ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1977

Seria: AUTOMATYKA z. 37 Nr kol. 507

Jerzy MIKULSKI

MODYFIKACJA METODY OKREŚLANIA REALIZOWALNOŚCI FUNKCJI PRZEŁĄCZAJĄCEJ NA ELEMENTACH PROGOWYCH

Streszczenie. W pracy przedstawiono możliwość zastosowania algorytmu Czernikowa do sprawdzania warunku koniecznego i wystarczającego realizowalności funkcji przełączającej na elementach progowych. Wobec konieczności stosowania maszyn cyfrowych podano modyfikację tej metody. Całość jest ilustrowana przykładami.

Ważnym problemem przy poszukiwaniu struktury sieci logicznej elementów progowych jest znajomość warunku koniecznego i wystarczającego, by realizowana funkcja przełączająca była funkcją progową. Funkcję przełączającą nazywamy progową, gdy spełnia następujące warunki:

$$F(x_1,x_2,\dots x_n) = 1\ , \quad \text{gdy} \quad w_0 + \sum_{i=1}^{n} w_i x_i > 0\ ,$$

$$F(x_1,x_2,\dots x_n) = 0\ , \quad \text{gdy} \quad w_0 + \sum_{i=1}^{n} w_i x_i \leqslant 0\ , \tag{1}$$

gdzie operacje sumy i iloczynu są operacjami arytmetycznymi.

Strukturę elementu progowego określa wektor współczynników

$$W = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix},$$

gdzie liczba całkowita w_0 określa próg elementu, a w_i jest liczbą całkowitą przyporządkowaną i-temu wejściu, zwaną wagą tego wejścia.

Dla ułatwienia rozważań definicję funkcji progowej można zapisać inaczej definiując stale równy jeden współczynnik x_0 i wyrównując groty nierówności:

$$
\begin{aligned}
F(x_1,x_2,\dots x_n) &= 1, \quad \text{gdy} \quad \sum_{i=0}^{n} w_i x_i \geqslant 1, \\
F(x_1,x_2,\dots x_n) &= 0, \quad \text{gdy} \quad -\sum_{i=0}^{n} w_i x_i > 0.
\end{aligned}
\tag{2}
$$

Jeżeli tablica wejściowa jest macierzą o wymiarze $(2^n \times (n+1))$:

$$
X = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & \dots & 0 \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}
$$

Wartości funkcji przełączającej realizowanej przez element progowy o strukturze W są podawane w postaci wektora kolumnowego:

$$
F = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{2^n-1} \end{bmatrix}
\quad
\begin{aligned}
f_i &= 1 \quad \text{jeśli} \quad F_i(x_1,x_2,\dots x_n) = 1 \\
f_i &= 0 \quad \text{jeśli} \quad F_i(x_1,x_2,\dots x_n) = 0 \\
f_i &= \emptyset \quad \text{jeśli} \quad F_i(x_1,x_2,\dots x_n) = \emptyset
\end{aligned}
$$

Macierz diagonalna A uwzględnia znaki współczynników w nierównościach (2):

$$
A = \begin{bmatrix} a_0 & 0 & 0 & \dots & 0 \\ 0 & a_1 & 0 & \dots & 0 \\ 0 & 0 & a_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_{2^n-1} \end{bmatrix}
\quad
\begin{aligned}
a_i &= 1 \quad \text{jeśli} \quad f_i = 1 \\
a_i &= -1 \quad \text{jeśli} \quad f_i = 0 \\
a_i &= 0 \quad \text{jeśli} \quad f_i = \emptyset
\end{aligned}
$$

Po wprowadzeniu tych zapisów ważoną sumę z definicji funkcji progowej można zapisać w postaci macierzowej:

$$AXW \geqslant F \tag{3}$$

Przykład 1

$$F = \bar{x}_1 x_2 + x_1 \bar{x}_2 + x_1 \bar{x}_3$$

Układ nierówności (1) ma postać:

$$\begin{array}{lllll}
w_0 & & & & \leqslant 0 \\
w_0 & & & + w_3 & \leqslant 0 \\
w_0 & & + w_2 & & > 0 \\
w_0 & & + w_2 & + w_3 & > 0 \\
w_0 & + w_1 & & & > 0 \\
w_0 & + w_1 & & + w_3 & > 0 \\
w_0 & + w_1 & + w_2 & & > 0 \\
w_0 & + w_1 & + w_2 & + w_3 & \leqslant 0
\end{array}$$

Co daje się zapisać w macierzowej postaci (3):

$$AXW = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} > \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Warunkiem koniecznym i wystarczającym by funkcja przełączająca była funkcją progową, to znaczy warunkiem, by układ (3) miał rozwiązanie W, jest aby układ równań:

$$X^T A^T B = 0 \tag{4}$$

nie miał rozwiązań nieujemnych względem B. Dowód tego twierdzenia jak również opis procedury projektowania sieci logicznej w oparciu o to twierdzenie został przedstawiony w [3].

Przykład 2

Dla funkcji przełączającej z przykładu 1 układ równań (4) ma postać:

$$X^TA^TB = \begin{bmatrix} 1&1&1&1&1&1&1&1 \\ 0&0&0&0&1&1&1&1 \\ 0&0&1&1&0&0&1&1 \\ 0&1&0&1&0&1&0&1 \end{bmatrix} \begin{bmatrix} -1&0&0&0&0&0&0&0 \\ 0&-1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&-1 \end{bmatrix} \begin{bmatrix} B_o \\ B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Jest to więc następujący układ równań:

$$\begin{aligned} -B_o - B_1 + B_2 + B_3 + B_4 + B_5 + B_6 - B_7 &= 0 \\ B_4 + B_5 + B_6 - B_7 &= 0 \\ B_2 + B_3 + B_6 - B_7 &= 0 \\ -B_1 + B_3 + B_5 - B_7 &= 0 \end{aligned}$$

Układ równań (4) jest liniowym, jednorodnym układem (n+1) równań o 2^n niewiadomych. Szukanie rozwiązań takiego układu (większa liczba niewiadomych niż równań) jest zagadnieniem skomplikowanym. W myśl założeń cytowanego twierdzenia [3] poszukiwane są ewentualne nieujemne rozwiązania układu (4). W związku z tym do rozwiązania problemu można korzystać z algorytmu Czernikowa [1,2].

Algorytm ten polega na kolejnym przekształcaniu macierzy, która dla omawianego problemu ma postać:

$$\left[I \;\middle|\; AX \right] \tag{5}$$

Lewa część tablicy (do podziałowej kreski) jest jednostkową macierzą o wymiarze (2^n x 2^n), prawa zaś jest macierzą współczynników układu nierówności (3) o wymiarze (2^n x (n+1)).

Kolejne przekształcenia, nie zmieniające ilości kolumn w obu częściach macierzy (5) odbywają się w etapach. W każdym z nich można wyróżnić trzy operacje dokonywane na tablicy. Macierzą wyjściową, na której dokonujemy operacji w etapie pierwszym, jest macierz (5). Wspomniane etapy są jak następuje:

1. Wybieramy jako bazową kolumnę przekształcanej tablicy, kolumnę z prawej jej części, mającą dużo elementów zerowych (jeśli takie istnieją) i dużo elementów o przeciwnych znakach (jeśli takie są).

2. Przepisujemy do nowej tablicy bez zmiany te wiersze tablicy pierwotnej, na przecięciu których z kolumną bazową stoją zera.
3. Tworzymy liniowe kombinacje wszystkich tych par wierszy, których elementy w kolumnie bazowej przekształcanej macierzy mają przeciwne znaki. Kombinacje tworzymy z takimi dodatnimi współczynnikami, żeby elementy bazowej kolumny stały się równe zeru. Tak utworzone wiersze dopisujemy do wierszy otrzymanych w punkcie 2. Utworzoną nową macierz traktujemy jako macierz wyjściową do kolejnego etapu przekształceniowego, a więc do punktu 1 jak wyżej.

Po którymś z omówionych etapów może wystąpić jeden z dwóch następujących przypadków:

- w którejkolwiek kolumnie nie otrzymaliśmy pozycji zerowych, ani pozycji o przeciwnych znakach,
- w otrzymanej tablicy, w prawej jej części wszystkie kolumny są zerowe.

W pierwszym przypadku układ równań (4) ma tylko jedno nieujemne rozwiązanie - zerowe. W drugim przypadku wektory wiersze lewej części ostatniej tablicy i ich liniowe kombinacje z nieujemnymi współczynnikami tworzą nieujemne rozwiązanie układu (4).

<u>Przykład 3</u>

Tablica (5) dla funkcji z przykładu 1 ma postać:

$$[I \mid AX] = \left[\begin{array}{cccccccc|cccc}
1&0&0&0&0&0&0&0&-1&0&0&0\\
0&1&0&0&0&0&0&0&-1&0&0&-1\\
0&0&1&0&0&0&0&0&1&0&1&0\\
0&0&0&1&0&0&0&0&1&0&1&1\\
0&0&0&0&1&0&0&0&1&1&0&0\\
0&0&0&0&0&1&0&0&1&1&0&1\\
0&0&0&0&0&0&1&0&1&1&1&0\\
0&0&0&0&0&0&0&1&-1&-1&-1&-1
\end{array}\right]$$

Kolejne przekształcenia tej tablicy są następujące:

$$\left[\begin{array}{cccccccc|cccc}
1&0&0&0&0&0&0&0&-1&0&0&0\\
0&0&1&0&0&0&0&0&1&0&1&0\\
0&0&0&0&1&0&0&0&1&1&0&0\\
0&0&0&0&0&0&1&0&1&1&1&0\\
0&1&0&1&0&0&0&0&0&0&1&0\\
0&1&0&0&0&1&0&0&0&1&0&0\\
0&0&0&1&0&0&0&1&0&-1&0&0\\
0&0&0&0&0&1&0&1&0&0&-1&0
\end{array}\right]$$

$$\left[\begin{array}{cccccccc|cccc}
0&1&0&1&0&0&0&0&0&0&1&0\\
0&1&0&0&0&1&0&0&0&1&0&0\\
0&0&0&1&0&0&0&1&0&-1&0&0\\
0&0&0&0&0&1&0&1&0&0&-1&0\\
1&0&1&0&0&0&0&0&0&0&1&0\\
1&0&0&0&0&0&1&0&0&1&1&0\\
1&0&0&0&1&0&0&0&0&1&0&0
\end{array}\right]$$

$$\left[\begin{array}{cccccccc|cccc}
0&1&0&1&0&0&0&0&0&0&1&0\\
0&0&0&0&0&1&0&1&0&0&-1&0\\
1&0&1&0&0&0&0&0&0&0&1&0\\
0&1&0&1&0&1&0&1&0&0&0&0\\
1&0&0&1&1&0&0&1&0&0&0&0\\
1&0&0&1&0&0&1&1&0&0&1&0
\end{array}\right]$$

$$\left[\begin{array}{cccccccc|cccc}
0&1&0&1&0&1&0&1&0&0&0&0\\
1&0&0&1&1&0&0&1&0&0&0&0\\
1&0&1&0&0&1&0&1&0&0&0&0\\
1&0&0&1&0&1&1&2&0&0&0&0
\end{array}\right]$$

Układ równań ma więc cztery różne rozwiązania nieujemne, a tym samym układ nierówności w myśl twierdzenia [3] nie ma rozwiązania. Rozwiązania tego układu są następujące:

$$B_1^T = [0\ 1\ 0\ 1\ 0\ 1\ 0\ 1]$$

$$B_2^T = [1\ 0\ 0\ 1\ 1\ 0\ 0\ 1]$$

$$B_3^T = [1\ 0\ 1\ 0\ 0\ 1\ 0\ 1]$$

$$B_4^T = [1\ 0\ 0\ 1\ 0\ 1\ 1\ 2]$$

Ponieważ tablica I (5) mająca wymiary (2^n x 2^n) jest niewygodna do stosowania (zajmuje ona dużą część obszaru pamięci) zrodziła się koncepcja jej modyfikacji.

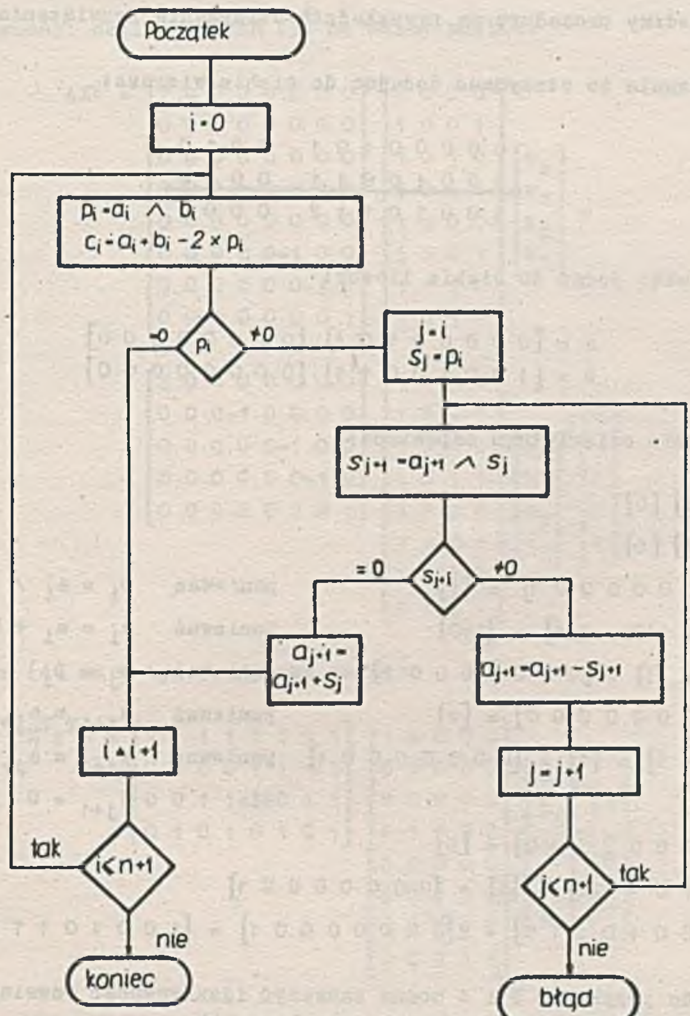
Tablica ta jest podobna w zapisie do zapisu kodu " 1 z n ". Opracowano więc zwięzły zapis tej tablicy przechodząc bezpośrednio do zapisu dziesiętnego przy przypisaniu odpowiednim pozycjom kolejnych wag. Ponieważ jednak w trakcie kolejnych przekształceń tablicy (5) można otrzymać jako rozwiązania nie tylko zera i jedynki ale również liczby większe (patrz przykład), więc zmodyfikowana tablica I musi posiadać (n+1) kolumn. Wynika to stąd, że tablica (5) posiadająca w prawej swej części (n+1) kolumn wymaga maksymalnie takiej ilości przekształceń a liczby na poszczególnych pozycjach macierzy powstałej z przekształcenia macierzy I mogą być równe(n+1) (chociaż - jak wynika z praktyki - nie jest to zjawisko zbyt częste).

W związku z powyższym i opisem samego algorytmu Czernikowa należało opracować procedurę dodawania liczb, które generalnie mogą mieć postać:

$$a = [a_0]\,[a_1]\,[a_2]\,\dots\,[a_n]\,,$$

$$b = [b_0]\,[b_1]\,[b_2]\,\dots\,[b_n]\,,$$

gdzie $[a_k]$ i $[b_k]$ są wektorami wierszowymi zer i jedynek, odpowiadającymi kolumnie (k-1)-krotnych rozwiązań.



Rys. 1. Schemat blokowy przekształceń przy zmodyfikowanej metodzie Czernikowa

W wyniku dodawania tych liczb otrzymuje się:

$$c = \sum_{k=0}^{n} 2^k \left[c_k\right],$$

gdzie $\left[c_k\right]$ jest wektorem wierszowym zer i jedynek.

Schemat zaproponowanej procedury przedstawia rys. 1.

Przykład 4

Prześledźmy procedurę na przykładzie otrzymania rozwiązania B_4 z przykładu 3.

Rozwiązanie to otrzymano dodając do siebie wiersze:

$$\begin{array}{cccccccccccccc} 0&0&0&0&0&1&0&1 & & 0&0&-1&0 \\ \underline{1}&\underline{0}&\underline{0}&\underline{1}&\underline{0}&\underline{0}&\underline{1}&\underline{1} & & \underline{0}&\underline{0}&\underline{1}&\underline{0} \\ 1&0&0&1&0&1&1&2 & & 0&0&0&0 \end{array}$$

Należało więc dodać do siebie liczby:

$$a = [0\ 0\ 0\ 0\ 0\ 1\ 0\ 1]\ [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$$
$$b = [1\ 0\ 0\ 1\ 0\ 0\ 1\ 1]\ [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$$

co w zapisie dziesiętnym odpowiada:

$a = [\ 5]\ [0]$

$b = [147]\ [0]$

$p_0 = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 1] = [1]$ ponieważ $p_i = a_i \wedge b_i$

$c_0 = [5 + 147 - 2.1] = [150]$ ponieważ $c_i = a_i + b_i - 2p_1$

$s_0 = p_0 = [1] = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]$ ponieważ $s_j = p_i$

$s_1 = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0] = [0]$ ponieważ $s_{j+1} = a_{j+1} \wedge s_j$

$a_1 = [0 + 1] = [1] = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]$ ponieważ $a_{j+1} = a_{j+1} + s_j$ dla $s_{j+1} = 0$

$p_1 = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0] = [0]$

$c_j = [1 + 0 - 2.0] = [1] = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]$

$c = [1\ 0\ 0\ 1\ 0\ 1\ 1\ 0] + 2[0\ 0\ 0\ 0\ 0\ 0\ 0\ 1] = [1\ 0\ 0\ 1\ 0\ 1\ 1\ 2]$

Porównując przykłady 3 i 4 można zauważyć identyczność rozwiązań.

Warto zauważyć, że metoda nie ogranicza się tylko do rozwiązywania funkcji przełączających całkowicie określonych. Bardzo istotnym i ważnym jest fakt, że w przypadku funkcji niecałkowicie określonych ulegają zmniejszeniu wymiary odpowiednich macierzy i przyspieszeniu ulega czas obliczeń. Ale zasada algorytmu pozostaje ta sama.

Przykład 5

$$F = \sum (0,7)_{x_1 x_2 x_3}$$

$$F = \prod (3,5,6)_{x_1 x_2 x_3}$$

Układ nierówności definicyjnych (3) ma wtedy postać:

$$AXW = \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&-1&0&0\\ 0&0&0&0&0&0&-1&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} 1&0&0&0\\ 1&0&0&1\\ 1&0&1&0\\ 1&0&1&1\\ 1&1&0&0\\ 1&1&0&1\\ 1&1&1&0\\ 1&1&1&1 \end{bmatrix} \begin{bmatrix} w_0\\ w_1\\ w_2\\ w_3 \end{bmatrix} =$$

$$= \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0\\ 0&0&0&0&0&-1&0&0\\ 0&0&0&0&0&0&-1&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} 1&0&0&0\\ 1&0&0&1\\ 1&0&1&0\\ 1&0&1&1\\ 1&1&0&0\\ 1&1&0&1\\ 1&1&1&0\\ 1&1&1&1 \end{bmatrix} \begin{bmatrix} w_0\\ w_1\\ w_2\\ w_3 \end{bmatrix} > \begin{bmatrix} 1\\ 0\\ 0\\ 0\\ 1 \end{bmatrix}$$

Układ równań (4) ma w tym przypadku postać:

$$X^T A^T B = \begin{bmatrix} 1&1&1&1&1&1&1&1\\ 0&0&0&0&1&1&1&1\\ 0&0&1&1&0&0&1&1\\ 0&1&0&1&0&1&0&1 \end{bmatrix} \begin{bmatrix} 1&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&-1&0&0&0\\ 0&0&0&0&0\\ 0&0&-1&0&0\\ 0&0&0&-1&0\\ 0&0&0&0&1 \end{bmatrix} \begin{bmatrix} B_0\\ B_3\\ B_5\\ B_6\\ B_7 \end{bmatrix} = \begin{bmatrix} 0\\ 0\\ 0\\ 0\\ 0 \end{bmatrix}$$

$$\begin{aligned} B_0 - B_3 - B_5 - B_6 + B_7 &= 0 \\ - B_5 - B_6 + B_7 &= 0 \\ - B_3 \quad\; - B_6 + B_7 &= 0 \\ - B_3 - B_5 \quad\; + B_7 &= 0 \end{aligned}$$

Układ ten ma nieujemne rozwiązanie:

$$B = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 2 \end{bmatrix}$$

Procedura przedstawiona została sprawdzona na maszynie cyfrowej GIER. Rozwiązywano problemy dla funkcji przełączających maksymalnie siedmiu argumentów wejściowych. W [4] określono pamięć potrzebną do przechowywania tablicy AX, która maksymalnie dla siedmiu argumentów wejściowych i trzeciego rzędu problemu [4] wynosi 1408 komórek. Należy pamiętać, że największego wymiaru pamięci wymaga jednak wykonanie procedury Czernikowa. W procedurze tej wymagana jest znajomość w każdej iteracji tablicy I. Wobec zmniejszenia w niej ilości kolumn procedurą modyfikującą, jej wymiary wynoszą $s \times (n+1+r)$, gdzie r określa rząd problemu [4], a s określa ilość powstałych par w kolejnej iteracji. Ponieważ jest bardzo trudno a priori określić s, gdyż to zależy od konkretnej funkcji, oszacowanie potrzebnego obszaru pamięci można przeprowadzić inaczej. Można określić maksymalną ilość par, które można zmieścić w pamięci. Ilość ta wynosi:

$$\frac{\text{pamięć operacyjna} - 1408}{n+1+r}$$

Powyższa liczba nie obejmuje par identycznych, które w trakcie obliczeń powinny być eliminowane. W przypadku omawianych obliczeń dla n = 7 ilość ta wynosiła 244. Niektóre więc przykłady nie mieściły się w pamięci operacyjnej maszyny cyfrowej GIER.

LITERATURA

[1] Czernikow S.N.: Liniejnyje nierawienstwa, 1968, Nauka.

[2] Czernikowa N.W.: Algorytm dla nachażdienija obszczej formuły nieatricatielnych reszenij sistiemy liniejnych uprawnienij, Żurnał wyczysłitielnoj matematiki i matematiczeskoj fizyki, 1964, t. 4, Nr 4, Nauka.

[3] Mikulski J., Klamka J.: Synteza minimalnej sieci złożonej z logicznych elementów progowych, ZN Politechniki Śląskiej "Automatyka",1974, Nr 27.

[4] Mikulski J., Klamka J.: Znak współczynnika wagowego wejścia przęgającego w sieci elementów progowych, ZN Politechniki Śląskiej "Automatyka", nr 37.

МОДИФИЦИРОВАННЫЙ МЕТОД ЧЕРНИКОВА

Р е з ю м е

В статье представлено применение алгоритма Черникова и проверка конечного и достаточного условия реализации переключающей функции на пороговых элементах. При применении вычислительной машины указано модификацию этого метода. Приводятся примеры указанного метода.

OF CZERNIKOW METHOD

S u m m a r y

The paper presents posibilities of application of a Czernikow algorithm in testing threshold function conditions. A modyfication of this algorithm is also presented. The method is illustrated by examples.