

LITERATURA

[1] Balas E., Christofides N.: A new penalty method for the travelin salesman problem, IX Inst.Symp.Math.Progr. Budapeszt 1976

[2] Balas E., Christofides N.: The restricted lagrangean approach to the traveling salesman problem, Techn.Rep.Carnegie-Mellon University, 1979

[3] Christofides N.: Bounds for the traveling salesman problem, Opns. Res., 20, 1972, 1044-1055

[4] Held M., Karp R.M.: The traveling-salesman problem and minimum spanning trees, Opns.Res. 18, 1970, 1138-1162

[5] Held M., Karp R.M.: The traveling-salesman problem and minimum spanning trees, part II, Math.Progr. 1, 1971, 6-25

[6] Madejski M.: Wybrane zagadnienia harmonogramowania i sterowania dyspozytorskiego procesami montażu, Praca magisterska, Instytut Automatyki PW, 1983 /niepublikowana/.

Recenzent: Prof.dr inż.Henryk Kowalowski

Wpłynęło do Redakcji do 30.03.1984r.

АЛГОРИТМ ДЛЯ МНОГОЭТАПНОЙ ПРОБЛЕМЫ ПЕРЕНАЛАДКИ ПРОИЗВОДСТВЕННОЙ ЛИНИИ

Р е з ю м е

В работе дан алгоритм чередования для многоэтапной проблемы минимизации затрат переналадки производственной линии, основан на совмещение метода раздела и ограничений с методом динамического программирования.

AN ALGORITHM FOR MULTISTAGE PROBLEM OF PRODUCTION LINE RECONFIGURATION

S u m m a r y

In the paper a sequential algorithm for a problem of cost minimization for production line reconfiguration is presented. Bounds and branching algorithm is joined with dynamic programming method.