ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1975

Seria: AUTOMATYKA z. 32 Nr kol. 449

Jerzy RUTKOWSKI
Instytut Elektroniki

ALGORYTM ZNAJDOWANIA DRZEWA I MACIERZY OCZKOWEJ PRZY NARZUCONYCH OGRANICZENIACH

Streszczenie. W pracy podano algorytm pozwalający znaleźć drzewo gdy narzucone są nań pewne ograniczenia, wyznaczający jednocześnie macierz oczkową. Praca zawiera ogólny opis algorytmu, przedstawiony na przykładzie opis algorytmu krok po kroku oraz opis procedury realizującej algorytm, napisanej w języku ALGOL 1204.

1. WSTĘP

Często przy analizie obwodów zachodzi potrzeba znalezienia drzewa spełniającego pewne dodatkowe warunki.

Przykładowo:

- analiza obwodów metodą wektora stanu wymaga by gałęzie, w których znajdują się pojemności, należały do drzewa a gałęzie, w których znajdują się indukcyjności znajdowały się poza nim,
- analiza obwodów metodą iteracyjną jest efektywniejsza, jeśli do drzewa należą gałęzie o najmniejszych rezystancjach itp.

Powstaje zatem problem znalezienia drzewa jeśli narzucone są nań pewne ograniczenia. Następnym problemem występującym zazwyczaj przy analizie obwodów jest znalezienie układu niezależnych równań opisujących dany obwód, w oparciu o znajomość drzewa grafu tego obwodu. Problem ten sprowadza się do znalezienia jednej z macierzy zorientowanego grafu [1], [2].

W pracy podano algorytm pozwalający znaleźć drzewo, gdy narzucone nań pewne ograniczenia, wyznaczający jednocześnie macierz oczkową [2] grafu danego obwodu. Podano również procedurę realizującą ten algorytm w języku ALGOL 1204. Gałęzie grafu ponumerujemy od 1 do g a węzły od 1 do w, gdzie g to liczba gałęzi danego obwodu, w liczba węzłów.

Każda gałąź łączy jak wiadomo dwa węzły, moża ją zatem opisać numerami węzłów które łączy.

Wielkości wyjściowe algorytmu; numery gałęzi drzewa (tree) umieścimy w macierzy $\underline{T} = [t_h]$ o wymiarze 1 x (w-1), a numery pozostałych gałęzi, tzw. gałęzi łączących (link) w macierzy $\underline{L} = [l_f]$ o wymiarze 1 x(g-w+1).

Macierz oczkowa, której elementów będziemy szukać równolegle ze znajdywaniem drzewa, to macierz $\underline{A} = [a_{hf}]$ o wymiarze (w-1)x(g-w+1). Każdemu wierszowi tej macierzy odpowiada jedna gałąź drzewa a każdej kolumnie jedna gałąź łącząca, czyli jedno niezależne oczko obwodu, przy czym:

a_{hf} = 1(-1), jeśli gałąź t_h znajduje się w oczku odpowiadającym gałęzi l_f i ma kierunek zgodny (przeciwny) z kierunkiem tego oczka.

a_{hf} = 0, jeśli gałąź t_h nie znajduje się w oczku odpowiadającym gałęzi l_f.

2. OGÓLNY OPIS ALGORYTMU

Idea algorytmu opiera się na tym, że do gałęzi najbardziej pożądanej w drzewie dodawane są po jednej pozostałe gałęzie, według ich przydatności w drzewie.

Po dodaniu kolejnej gałęzi następuje sprawdzenie czy tworzy ona oczko z zaakceptowanymi już do drzewa gałęziami, czy nie. W pierwszym przypadku gałąź ta powiększa zbiór gałęzi łączących, w drugim zbiór gałęzi drzewa. Szukając drzewa posługiwać się będziemy grafem niezorientowanym obwodu. Znajomość kierunków gałęzi będzie nam potrzebna dopiero przy wyznaczaniu znaku niezerowych elementów macierzy A. Dane o strukturze obwodu umieścimy w macierzach $\underline{NG} = [ng_i]$, $\underline{WP} = [wp_i]$, $\underline{WK} = [wk_i]$ p wymiarach 1 x g. Elementami $\underline{NG}$ będą numery gałęzi a $\underline{WP}$;$\underline{WK}$ węzły początkowe i końcowe tych gałęzi. Przed przystąpieniem do zastosowania algorytmu należy uporządkować elementy macierzy $\underline{NG}$, $\underline{WP}$ i $\underline{WK}$ tak by pierwszym pozycjom tych macierzy odpowiadały, gałęzie, które muszą należeć do drzewa a ostatnim gałęzie, które nie mogą należeć do drzewa lub są w nim najmniej pożądane. Oczywiście, danej pozycji w $\underline{NG}$, $\underline{WP}$ i $\underline{WK}$ musi odpowiadać ta sama gałąź.

Utworzymy dwie pomocnicze macierze $\underline{X} = [x_j]$ i $\underline{Y} = [y_j]$ o w wymiarach 1 x w, na których dokonywane będą operacje poszukiwania drzewa i macierzy $\underline{A}$. Elementami $\underline{Y}$ będą numery gałęzi, a elementami $\underline{X}$ numery węzłów końcowych tych gałęzi. Pozycja, na jakiej będzie znajdowała się dana gałąź w $\underline{Y}$ i jej węzeł końcowy w $\underline{X}$ będzie określona przez węzeł początkowy tej gałęzi.

Elementom macierzy $\underline{X}$ i $\underline{Y}$ odpowiada zatem pewien podgraf. Weźmy dla przykładu graf z rys. 2 macierzom $\underline{Y} = [1,7,4,3,8]$, $\underline{X} = [3,1,4,2,1]$ odpowiada podgraf, na który składają się gałęzie o numerach 1,7,4,3,8 lub operując numerami węzłów (1-3), (2-1), (3-4), (4-2), (5-1); a macierzom $\underline{Y} = [1,0,0$ $4,8]$, $\underline{X} = [3,0,0,3,1]$ podgraf 1,0,0,4,8 czyli (1-3),(2-0),(3-0), (4-3), (5-1); gdzie węzeł o numerze 0 jest nie istniejącym w obwodzie węzłem odniesienia. Jeśli w macierzach $\underline{X}$ i $\underline{Y}$ na danej pozycji znajduje się zero to oznacza to, że w podgrafie znajdującym się w tych macierzach nie ma gałęzi o węźle początkowym równym numerowi tej pozycji.

Taką fikcyjną gałąź 0 (k-0) nazywać będziemy gałęzią zerową. Jeśli w podgrafie znajduje się co najwyżej jedno oczko to można go umieścić w $\underline{X}$ i $\underline{Y}$. Inaczej mówiąc, gdy w danym podgrafie nie ma oczka lub jest tylko jedno oczko to przez odwracanie kierunków gałęzi (zamianę węzła początkowego z końcowym) zawsze można doprowadzić do sytuacji, kiedy w tym podgrafie nie będzie dwóch gałęzi o takim samym węźle początkowym.

W kolejnym kroku i, należy wykonać następujące operacje:

1. Wprowadzić na pozycję odpowiadającą numerowi węzła początkowego tej gałęzi - wp_i, numer gałęzi - ng_i do $\underline{Y}$ a numer węzła końcowego gałęzi - wk_i do $\underline{X}$.
   Pociąga to na ogół za sobą konieczność odwracania kierunków niektórych a w najgorszym przypadku wszystkich gałęzi, znajdujących się dotychczas w $\underline{X}$:$\underline{Y}$, należy bowiem uporządkować kierunki gałęzi tak, by nie było dwóch gałęzi o takim samym węźle początkowym.
2. Sprawdzić ścieżkę [1] zaczynającą się od gałęzi ng_i. Ścieżka ta musi osiągnąć jedną z dwóch gałęzi:

   a) gałąź zerową np. (wp_i-wk_i), (wk_i-s), (s-t), (t-0)
   b) gałąź o węźle końcowym wp_i np. (wp_i-wk_i), (wk_i-s), (s-t),(t-wp_i).

   Jeśli zachodzi przypadek 2a, tzn. gałąź ng_i nie tworzy oczka z podgrafem znajdującym się w $\underline{Y}$; $\underline{X}$ należy:

3a Wprowadzić gałąź ng_i do macierzy $\underline{T}$ (t_h = ng_i)
4a Przejść do następnej gałęzi (i = i+1)

   Jeśli zachodzi przypadek 2b, tzn. gałąź ng_i tworzy oczko z podgrafem znajdującym się w $\underline{Y}$;$\underline{X}$ należy:

3b Wprowadzić gałąź ng_i do macierzy $\underline{L}$ (l_f = ng_i)
4b Sprawdzić kierunki gałęzi drzewa tworzących oczko.
   W wierszach macierzy $\underline{A}$ odpowiadających tym gałęziom a kolumnie odpowiadającej aktualnie rozważanej gałęzi łączącej l_f wstawiana jest 1 lub -1, zależnie od tego czy kierunek danej gałęzi drzewa jest zgodny z kierunkiem obchodzenia oczka (wyznaczonym przez kierunek l_f) czy nie tzn. ...zależnie od tego czy pozycja na jakiej znajduje się dana gałąź drzewa w $\underline{Y}$ zgodna jest z jej rzeczywistym węzłem początkowym czy nie.

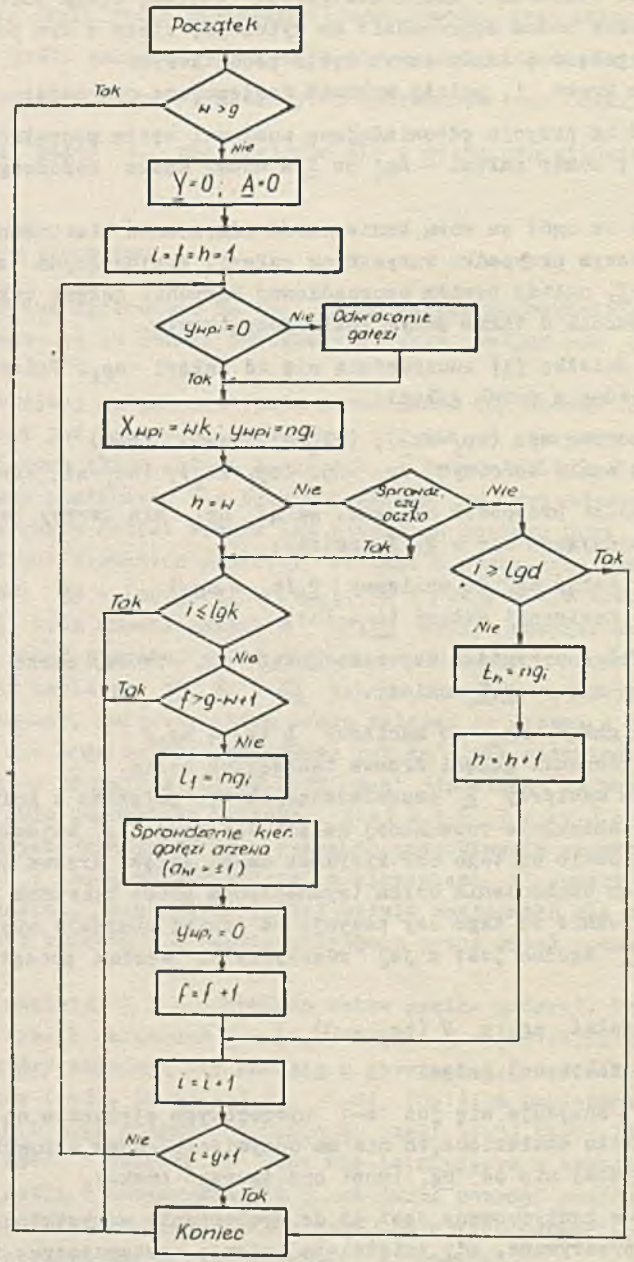
5b Wykreślić gałąź ng_i z $\underline{Y}$ (ng_i = 0)

6b Przejść do następnej gałęzi (i = i+1)

   Jeśli w $\underline{T}$ znajduje się już w-1 niezerowych elementów co oznacza,że drzewo zostało znalezione,to nie ma oczywiście sensu sprawdzanie ścieżki zaczynającej się od ng_i (musi ona tworzyć oczko).

Powyższy proces kontynuowany jest aż do wyczerpania wszystkich gałęzi. Obliczenia są przerywane, gdy zaistnieje jeden z następujących trzech przypadków:

1. Gałęzie, które muszą należeć do drzewa, tworzą oczko (ten przypadek nazywany będzie dalej konfiguracją osobliwą 1).



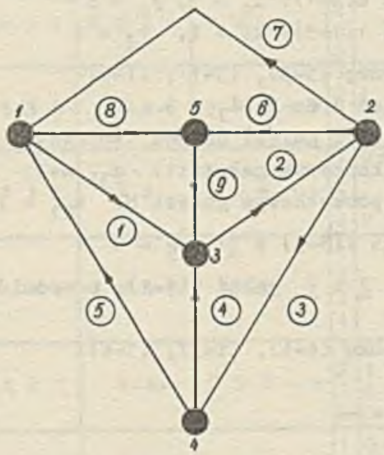
Rys. 1. Schemat blokowy algorytmu
lgk - liczba gałęzi, które muszą koniecznie znaleźć się w drzewie, lgd - liczba gałęzi dopuszczalnych w drzewie

2. Nie da się utworzyć drzewa bez gałęzi, które do drzewa nie mogą należeć (ten przypadek nazywany będzie konfiguracją osobliwą 2).
3. Graf jest niespójny.

Schemat blokowy algorytmu przedstawiony jest na rys. 1.

3. OPIS ALGORYTMU KROK PO KROKU

Opisu algorytmu krok po kroku dokonamy na przykładzie. Dany jest obwód, którego graf przedstawiony jest na rys. 2. Załóżmy, że gałęzie 4 i 8 muszą znaleźć się w drzewie, a pozostałe gałęzie uszeregowane są według ich przydatności w drzewie następująco: 1,9,5,3,2. Rozmieszczenie elementów macierzy NG, WP i WK ilustruje tablica 1.



Rys. 2. Graf przykładowego obwodu

Tablica 1

i nr kolejny pozycji	ng_i nr gałęzi	wp_i węzeł początkowy	wk_i węzeł końcowy	Uwagi
1 2	4 8	4 1	3 5	gałęzie, które muszą należeć do drzewa lgk=2
3 4 5 6 7	1 9 5 3 2	1 5 4 2 3	3 3 1 4 2	gałęzie, które nie muszą należeć do drzewa uszeregowane wg ich przydatności w drzewie lgd=7
8 9	6 7	2 2	5 1	gałęzie nie mogące należeć do drzewa

Algorytm szukania drzewa krok po kroku przedstawia się następująco:

0. Wyzerować $\underline{Y}$ oraz $\underline{A}$

I. Wprowadzić do $\underline{X}$, $\underline{Y}$ i $\underline{T}$ gałąź 4(4-3); $y_4 = 4$, $x_4 = 3$, $t_1 = 4$

II. 1. Wprowadzić do $\underline{X}$ i $\underline{Y}$ gałąź 8(1-5): $y_1=0$, $x_1 = 5$

2. Sprawdzić ścieżkę zaczynającą się od gałęzi 8:(1-5), (5-0)

3. Wprowadzić do $\underline{T}$ gałąź 8: $t_2=8$.

Jak widać gałęzie, które muszą znakeźć się w drzewie nie tworzą oczka.

III. 1. Wprowadzić do $\underline{X}$ i $\underline{Y}$ gałąź 1(1-3): $x_1 = 3$, $y_1 = 1$

2. Odwrócić gałąź 8(1-5): $x_5 = 1$, $y_5 = 8$

2. Sprawdzić ścieżkę (1-3), (3-0)

3. Wprowadzić gałąź 1 do $\underline{T}$: $t_3 = 1$

IV. 1. Wprowadzić do $\underline{X}$ i $\underline{Y}$ gałąź 9(5-3): $x_5 = 3$, $y_5 = 9$

Odwrócić gałąź 8(5-1): $x_1 = 5$, $y_1 = 8$

Odwrócić gałąź 1(1-3): $x_3 = 1$, $y_3 = 1$

2. Sprawdzić ścieżkę (5-3), (3-1), (1-5)

3. Wprowadzić gałąź 9 do $\underline{L}$: $l_1 = 9$

4. Sprawdzić kierunki gałęzi drzewa tworzących oczko:

3 jest węzłem końcowym gałęzi 1: $a_{31} = -1$

1 jest węzłem początkowym gałęzi 8: $a_{21} = 1$

5. Wykreślić gałąź 9(5-3) z $\underline{Y}$: $y_5 = 0$

V. 1. Wprowadzić do $\underline{X}$ i $\underline{Y}$ gałąź 5(4-1), odwrócić gałęzie 4(4-3),1(3-1) 8(1-5)

2. Sprawdzić ścieżkę (4-1), (1-3), (3-4)

3. $l_2 = 5$

4. $a_{32} = 1$, $a_{12} = -1$

5. $y_4 = 0$.

VI. 1. Wprowadzić do $\underline{X}$ i $\underline{Y}$ gałąź 3(2-4)

2. Sprawdzić ścieżkę (2-4), (4-0)

3. $t_4 = 3$.

W $\underline{Y}$:$\underline{X}$ znajduje się w-1 elementów niezerowych, tzn. drzewo zostało znalezione, przy czym nie znajduje się w nim żadna gałąź niedozwolona. Podobnie postępując z gałęziami 2,6 i 7 (pomijając punkt 2) otrzymamy pozostałe niezerowe elementy macierzy $\underline{A}$.

Proces tworzenia drzewa i macierzy $\underline{A}$ ilustruje tablica 2.

Tablica 2

Nr kroku	Y 1 2 3 4 5	X 1 2 3 4 5	T 1 2 3 4	L 1 2 3 4 5		A 1 2 3 4 5	Uwagi
0	0 0 0 0 0	- - - - -	- - - -	- - - - -	1 2 3 4	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
I	0 0 0 4 0	- - - 3 -	4 - - -	b.z.	1 2 3 4	b.z.	
II	8 0 0 4 0	5 - - 3 -	4 8 - -	b.z.	1 2 3 4	b.z.	gałęzie, które muszą znaleźć w drzewie nie tworzą oczka
III	1 0 0 4 8	3 - - 3 1	4 8 1 -	b.z.	1 2 3 4	b.z.	
IV	8 0 1 4 9	5 - 1 3 3	b.z.	9 - - - -	1 2 3 4	0 0 0 0 0 1 0 0 0 0 -1 0 0 0 0 0 0 0 0 0	
V	1 0 4 5 8	3 - 4 1 1	b.z.	9 5 - - -	1 2 3 4	0 -1 0 0 0 1 0 0 0 0 -1 1 0 0 0 0 0 0 0 0	
VI	1 3 4 0 8	3 4 4 - 1	4 8 1 3	b.z.	1 2 3 4	b.z.	utworzone zostało drzewo bez gałęzi niedopuszczalnych
VII	1 2 4 3 1	3 3 4 2 1	b.z.	9 5 2 - -	1 2 3 4	0 -1 -1 0 0 1 0 0 0 0 -1 1 0 0 0 0 0 -1 0 0	
VIII	1 6 4 3 1	3 5 4 2 1	b.z.	9 5 2 6 -	1 2 3 4	0 -1 -1 -1 0 1 0 0 -1 0 -1 1 0 1 0 0 0 -1 -1 0	
IX	1 7 4 3 1	3 1 4 2 1	b.z.	9 5 2 6 7	1 2 3 4	0 -1 -1 -1 -1 1 0 0 -1 0 -1 1 0 1 1 0 0 -1 -1 -1	koniec obliczeń

Uwagi:

- na danej pozycji oznacza dowolną wartość

0 na danej pozycji oznacza, że po danym kroku pozyjca ta jest zerowana

b.z. oznacza, że w danym kroku dana macierz pozostaje bez zmian.

4. OPIS PROCEDURY REALIZUJACEJ ALGORYTM SZUKANIA DRZEWA I MACIERZY OCZKOWEJ

Wielkościami wejściowymi do procedury są:

1. w - liczba węzłów obwodu.
2. g - liczba gałęzi obwodu.
3. $\underline{NG} = [ng_i]$ - macierz o wymiarze 1 x g zawierającą numery gałęzi.
4. $\underline{WP} = [wp_i]$ - macierz o wymiarze 1 x g zawierającą numery węzłów początkowych.
5. $\underline{WK} = [wk_i]$ - macierz o wymiarze 1 x g zawierającą numery węzłów końcowych.

   UWAGA: Gałęzie w macierzach NG, WP i WK należy przed wejściem do procedury uszeregować wg ich przydatności w drzewie.
6. lgk - liczba gałęzi, które koniecznie należeć muszą do drzewa
7. lgd - liczba gałęzi, których obecność jest dopuszczalna w drzewie.

Wielkościami wyjściowymi z procedury są:

1. $\underline{T} = [t_i]$ - macierz o wymiarze 1 x(w-1) zawierająca numery gałęzi drzewa.
2. $\underline{L} = [l_i]$ - macierz o wymiarze 1 x(g-w+1) zawierająca numery gałęzi łączących.
3. $\underline{A} = [a_{ij}]$ - macierz oczkowa o wymiarze (w-1)x(g-w+1).
4. lab - etykieta do której następuje przeskok w wypadku wykrycia że graf jest niespójny, lub wystąpiła jedna z konfiguracji osobliwych.

Czas obliczania procedury na m.c. ODRA 1204 dla obwodów posiadających do 100 gałęzi i do 20 węzłów waha się od kilku do kilkunastu sekund. Jak wykazały doświadczenia dla różnych drzew tego samego grafu czasy obliczeń różnią się i tak, dla zupełnego grafu o 5 węzłach, czas obliczeń dla różnych drzew wahał się od 1 do 5 sekund. Ponadto, czas obliczeń zależy od stopnia zupełności grafu.

TABULOGRAM PROCEDURY

```
procedure drzewo (w,g,ng,wp,wk,lgk,lgd,t,l,a,lab);
integer w,g,lgk,lgd;
integer array ng, wp,wk,t,l,a;
label  lab;
begin
integer i,j,f,h,u,u1,z,z1;
integer array x,y[1:w],r[1:g];
if g<w-1 then go to gn
          else go to pocz;
gn:print('?graf  niespojny ');
go to lab;
```

```
pocz:for i:=1 step 1 until w-1 do
 for j:=1 step 1 until g-q+1 do a[i,j]:=0;
for i:=1 step 1 until w do y[i]:=0;
f:=h:=1;
for i:=1 step 1 until g do
begin
j:=wp[ng[i]] ;
if y[j]=0
 then begin
      y[j]:=ng[i];
      x[j]:=wk[ng[i]] ;
      go to czyoczko
      end
else begin
      u:=x[j];
      z:=y[j];
      y[j]:=ng[i];
      x[j]:=wk[ng[i]] ;
      eta:if y[u]= 0
      then begin
           y[u]:=z;
           x[u]:=j;
           go to czyoczko
           end
      else begin
           u1:=x[u];
           z1:=y[u];
           y[u]:=z;
           x[u]:=j;
           j:=u;
           u:=u1;
           z:=z1;
           go to eta
           end
      end;
czyoczko:if h=w then go to oczko;
j:=x[wp[ng[i]]] ;
fi:if y[j]=0 then go to nieoczko
                 else j:=x[j];
if j≠wp[ng[i]] then go to fi;
oczko:if lgk>1
 then begin
      print ( ?konfiguracja  osobliwa ␣1');
      go to lab
      end;
if f>g-w+1 then go to gn;
l[f]:=ng[i];
j:=x[wp[ng[i]]] ;
fi1:if wp[y[j]]=j then u:=1 else u:=-1;
a[r[y[j]], f]:=u;
j:=x[j];
if j≠wp[ng[i]] then go to fi1;
y[wp[ng[i]]]:=0;
f:=f+1;
go to dalej;
nieoczko:if i >lgd
 then begin
      print( ?konfiguracja  osobliwa ␣2');
      go to lab
      end
 else begin
      r[ng[i]]:=h;
      t[h]:=ng[i];
      h:=h+1
      end;
 dalej:end
 end;
 ?
```

LITERATURA

[1] SESHU S., REED M.: Linear graphs and electrical networks - Addison - Wesley P.C. 1961.

[2] DESOER Ch., KUH E.: Basic circuit theory - McGraw Hill 1969.

[3] CHAR J.P.: Generation of trees, two-trees, and storage of master forest - IEEE Trans. 1968 CT-15.

[4] CHAR J.P.: Circuit, cutset and path enumeration, and other applica - tions of edge - numbering convention - Proc. IEEE vol. 117, No 3, March 1970.

АЛГОРИТМ ОТЫСКИВАНИЯ ДЕРЕВА И КОНТУРНОЙ МАТРИЦЫ ЦЕПИ ЕСЛИ ПОДАНЫ ОГРАНИЧЕНИЯ

Р е з ю м е

В работе описан алгоритм отыскивания дерева графа цепи, если поданы ограничения на дерево, с одновременным вычислением контурной матрицы цепи. На примере постепенно описано алгоритм метода.
Дана реализация этого алгоритма по программе на языке ALGOL 1204.

AN ALGORITHM FOR FINDING THE TREE AND LOOP MATRIX IN ASSUMED RESTRICTIONS

S u m m a r y

This articles describes ab algorithm for finding the tree of graph when some restrictions are assumed, and the simultaneous calculation of the loop matrix. A step-by-step description of the algorithm based on an example is presented. Along with a procedure algorithm realizing in ALGOL 1204.