

Na prawach rękopisu

INSTYTUT CYBERNETYKI TECHNICZNEJ
POLITECHNIKI WROCŁAWSKIEJ

Raport Serii PREPRINTY Nr 20/82

ALGORYTMY OPTYMALIZACJI
NIELINIOWYCH SIECI PRZEPŁYWOWYCH
DROGĄ MINIMALIZACJI
STRAT ENERGETYCZNYCH

praca doktorska

Jan Nikodem

Promotor:

doc.dr inż. Tadeusz Stanicki

Słowa kluczowe:

sieci nieliniowe, system złożony, optymalizacja nieliniowa, algorytm, funkcja celu

Wydawnictwo Politechniki Wrocławskiej
Wrocław 1982 r.

SPIS TREŚCI

	str.
1. WSTĘP	4
1.1. Zagadnienie optymalizacji w sieciach przepływowych.	4
1.2. Formalizacja opisu sieci przepływowych	8
1.3. Określenie zadania optymalizacji	14
1.4. Sformułowanie celu i tezy rozprawy	17
2. ANALIZA STOSOWANYCH METOD OPTYMALIZACJI NIELINIOWYCH SIECI PRZEPŁYWOWYCH W ASPEKcie TEZY ROZPRAWY	18
2.1. Wybrane metody optymalizacji nieliniowych sieci przepływowych	18
2.2. Optymalizacja sieci przepływowych drogą redukcji zadania wyjściowego	22
2.3. Podsumowanie analizy efektywności omówionych metod.	24
3. ZASTOSOWANIE METODY PRĄDÓW SKŁADOWYCH I WYRÓWNAWCZYCH DO ROZWIĄZYWANIA ZADAŃ OPTYMALIZACJI NIELINIOWYCH SIECI PRZEPŁYWOWYCH	25
3.1. Metoda prądów składowych i wyrównawczych - warunki konieczne i wystarczające punktu optymalnego	26
3.2. Uogólnienie metody prądów składowych i wyrównaw- czych w celu redukcji problemu wyjściowego	31
3.3. Zasady optymalizacji nieliniowych sieci przepływo- wych w oparciu o transformację zadania wyjściowego do układu równań algebraicznych	40
4. ALGORYTMY MINIMALNO-ENERGETYCZNEJ OPTYMALIZACJI SIECI PRZEPŁYWOWYCH OPARTE NA POSTACI ZREDUKOWANEJ PROBLEMU WYJŚCIOWEGO	48
4.1. Metoda Crossa-Łobaczewa iteracyjnego rozwiązywania układów równań nieliniowych	49
4.2. Opis procedury redukcji i procedury przyspieszają- cej otrzymanie rozwiązania optymalnego	53
4.3. Algorytmy optymalizacji nieliniowych sieci przepły- wowych z wykorzystaniem postaci zredukowanej prob- lemu wyjściowego	59
4.4. Sformułowanie i analiza warunków zbieżności	74
4.5. Badania testowe i analiza własności opracowanych algorytmów	79
4.6. Przykłady obliczeniowe	87

5. WYNIKI BADAŃ PORÓWNAWCZYCH OPRACOWANEGO ALGORYTMU	93
5.1. Sprowadzenie zadania optymalizacji do zagadnienia transportowego z nieliniową funkcją kosztów	93
5.2. Rozwiązanie nieliniowego zadania transportowego metodami Rosenbrocka, Fletchera-Powella-Dawidona i zmodyfikowaną metodą Newtona	95
5.3. Ocena opracowanego algorytmu optymalizacji w świetle przeprowadzonych badań porównawczych	96
6. PRZYKŁADOWE ZASTOSOWANIA PRAKTYCZNE OPRACOWANEJ METODY..	101
6.1. Projektowanie i modernizacja sieci hydraulicznych, gazowych i wentylacyjnych	101
6.2. Optymalizacja procesu sterowania siecią wodociągową w czasie rzeczywistym z wykorzystaniem minikomputera MERA-400	104
7. UWAGI I WNIOSKI KOŃCOWE	108
8. LITERATURA	111
9. DODATEK	115
9.1. Postać źródłowa programu #ALG1 - opracowanego w oparciu o algorytm I	115
9.2. Postać źródłowa programu #ALG2 - opracowanego w oparciu o algorytm II	121
9.3. Wyniki obliczeń programu #ALG1 dla przykładu 1 ...	124
9.4. Wyniki obliczeń programu #ALG2 dla przykładu 2 ...	125

Praca dotyczy zagadnienia optymalizacji statycznej w nieliniowych sieciach przepływowych (1.2.21)-(1.2.23). Postawione zostało wielowymiarowe zadanie (1.3.4)-(1.3.7) optymalizacji nieliniowej z ograniczeniami. Scharakteryzowano znaną ideę prądów składowych i wyrównawczych dla modeli Kirchhoffa (3.1.1)-(3.1.4), a następnie korzystając z jej uogólnienia, określono warunki konieczne i wystarczające rozwiązania optymalnego rozpatrywanego problemu. Przyjmując sformułowane warunki (1.3.4)-(1.3.6), (3.2.36) za podstawę do wyznaczania rozwiązania problemu podstawowego (1.3.4)-(1.3.7), przeprowadzono redukcję tego zagadnienia do układu nieliniowych równań algebraicznych (3.3.36). Na bazie opracowanej metody skonstruowano dwa algorytmy obliczeniowe, które oprogramowano i przetestowano pod kątem własności numerycznych. Przeprowadzono analizę zbieżności oraz wykonano badania porównawcze względem zmodyfikowanej metody Newtona, metody Fletchera-Powella-Davidona i metody Rosenbrocka.

1. WSTĘP

1.1. Zagadnienie optymalizacji w sieciach przepływowych

Optymalizacja sieci przepływowych to zagadnienie, które coraz częściej pojawia się zarówno w publikacjach naukowych, jak również w codziennej praktyce inżynierskiej. W okresie kilkunastu ostatnich lat, z uwagi na szerokie zastosowanie w tych pracach maszyn matematycznych, nastąpił niezwykle gwałtowny wzrost zainteresowania tą dziedziną badań, w której teorię grafów stosuje się z wyjątkowo dużym powodzeniem.

Sieć, jako pojęcie formalne, stanowi naturalny i poglądowy model matematyczny szeregu rodzajów obiektów fizycznych. Pierwszy wykorzystał to w 1847 r. G. Kirchhoff [27], który definiując zmienne poprzeczne (prądy) oraz zmienne podłużne (napięcia), określił warunki (I, II prawa Kirchhoffa, prawo Ohma) jakie powinny one spełniać, aby dana sieć modelowała obwód elektryczny. Udoskonaleniem jego wyników jest praca J. Maxwella [38] z 1892 r., w której autor wykazuje, że zagadnienie analizy obwodu elektrycznego sprowadza się do problemu optymalnego określenia przepływów, które minimalizują kwadratową funkcję celu przy liniowych ograniczeniach.

A. Łobaczew [36] w 1936 r. modyfikując sieciowy model obiektów fizycznych zaproponowany przez Kirchhoffa, opracował metodę analizy nieliniowych obwodów hydraulicznych i pneumatycznych, a niezależnie od niego wynik ten uzyskał również H. Cross [10]. Wykazał

on ponadto analogię pomiędzy potencjałami w sieciach przepływowych a rozkładem naprężeń w konstrukcjach mechanicznych.

Dalszy rozwój optymalizacji sieci przepływowych wiąże się z opublikowaniem w 1956 r. przez Forda i Fulkersona [18] oraz niezależnie Eliasa, Feinsteina i Shannona [13], prac poświęconych optymalizacji warunków przepływu w sieciach. W publikacjach tych, które dały początek bieżącemu rozwojowi takich dziedzin badań operacyjnych, jak sieci transportowe czy sieci czynności, pokazano jak wiele zagadnień kombinatorycznych z zakresu badań operacyjnych można sprowadzić i rozwiązać w oparciu o zagadnienie optymalizacji przepływu w sieciach.

Duża wartość użytkowa aparatu formalnego, jakim jest pojęcie sieci, spowodowała również w ramach informatyki, automatyki i sterowania, rozwój takich gałęzi wiedzy, jak teoria sieci transmitancyjnych czy komputerowych. Do podstawowych i charakterystycznych prac w tym zakresie można między innymi zaliczyć prace Mansona [61] i Lorensa [60].

Jeżeli jednak za kryterium jakości modelu formalnego przyjąć liczbę jego konkretnych zastosowań praktycznych, wtedy pojęcie sieci zaproponowane przez Kirchhoffa okaże się jednym z korzystniejszych. I tak Praeger [50] w 1954 r. rozwiązując problem optymalizacji sieci transportowych, posłużył się modelem sieciowym tego zagadnienia opartym na koncepcji sieci Kirchhoffa. Youngs [59] w 1971 r. określił warunki optymalnego kolorowania map korzystając także z pojęcia sieci Kirchhoffa, które spotykane jest również w zagadnieniach z zakresu architektury [37], elektryczności [38] czy mechaniki [40].

Sieci Kirchhoffa stanowią również podstawę do takich uogólnień, jak pojęcie sieci Leontieffa [62] znajdujących szerokie zastosowanie w badaniu zagadnień optymalizacji procesów przygotowania i organizacji produkcji.

Bogata literatura z zakresu optymalizacji sieci, wśród wielu pozycji dotyczących głównie problemów optymalizacji liniowej, zawiera również grupę prac poświęconych optymalizacji nieliniowych sieci przepływowych. Zagadnienie to rozpatrywane było m.in. przez Watanadę [57], Treumpera [55], Rasmusena [52], Haarhoffa i Buysa [21], którzy zaproponowali algorytmy wykorzystujące metody programowania nieliniowego, budując procedury oparte w przeważającej

większości na algorytmach gradientowych. Wielu autorów formułując zasady optymalizacji wybranych typów sieci nieliniowych, podkreśla ich zadowalającą efektywność w odniesieniu do sieci małych i średnich rozmiarów. Odczuwalny jest natomiast, wynikający z potrzeb praktycznych, brak tego typu metod dostosowanych do optymalizacji sieci dużych.

W związku z powyższymi uwagami, w niniejszej pracy dokonuje się próby opracowania metody optymalizacji nieliniowych sieci przepływowych, umożliwiającej budowę prostych i szybkich algorytmów obliczeniowych nawet w przypadku sieci dużych rozmiarów.

W rozdziale 1 przedstawiono aparat formalny wykorzystywany w pracy oraz określono zadanie optymalizacji sieci przepływowych. Sformułowano w nim także tezę i określono cel rozprawy doktorskiej.

Rozdział 2 zawiera przegląd spotykanych w literaturze metod rozwiązywania zagadnień optymalizacji nieliniowych sieci przepływowych. Zamieszczono w nim również analizę efektywności omówionych metod w aspekcie tezy rozprawy.

W rozdziale 3 przedstawiono ideę prądów składowych i wyrównawczych dla modeli liniowych Kirchhoffa (3.1.1)-(3.1.4) oraz zaprezentowano koncepcję uogólnienia tej idei również na nieliniowe sieci przepływowe. Określono warunki konieczne i wystarczające rozwiązania optymalnego a następnie sformułowano zasady optymalizacji sieci przepływowych w oparciu o transformację zadania wyjściowego (1.3.4)-(1.3.7) do układu równań algebraicznych (3.3.36), umożliwiającą redukcję wymiarowości rozpatrywanego zagadnienia.

Rozdział 4 zawiera opisy procedur i algorytmów optymalizacji opracowanych przez autora. Ponadto sformułowane w nim zostały warunki zbieżności, opisano badania testowe oraz przeprowadzono analizę własności numerycznych przedstawionych algorytmów. Na zakończenie, w celu pełniejszego zilustrowania przebiegu obliczeń, rozwiązano dwa przykłady.

Porównanie opracowanej przez autora metody optymalizacji sieci przepływowych w odniesieniu do zmodyfikowanej metody Newtona, metody Fletchera-Powella-Davidona i metody Rosenbrocka oraz ocenę efektywności działania opracowanych algorytmów, zamieszczono w rozdziale 5 zawierającym także uzyskane wyniki badań numerycznych.

W rozdziale 6 przedstawiono dwa przykłady praktycznych zastosowań opracowanej metody. Pierwszy zaprezentowany na III Konferencji Metod i Środków Projektowania Automatycznego - Warszawa 1981, dotyczy wykorzystania opracowanej metody w projektowaniu i modernizacji sieci hydraulicznych, gazowych i wentylacyjnych. Drugi, przedstawiony na III Konferencji nt. Zastosowanie Komputerów w Przemśle - Szczecin 1981, ukazuje w jaki sposób otrzymane wyniki umożliwiają automatyzację procesu optymalnego sterowania sieciami przepływowymi w czasie rzeczywistym.

Pracę zamykają uwagi i wnioski końcowe, wykaz pozycji literaturowych oraz dodatek zawierający wydruki programów źródłowych i wyniki obliczeń przykładów przedstawionych w rozdziale 4.

Przyjęta w pracy symbolika zaczerpnięta jest m.in. ze znanych i ogólnie dostępnych prac [6,11,22]. Jednakże, w celu łatwiejszego posługiwania się pracą, przedstawimy listę ważniejszych, stosowanych w pracy oznaczeń:

- $\wedge, \vee, \bar{\vee}, \Leftrightarrow, \Rightarrow$ - funktory rachunku zdań: (i), (lub), (albo), równoważność, implikacja
- $\bigcap_{j=1}^k, \bigcup_{i=1}^p$ - operacje mnogościowe odpowiednio: iloczynu i sumy
- $\forall, \exists, \exists!$ - kwantyfikatory: ogólny (dla każdego), szczegółowy (istnieje takie), szczegółowy jednostkowy (istnieje dokładnie jedno)
- $\{a \in A \mid \varphi(a)\}$ - zbiór wszystkich elementów zbioru A, dla których prawdziwa jest funkcja zdaniowa $\varphi(a)$
- $\{\varphi_j\}$ - zbiór funkcji φ_j
- $\{x^n\}$ - ciąg elementów
- $X\{\varphi_i \mid i=1, 2, \dots, n\}$ - produkt kartezjański zbioru n-elementowego
- $\emptyset$ - zbiór pusty
- $\|a\|$ - norma elementu "a"
- $|b|$ - wartość bezwzględna liczby "b"
- $\nabla \varphi(x)$ - gradient funkcji φ w punkcie x
- $(x_1, x_2, \dots, x_n)$ - układ nieuporządkowany
- $X = \langle x_i \mid i \in N \rangle$ - układ (ciąg) uporządkowany
- $X, Y, U, W, E, \dots$ - zbiory
- $x, y, p, q, h, \dots$ - wektory
- $f, g, \psi, \pi, \xi, \dots$ - funkcje
- A, B, C - macierze: incydencji, obwodów i przekrojów danego grafu

A_z	- zredukowana macierz incydencji wymiaru $(w-1) \times n$
B_0	- macierz obwodów podstawowych względem dendrytu O
0	- macierz zerowa
1_d	- diagonalna macierz jednostkowa
$\lambda_i(A)$	- i-ta wartość własna macierzy A
mr	- r-ta marszruta w danym grafie G
E_{jk}^r	- r-ty łańcuch, łączący wierzchołek początkowy - w_j z wierzchołkiem końcowym - w_k
C^r	- cykl r-ty
C_D^r	- r-ty cykl podstawowy względem dendrytu D
S_{jl}	- zbiór wszystkich łańcuchów prostych łączących wierzchołki: w_j oraz w_l
R^+, R^-	- zbiory liczb rzeczywistych: dodatnich i ujemnych
N_k^1	- zbiór wszystkich liczb naturalnych nie mniejszych niż "k" i nie większych niż "l"
$H(x) = \frac{\partial^2 f(x)}{\partial x^T \partial x}$	- hesjan funkcji f w punkcie x
$\hat{x}, \hat{y}, \hat{w}, \dots$	- rozwiązania w punkcie optymalnym
$\text{card } X$	- moc zbioru X
crd_k^y	- k-ta składowa wektora y
$\det A$	- wyznacznik macierzy A
$\text{rank } A$	- rząd macierzy A
$\text{sgn}(x_i)$	- znak x_i
$(\# \# \#)$	- znacznik końca dowodu
$:=, 999$	- oznaczenia używane w opisach algorytmów obliczeniowych: znak podstawienia, znacznik wierzchołka wiszącego (w dendrycie D)

1.2. Formalizacja opisu sieci przepływowych

W ujęciu ogólnej teorii systemów [39], system to relacja będąca podzbiorem produktu kartezjańskiego

$$S \subset X \{ P_j \mid j \in I \} \quad (1.2.1)$$

gdzie elementy P_j nazywane są obiektami systemu S , natomiast I jest pewnym zbiorem indeksów.

System S nazywany wejściowo-wyjściowym w przypadku, gdy zależność (1.2.1) można zapisać w formie:

$$S \subset U \times W \quad (1.2.2)$$

$U \subset \mathbb{R}^l$ - jest zbiorem wielkości wejściowych, a $W \subset \mathbb{R}^k$ zbiorem wyjść systemu S .

Często spotykany w problemach analizy systemowej podejściem jest traktowanie systemu złożonego, jako zbioru wyodrębnionych, lecz powiązanych ze sobą elementów składowych. Każdy z tych elementów może być rozważany jako pewien podsystem S_j ($j=1,2,\dots,m$), który w szczególności jest również systemem typu (1.2.2)

$$S_j \subset U_j \times W_j \quad (1.2.3)$$

W celu otrzymania pełnego opisu systemu złożonego S wymagana jest znajomość relacji S_j opisujących wszystkie obiekty składowe oraz znajomość struktury wzajemnych powiązań $\mathcal{P}$ pomiędzy tymi obiektami. Powiązania te określa ogólnie zależność

$$\mathcal{P}(S) = \mathcal{P}(U, W) = 0 \quad (1.2.4)$$

$$\text{przy czym } U = \{U_j \mid j=1,2,\dots,m\} \text{ oraz } W = \{W_j \mid j=1,2,\dots,m\} \quad (1.2.5)$$

są odpowiednio: zbiorami wejść i wyjść wszystkich obiektów składowych systemu S .

W przypadku, gdy wejścia i wyjścia są podzbiorami przestrzeni Euklidesowej, zależność (1.2.4) można przedstawić za pomocą grafu G reprezentującego strukturę połączeń systemu S . Wierzchołki grafu G odpowiadają elementom składowym systemu, krawędzie natomiast odzwierciedlają powiązania pomiędzy poszczególnymi wejściami i wyjściami tych elementów.

Rozbijmy zbiory U, W na podzbiory X, Y tak, aby spełnione zostały warunki

$$\mathcal{P}_X(X) = 0 \wedge \mathcal{P}_Y(Y) = 0 \quad (1.2.6)$$

$$S_i \subset X_i \times Y_i \text{ dla } i=1,2,\dots,n \quad (1.2.7)$$

$$\text{gdzie } X = \{X_i \mid i=1,2,\dots,n\} \text{ oraz } Y = \{Y_i \mid i=1,2,\dots,n\} \quad (1.2.8)$$

Dla pewnej klasy rozważanych w literaturze systemów typu (1.2.2)-(1.2.4) postępowanie takie jest stosunkowo proste w reali-

zacji i pozwala na wyrażenie zależności (1.2.4) za pomocą funkcji macierzowych grafu modelującego strukturę połączeń systemu S. Przykładem takiego podejścia jest zaproponowany przez Kirchhoffa [27] model sieci elektrycznej, w którym S_i jest funkcją postaci:

$$S_i : X_i \longrightarrow Y_i \quad (1.2.9)$$

$$S_i(x_i) = -R_i x_i = y_i \text{ dla } i=1,2,\dots,n : R_i \in \mathbb{R}^+ \quad (1.2.10)$$

natomiast zależność (1.2.6) przyjmuje postać:

$$\mathcal{P}_x(X) = Ax - q = 0 \quad (1.2.11)$$

$$\mathcal{P}_y(Y) = By = 0 \quad (1.2.12)$$

Składowe wektora $x = [x_1, x_2, \dots, x_n]^T$ nazywane są zmiennymi poprzecznymi, wektora $y = [y_1, y_2, \dots, y_n]^T$ - zmiennymi podłużnymi, natomiast wektora $q = [q_1, q_2, \dots, q_w]^T$ - odpływami. Macierze B i A są odpowiednio: $[p \times n]$ wymiarową macierzą obwodów podstawowych oraz macierzą incydencji wymiaru $[w \times n]$, opisującymi digraf bez pętli

$$G = \langle W, E, P \rangle \quad (1.2.13)$$

reprezentujący strukturę połączeń pomiędzy poszczególnymi elementami sieci S. Przy czym

$$W = \{ w_i \mid i=1,2,\dots,w \} \neq \emptyset \quad (1.2.14)$$

jest zbiorem wszystkich wierzchołków sieci, zaś

$$E = \{ \tau_j \mid j=1,2,\dots,N \} \neq \emptyset \quad (1.2.15)$$

jest niepustym zbiorem łuków odwzorowujących elementy S_i , natomiast funkcja

$$P; E \longrightarrow W \times W$$

$$(\forall \tau \in E) (\exists ! \langle w_p, w_k \rangle \in W \times W ; w_p \neq w_k) (P(\tau) = \langle w_p, w_k \rangle) \quad (1.2.16)$$

gdzie: w_p - wierzchołek początkowy, w_k - wierzchołek końcowy łuku .

Dla potrzeb modelowania systemów (1.2.6)-(1.2.8), za pracą [31], przyjmiemy następującą definicję sieci;

DEFINICJA 1.1

Siecią nazywamy uporządkowaną trójkę

$$S = \langle G; \{\nu_i\}; \{\psi_j\} \rangle \quad (1.2.17)$$

gdzie: $\{\nu_i\} = \{\zeta, \pi\}$ - zbiór funkcji $\nu_i: W \rightarrow R$, określonych na zbiorze wierzchołków digrafu G ,

$\{\psi_j\} = \{\psi\}$ - zbiór singletonowy funkcji $\psi_j: E \rightarrow R$, określonych na zbiorze łuków digrafu G .

Funkcje ζ, π, ψ określone są następująco:

$$\zeta: W \rightarrow R \quad (\text{wysokość wierzchołka})$$

$$(\forall w \in W)(\exists \zeta_w \in R)(\zeta(w) = \zeta_w) \quad (1.2.18)$$

$$\pi: W \rightarrow R \quad (\text{potencjał wierzchołka})$$

$$(\forall w \in W)(\exists \pi_w \in R)(\pi(w) = \pi_w) \quad (1.2.19)$$

$$\psi: E \rightarrow R \quad (\text{oporność łuku})$$

$$(\forall \tau \in E)(\exists d_\tau \in R)(\psi(\tau) = d_\tau) \quad (1.2.20)$$

W dalszych rozdziałach pracy, rozważania nasze ograniczymy do pewnej klasy sieci (1.2.17), w których zmienne poprzeczne - x_i oraz zmienne podłużne - y_i , spełniają prawa Kirchhoffa (1.2.11)-(1.2.12). Teoria tego typu sieci, określanych mianem sieci Kirchhoffa [28] rozwijana jest m.in. w ramach zagadnień dotyczących transportu [50], problemów kolorowania [59], przepływów w sieciach [18] czy mechanice [10].

DEFINICJA 1.2

Sieć (1.2.17) nazywamy siecią Kirchhoffa wtedy, gdy spełnione są zależności:

$$Ax - q = 0 \quad (1.2.21)$$

$$By = 0 \quad (1.2.22)$$

$$\text{gdzie: } y_i = d_i \cdot |x_i|^k \cdot \text{sgn}(x_i) + h_i; \quad i=1,2,\dots,n; \quad k \in R^+ \quad (1.2.23)$$

Jeżeli $k \neq 1$, to sieć (1.2.21)-(1.2.23) określać będziemy jako nieliniową sieć Kirchhoffa, przy czym związki pomiędzy zmiennymi poprzecznymi - x_i i podłużnymi - y_i a funkcjami (1.2.18)-(1.2.20), określone są następująco:

$$(\forall \tau \in E)(\exists w_p, w_k \in W)(P(\tau) = \langle w_p, w_k \rangle \Rightarrow \zeta(w_p) - \zeta(w_k) = h_\tau) \quad (1.2.24)$$

$$(\forall \tau \in E)(\exists w_p, w_k \in W)(P(\tau) = \langle w_p, w_k \rangle \Rightarrow \pi(w_k) - \pi(w_p) = y_\tau) \quad (1.2.25)$$

Nietrudno spostrzec, że tak zdefiniowana nieliniowa sieć przepływowa Kirchhoffa stanowi naturalny model matematyczny wielu złożonych systemów wejściowo-wyjściowych typu (1.2.6)-(1.2.8), takich jak obwody elektryczne, hydrauliczne, pneumatyczne czy sieci transportowe. Przy takiej interpretacji sieci (1.2.17), (1.2.21)-(1.2.25) oraz grafu G reprezentującego strukturę jej połączeń, w następnych rozdziałach pracy wykorzystywać będziemy takie pojęcia teorii grafów jak: marszruta, łańcuch, cykl, które zostaną zdefiniowane poniżej.

DEFINICJA 1.3

Marszrutą (r -tą) w grafie G nazywamy dowolny ciąg przemieniny wierzchołków $w_i^r \in W$ oraz łuków $z_j^r \in E$, o postaci

$$m^r = \{w_0^r, z_1^r, w_1^r, z_2^r, \dots, w_{l-1}^r, z_l^r, w_l^r\} \quad (1.2.26)$$

spełniający warunek

$$(\forall k \in \mathbb{N}_1^l) ((P(z_k^r) = \langle w_{k-1}^r; w_k^r \rangle) \vee (P(z_k^r) = \langle w_k^r; w_{k-1}^r \rangle)) \quad (1.2.27)$$

DEFINICJA 1.4

Łańcuchem $E_{j,k}^r$, łączącym wierzchołek początkowy - w_j z wierzchołkiem końcowym - w_k , jest każda marszruta (1.2.26) postaci

$$m^r = \{w_j^r, z_1^r, w_1^r, \dots, z_l^r, w_k^r\} \quad (1.2.28)$$

jeżeli spełniony jest warunek

$$(\forall n, i \in \mathbb{N}_1^l; n \neq i) (z_i^r, z_n^r \in E_{j,k}^r \Rightarrow z_i^r \neq z_n^r) \quad (1.2.29)$$

Dążąc do uproszczenia zapisu, łańcuch $E_{j,k}^r$ określać będziemy za pomocą wektora

$$E_{jk}^r = [z_1^r, z_2^r, \dots, z_{n-1}^r, z_n^r] \quad (1.2.30)$$

przy czym

$$z_i^r = \begin{cases} 1 & \text{jeśli } z_i^r \in E_{jk}^r \text{ i posiada zwrot zgodny ze zwrotem łańcucha} \\ 0 & \text{jeżeli } z_i^r \notin E_{jk}^r \\ -1 & \text{jeśli } z_i^r \in E_{jk}^r \text{ i posiada zwrot przeciwny do zwrotu łańcucha} \end{cases}$$

DEFINICJA 1.5

Długością - l , łańcucha $E_{j,k}^r$ w grafie G , określać będziemy ilość łuków tego łańcucha tj. liczbę elementów niezerowych występujących w wektorze (1.2.30).

Z definicji 1.4 wynika, że krotność występowania dowolnego wierzchołka w łańcuchu jest nieograniczona. Dlatego wśród wszystkich łańcuchów grafu G wyróżnimy łańcuchy proste, określone następująco:

DEFINICJA 1.6

Łańcuchem prostym nazywamy łańcuch (1.2.28)-(1.2.29), którego wszystkie wierzchołki są różne, tj.

$$(\forall n, i \in \mathbb{N}_1^l; n \neq i) (w_i^r, w_n^r \in E_{j,k}^r \implies w_i^r \neq w_n^r) \quad (1.2.31)$$

DEFINICJA 1.7

Cyklem C^r nazywamy łańcuch $E_{j,j}^r$ o długości $l > 0$, postaci

$$m^r = \{w_j^r, z_1^r, w_1^r, \dots, z_l^r, w_j^r\} \quad (1.2.32)$$

Zauważmy, że w ciągu (1.2.32) określającym cykl C^r mamy, $w_0^r = w_1^r$ oraz, że niektóre wierzchołki pośrednie mogą się powtarzać. Natomiast z definicji 1.4 i definicji 1.7 wynika, że wszystkie łuki cyklu są różne. Stąd

DEFINICJA 1.8

Cyklem prostym C^r o długości - l nazywamy taki cykl (1.2.32), dla którego zachodzi

$$(\forall j, i \in \mathbb{N}_1^{l-1}; j \neq i) (w_i^r, w_j^r \in C^r \implies w_i^r \neq w_j^r) \quad (1.2.33)$$

DEFINICJA 1.9

Cyklem podstawowym C_0^r , względem dendrytu D , nazywamy taki graf prosty, który zawiera dokładnie jedną cięciwę tego dendrytu.

Oczywiście odpowiadający notacji (1.2.32) zapis cyklu w formie (1.2.30) w sposób jednoznaczny określa kierunek tworzenia ciągu składającego się na dany cykl podstawowy, który w poniższej pracy przyjmowany będzie zawsze zgodnie ze zwrotem cięciwy dendrytu.

DEFINICJA 1.10

Liczbą stabilności wewnętrznej grafu G - $\alpha(G)$, nazywać

będziemy liczbę wierzchołków najliczniejszego (w sensie wierzchołków) podgrafu pustego rozpatrywanego grafu G .

Przytoczymy teraz, zaczerpniętą z pracy [11], znaną w teorii grafów własność, która wykorzystywana będzie w dalszych częściach pracy.

WŁASNOŚĆ 1.1

Jeżeli A jest $[w \times n]$ wymiarową, węzłowo-gałęziową macierzą incydencji grafu spójnego (1.2.13) o " w " wierzchołkach, to rząd macierzy A jest równy $(w-1)$ ([11] str.188)

$$\text{rank } A = w-1 \quad (1.2.34)$$

Ponieważ, jak wynika z własności macierzy A [11], każda jej kwadratowa podmacierz jest nieosobliwa wtedy i tylko wtedy, gdy odpowiadający jej podgraf jest drzewem, stąd

WNIOSEK 1.1

Jeżeli zredukowaną macierz incydencji A_z , powstałą w wyniku wykreślenia z macierzy A , wiersza odpowiadającego węzłowi odniesienia - z , podzielić na dwie podmacierze

$$A_z = [A_D | A_C]_{[(w-1) \times n]} \quad (1.2.35)$$

gdzie: A_D - składa się z $(w-1)$ kolumn odpowiadających gałęziom dendrytu D

A_C - jest pozostałą podmacierzą odpowiadającą $(n-w+1)$ cięciwom dendrytu D ,

to otrzymana podmacierz A_D jest nieosobliwa ([11] str.189)

$$\det A_D \neq 0 \quad (1.2.36)$$

1.3. Określenie zadania optymalizacji

Niech dana będzie skalarna funkcja celu f określona na wejściowo-wyjściowym systemie (1.2.2)

$$f : S \longrightarrow R \quad (1.3.1)$$

Problem optymalizacji systemu S rozumiany będzie jako zadanie:

$$\begin{aligned} f(U, W) &\longrightarrow \text{extremum} \\ (U, W) &\in S \end{aligned} \quad (1.3.2)$$

W przypadku, gdy S jest nieliniową siecią przepływową (1.2.17), (1,2,21)-(1.2.25), zadanie jej optymalizacji zapiszemy następująco:

$$f(x, y) \longrightarrow \text{extremum} \quad (1.3.3)$$

przy ograniczeniach

$$Ax - q = 0 \quad (1.3.4)$$

$$By = 0 \quad (1.3.5)$$

$$y_i = d_i \cdot |x_i|^k \cdot \text{sgn } x_i + h_i, \quad i = 1, 2, \dots, n; \quad k \in \mathbb{R}^+ \quad (1.3.6)$$

Powyższe zadanie nieliniowej optymalizacji statycznej ze zmiennymi ciągłymi określane w literaturze [62] mianem optymalizacji rozprężu w sieciach nieliniowych (Kirchhoffa), rozpatrywane było przez wielu autorów. Analiza prac dotyczących omawianego problemu pozwala, ze względu na przyjmowaną postać funkcji celu (1.3.3), wyróżnić trzy podstawowe grupy zagadnień optymalizacji sieci przepływowych.

Zagadnienie 1

Przyjęcie funkcji celu (1.3.8) postaci

$$f_1(x, y) = - \sum_{i=1}^n x_i \cdot y_i = -x^T y \quad (1.3.7)$$

prorowadzi do nieliniowego zadania optymalizacji statycznej, rozwiązanie którego pozwala na wyznaczenie, dla ustalonej struktury przestrzennej (1.2.17), optymalnych warunków rozprężu w danej sieci S . Problemowi temu poświęcone są prace [24,33] a omówienie wyników w nich zamieszczonych zawiera rozdział 2.

Zauważmy, że funkcja celu (1.3.7), proporcjonalna do całkowitych strat energii w sieci przepływowej, wyraża moc w niej tracącą, przy czym zgodnie z prawami bilansu energetycznego, mamy

$$\sum_{i=1}^n x_i y_i = x^T y = \sum_{j=1}^w q_j \pi_j = q^T \pi \quad (1.3.8)$$

Zagadnienie 2

Jeżeli funkcję celu (1.3.3) określimy jako:

$$f(x, y, d) = f_1(x, y) + f_2(x, d) \quad (1.3.9)$$

gdzie $d = [d_1, d_2, d_3, \dots, d_n]^T$

to problem (1.3.4)-(1.3.6), (1.3.9) staje się zagadnieniem optymalizacji rozprężu przy zmiennej strukturze przestrzennej danej sieci. Efekt ten uzyskiwany jest poprzez przyjęcie, iż funkcja ψ , występująca w zależności (1.2.17), zależy od wartości parametrów s_i , tj.

$$\psi: \mathbb{L} \times S \longrightarrow \mathbb{R} \quad (1.3.10)$$

$$(\forall l \in \mathbb{L}) (\forall s_t \in S) (\exists d_l \in \mathbb{R}) (\psi(l, s_t) = d_l)$$

gdzie: $S = \{s_i | i=1, 2, \dots, m; s_i - \text{dyskretne}\}$

Przykładem takiego podejścia mogą być prace [24, 32], w których przyjmując funkcję celu

$$f(x, y, s) = - \sum_{i=1}^n d_i \cdot x_i \cdot y_i + \sum_{j=1}^n \alpha_j \frac{x_j^k}{s_j^m} l_j \quad (1.3.11)$$

gdzie: $l_j, \alpha_j, k \in \mathbb{R}^+$ - to stałe parametry,

rozpatrzono zagadnienie optymalizacji struktury i przepływu w sieci S z uwzględnieniem kryterium kosztów. Poszczególne składniki funkcji celu (1.3.11) określają odpowiednio: koszty związane z mocą traconą w sieci oraz koszty zmiany przepustowości poszczególnych łuków.

Zaletą takiego postawienia problemu jest objęcie procesem optymalizacji również struktury połączeń sieci. Jednakże osiągnięte jest to kosztem wzrostu wymiarowości zagadnienia i ilości ograniczeń, co znacznie zwiększa stopień złożoności problemu.

Zagadnienie 3

Problem (1.3.4)-(1.3.6), (1.3.9) ulega znacznej redukcji, jeżeli możliwym jest pominięcie w wyrażeniu (1.3.9) pierwszego składnika funkcji celu. Postępowanie takie znalazło praktyczne zastosowanie m.in. w zagadnieniach z zakresu projektowania i modernizacji sieci, a odpowiada ono ograniczeniu procesu optymalizacji jedynie

do struktury połączeń.

Przykładowo, w pracy [57] przyjmując funkcję celu

$$f_2(x, s) = \sum_{i=1}^n d_i \frac{x_i^k}{s_i^m} l_i \quad (1.3.12)$$

przedstawiono opartą na algorytmie transportowym, metodę wykorzystującą własność funkcji (1.3.12), umożliwiającą rozwiązanie zagadnienia (1.3.4)-(1.3.6), (1.3.12) doboru optymalnych przepustowości łuków oraz struktury połączeń ze względu na kryterium kosztów wyrażonych zależnością (1.3.12).

W pracy niniejszej obszar naszych zainteresowań ograniczymy do Zagadnienia 1. Wybór taki, podyktowany został zarówno dużym zapotrzebowaniem na szybkie i proste algorytmy rozwiązujące omawiane zagadnienie, jak również tym, że znajomość rozwiązania optymalnego problemu (1.3.4)-(1.3.7) w sposób istotny ułatwia rozwiązanie Zagadnienia 2. Umożliwia ona bowiem sprowadzenie zagadnienia 2 do postaci (1.3.4)-(1.3.6), (1.3.12) z dodatkowymi ograniczeniami nieliniowymi otrzymanymi w wyniku rozwiązania problemu (1.3.4)-(1.3.7).

1.4. Sformułowanie celu i tezy rozprawy

Teoria optymalizacji dysponuje szeregiem metod i algorytmów, które z powodzeniem można wykorzystać w celu rozwiązywania różnorodnych zagadnień optymalizacyjnych. Ogólnie jednak, wybór algorytmu rozwiązywania złożonego zadania optymalizacji jest swoistego rodzaju sztuką. Bardzo często korzystnym jest zmodyfikowanie istniejących, bądź opracowanie nowych algorytmów, przystosowanych do konkretnego typu zagadnienia. Postępowanie takie pozwala, poprzez wykorzystanie specyficznych właściwości analizowanego zagadnienia, uprościć proces rozwiązywania, najczęściej drogą zmniejszenia liczby iteracji czy redukcji całkowitego czasu obliczeń.

Mając na uwadze powyższe spostrzeżenia, przyjęto następującą tezę rozprawy:

Uwzględnienie w procesie wyznaczania rozwiązania optymalnego problemu (1.3.4)-(1.3.7), zależności właściwych nieliniowym sieciom przepływowym typu (1.2.17), (1.2.21)-(1.2.25), znacznie

poprawia efektywność stosowanych algorytmów oraz pozwala na opracowanie algorytmów nowych o wyraźnie skróconym czasie trwania obliczeń i zredukowanej zajętości pamięci operacyjnej maszyn cyfrowych.

W pracy wykazana zostanie również celowość zastosowania idei leżącej u podstaw metody prądów składowych i wyrównawczych [38], w celu rozwiązywania zadań postaci (1.3.4)-(1.3.7). Ponadto określone zostaną warunki konieczne i wystarczające optymalności rozwiązania Zagadnienia 1, jak również przedstawione będą algorytmy obliczeniowe wyznaczające zbiór rozwiązań optymalnych tego zagadnienia.

Uwzględniając powyższe, zasadniczy cel niniejszej pracy polega na:

Opracowaniu metody i algorytmów optymalizacji nieliniowych sieci przepływowych typu (1.2.17), (1.2.21)-(1.2.25) ze względu na kryterium minimalizacji strat energetycznych (1.3.7), charakteryzujących się takimi parametrami eksploatacyjnymi, które umożliwiają ich wykorzystanie w procesie projektowania, modernizacji i sterowania pracą rozpatrywanych sieci, zarówno w trybie off-line, jak również w czasie rzeczywistym.

2. ANALIZA STOSOWANYCH METOD OPTYMALIZACJI NIELINIOWYCH SIECI PRZEPŁYWOWYCH W ASPEKCIE TEZY ROZPRAWY

Rozdział ten zawiera krótki przegląd spotykanych w literaturze metod rozwiązywania zagadnień optymalizacji nieliniowych sieci przepływowych, postaci (1.3.3)-(1.3.6). Rezultaty otrzymane w wyniku zastosowania w celu rozwiązania omawianego problemu, znanych metod statycznej optymalizacji nieliniowej, przedstawiono w rozdz. 2.1. Następnie opisano ideę redukcji zagadnienia (1.3.4)-(1.3.7) opublikowaną w pracy [34] już w trakcie pisanja rozprawy, umożliwiającą znaczne zwiększenie efektywności stosowanych algorytmów. W rozdz. 2.2 omówiono także algorytmy opracowane w oparciu o wspomnianą ideę oraz przeprowadzono analizę efektywności omawianych metod w aspekcie tezy rozprawy (rozdz. 2.3).

2.1. Wybrane metody optymalizacji nieliniowych sieci przepływowych

Zagadnieniu rozwiązywania nieliniowych zadań optymalizacji statycznej z ograniczeniami poświęcono w literaturze niemało uwagi.

Analiza dostępnych prac wykazuje, że w zakresie sposobów rozwiązywania wspomnianego zagadnienia istnieje szereg metod i algorytmów umożliwiających otrzymanie rozwiązania zagadnienia (1.3.4)-(1.3.7). Jednakże ze względu na specyfikę rozpatrywanego problemu, jedynie zastosowanie niektórych z nich daje zadowalające efekty obliczeniowe.

Próby rozwiązania zagadnienia (1.3.4)-(1.3.7) w oparciu o metodę Carrola [15] rozwiązywania zadań programowania nieliniowego z ograniczeniami, jak również metodę mnożników Hestenesa [23] zmodyfikowaną przez Schulda [53] przyniosły wynik negatywny.

Metodę Carrola zastosowano w wersji zmodyfikowanej przez Fiaccó i Mc.Cormicka [14], rozszerzając funkcję celu w następujący sposób

$$F(x,y,k) = f_1(x,y) - k \cdot \sum_{i=1}^n q_i^{-1}(x) + 1/\sqrt{k} \sum_{j=1}^{2n} h_j^2(x,y) \quad (2.1.1)$$

gdzie $h_j(x,y)$ - ograniczenia równościowe rozpatrywanego problemu
 $g_i(x)$ - ograniczenia nierównościowe rozpatrywanego problemu

Następnie, iteracyjnie metodą Fletchera-Powella-Davidona [16] minimalizowano funkcję (2.1.1) dla monotonicznie malejących, dyskretnych wartości współczynnika kary "k" skąd dla granicznej wartości $k=0$, osiągnęte zostaje minimum zadania (1.3.4)-(1.3.7).

Z uwagi na fakt, że punkt startowy metody Carrola powinien spełniać ograniczenia (1.3.4)-(1.3.6), jego doboru dokonywano iteracyjnie w oparciu o procedurę Newtona-Raphsona [29] rozwiązywania układów równań nieliniowych.

W metodzie Hestenesa [23] będącej kompilacją metody zewnętrznej funkcji kary z metodą mnożników Lagrange'a, wykorzystano iteracyjnie w ciągu bezwarunkowych optymalizacji również algorytm Fletchera-Powella-Davidona, co ułatwia porównanie obu wspomnianych algorytmów. W procesie optymalizacji zarówno stała kary jak i mnożniki Lagrange'a były uaktualniane po każdej bezwarunkowej minimalizacji, jednak w przeciwieństwie do metody Carrola nie wymagano, aby punkt startowy spełniał ograniczenia (1.3.4)-(1.3.6), co znacznie uprościło proces obliczeń.

Z otrzymanych rezultatów wynika, że zastosowanie powyższych metod bezpośrednio do zagadnienia (1.3.4)-(1.3.7), jakkolwiek możliwe, jest jednak mało efektywne, czasochłonne oraz wymaga

rezerwacji znacznych obszarów pamięci operacyjnej. Główną przyczyną takiego stanu rzeczy wydaje się być duża liczba zmiennych występujących w omawianym zagadnieniu, jak i znaczna ilość ograniczeń.

Inne spojrzenie na sposób rozwiązywania problemu (1.3.4)-(1.3.7) reprezentują prace [33,52], w których w celu wyznaczenia rozwiązania optymalnego omawianego zagadnienia wykorzystuje się metodę dekompozycji oraz metodę podziału zadania, opracowaną przez Geoffriona [20].

Zagadnieniu wykorzystania metod dekompozycji do zadania (1.3.4)-(1.3.7) poświęcona jest praca Rasmusena [52]. Idea przedstawionej w niej metody polega na rozwiązywaniu równań (1.3.4)-(1.3.7) tworzących model matematyczny rozpatrywanej sieci i wyznaczeniu tzw. węzła miarodajnego - w_k , charakteryzującego się najmniejszą wartością potencjału węzłowego $\pi(w_k)$, tj

$$(\forall i=1,2,\dots,W)(\forall w_i \in W)(\pi(w_k) \leq \pi(w_i)) \quad (2.1.2)$$

Następnie na poziomie wyższym, wyznaczane są drogi łączące węzeł miarodajny ze wszystkimi źródłami sieci w ten sposób, aby zapewnić minimalizację strat ponoszonych na dostarczenie dodatkowej ilości przepływu - Δ_j od źródeł do wyznaczonego węzła. Dysponując, najlepszymi z punktu widzenia przyjętego kryterium, drogami - s_i ($i=1,2,\dots,z$) oraz przepływami - x_i ($i=1,2,\dots,n$) spełniającymi (1.3.4)-(1.3.7), na poziomie niższym rozwiązywane jest zagadnienie rozdziału przepływów na poszczególne drogi postaci:

$$f(\Delta) = \sum_{i=1}^n \sum_{j=1}^z |b_{ji}| \cdot |d_i(b_{ji} \cdot \Delta_j + x_i)|^{3+h_i(b_{ji} \cdot \Delta_j + x_i)} \rightarrow \min \quad (2.1.3)$$

przy ograniczeniu

$$\sum_{j=1}^z \Delta_j = 0 \quad (2.1.4)$$

gdzie: $b_{ji} \in \mathbb{R}$ - współczynniki określające przynależność i zwrot i -tego łuku do j -tej drogi.

Otrzymane, nowe wartości przepływów

$$x_i^{(r+1)} = x_i^{(r)} + \sum_{j=1}^z b_{ji} \cdot \Delta_j^{(r+1)}; \quad i=1,2,\dots,n \quad (2.1.5)$$

przekazywane są na poziom wyższy, gdzie stanowią nowy punkt startowy dla procedury rozwiązywania układu równań (1.3.4)-(1.3.7). Po ponownym ich rozwiązaniu wyznaczany jest nowy węzeł miarodajny oraz powtórzona zostaje procedura wyboru optymalnych dróg łączących go ze wszystkimi źródłami sieci itd.

Proces obliczeń zakończony zostaje w chwili, gdy norma z otrzymanego na poziomie niższym wektora

$$\Delta = [\Delta_1, \Delta_2, \dots, \Delta_z]^T \quad (2.1.6)$$

jest dostatecznie mała.

Do wyznaczenia rozwiązania problemu (1.3.4)-(1.3.7) posłużyć się można również opracowanym w 1972 r. przez Geoffriona [20] algorytmem będącym uogólnieniem wyników Bendersa [2] na następującą klasę zadań,

$$f(x, y) \longrightarrow \min \quad (2.1.7)$$

przy ograniczeniach

$$G(x, y) \leq 0 \quad (2.1.8)$$

$$x \in X; y \in Y$$

Metoda Geoffriona opiera się na podziale zbioru zmiennych decyzyjnych na dwie grupy, x, y (w przypadku problemu (1.3.4)-(1.3.7) podział ten jest identyczny jak podział na zmienne poprzeczne i podłużne), a następnie poszukiwaniu rozwiązania optymalnego za pomocą iteracyjnego algorytmu o strukturze dwupoziomowej.

Na nadrzędnym poziomie optymalizacji, dla otrzymanych z poziomu niższego wartości zmiennych $y^{(r)}$, rozwiązywany jest problem poszukiwania optymalnych wartości zmiennych $x^{(r)}$. Wyniki obliczeń poziomu nadrzędnego wykorzystywane są w zadaniu poziomu niższego, gdzie szukane są optymalne wartości $y^{(r+1)}$ itd. Obliczenia te powtarzane są iteracyjnie, aż do momentu, gdy wyniki procesów optymalizacji na obydwu poziomach (tj. wartości funkcji celu) zrównają się z przyjętą dokładnością.

Efektywność obliczeń przedstawionych algorytmów Geoffriona i Rasmusena jest w odniesieniu do problemu (1.3.4)-(1.3.7) lepsza niż metod Carrola czy Hestenesa. Jednakże i w tym przypadku, wykorzystanie ich w celu optymalizacji rozpięty w średnich i dużych sieciach nieliniowych jest ograniczone przede wszystkim przez znaczną czasochłonność obliczeń związaną z dużą ilością ograniczeń.

postaci (1.3.4)-(1.3.5). Z tych samych powodów także inne podejścia do zagadnienia optymalizacji (1.3.4)-(1.3.7), polegające na bezpośrednim wykorzystywaniu metod optymalizacji nieliniowej z ograniczeniami, nie przyniosły zadowalających rezultatów. Wymienić tu można algorytmy opracowane przez Jewdokimowa [24] oraz Haarhoffa i Bayesa [21] oparte na redukcji gradientu rozszerzonej funkcji celu.

Dopiero przedstawiona w pracy [34] redukcja problemu (1.3.4)-(1.3.7) wykorzystująca fakt nieaktywności ograniczeń (1.3.5), połączona z eliminacją części zmiennych, stworzyła możliwość optymalizacji przepływu w sieciach nieliniowych o stosunkowo dużych rozmiarach.

2.2. Optymalizacja sieci przepływowych drogą redukcji zadania wyjściowego

Z przeprowadzone w rozdz. 2.1 przeglądu metod optymalizacji sieci (1.2.17), (1.2.21)-(1.2.23) wynika, że możliwość zastosowania znanych metod statycznej optymalizacji nieliniowej w celu otrzymania rozwiązania zagadnienia (1.3.4)-(1.3.7), uzależniona jest przede wszystkim liczbą ograniczeń oraz wymiarowością rozpatrywanego zagadnienia.

Własności te, charakteryzujące stopień złożoności rozpatrywanego problemu, w sposób istotny wpływają na wymaganą zajętość pamięci operacyjnej maszyn cyfrowych i na całkowity czas obliczeń, decydując o przydatności określonej metody. Powoduje to, że opisana w pracy [34] metoda redukcji problemu (1.3.4)-(1.3.7) nabiera szczególnego znaczenia. Wykorzystanie przez autorów tej pracy specyficznych własności rozpatrywanego zadania umożliwiło im dokonanie szeregu istotnych uproszczeń, prowadząc w konsekwencji do znacznej redukcji czasu trwania obliczeń.

Zasadnicza idea redukcji [34] polega na określeniu dla problemu (1.3.4)-(1.3.7) warunków Khuna-Tuckera [15] i wykazaniu nieaktywności ograniczeń nieliniowych (1.3.5). Na tej podstawie rozpatrywane zadanie sprowadza się do zagadnienia transportowego z nieliniową funkcją celu postaci,

$$f(x, y) = \sum_{i=1}^n x_i \cdot y_i = -x^T y \longrightarrow \min \quad (2.2.1)$$

przy ograniczeniach

$$Ax - q = 0 \quad (2.2.2)$$

gdzie

$$y_i = d_i \cdot |x_i|^k \cdot \text{sgn}(x_i) + h_i; \quad i=1,2,\dots,n; \quad k \in \mathbb{R}^+ \quad (2.2.3)$$

Ponadto fakt, iż występująca w ograniczeniu (2.2.2) macierz A jest macierzą incydencji grafu (1.2.13), umożliwił eliminację części zmiennych zadania wyjściowego.

Dokonano tego poprzez wyznaczenie dowolnego dendrytu D i rozbicie wektorów zmiennych decyzyjnych na dwie grupy $x = [x_1 | x_2]^T$; $y = [y_1 | y_2]^T$, z których pierwsza odpowiada gałęziom dendrytu D , druga natomiast jego cięciwom. Stąd

$$Ax = A[x_1 | x_2]^T = A_D x_1 + A_C x_2 = q \quad (2.2.4)$$

czyli

$$x_1 = A_D^{-1} q - A_D^{-1} A_C x_2 \quad (2.2.5)$$

Ostatnia zależność, wyznaczająca wartości przepływów w łukach tworzących dendryt D w oparciu o przepływy płynące w jego cięciwach, po podstawieniu do zagadnienia (2.2.1)-(2.2.3) pozwoliła na redukcję wymiarowości tego problemu. W ten sposób otrzymano $(n-w+1)$ (n, w - liczby łuków i wierzchołków sieci S) wymiarowe zagadnienie optymalizacji nieliniowej bez ograniczeń.

Zauważmy, że otrzymane zadanie zredukowane pozbawione jest tych niedogodności, które w sposób istotny decydowały o małej efektywności obliczeniowej algorytmów omawianych w rozdziale poprzednim. Transformacja zadania (1.3.4)-(1.3.7) do problemu o mniejszej wymiarowości i całkowicie bez ograniczeń, umożliwiła ponadto wykorzystanie w celu znalezienia rozwiązania optymalnego tego zagadnienia, dowolnej metody rozwiązywania problemów optymalizacji nieliniowej bez ograniczeń.

W pracy [19], rozwiązując omawiane zagadnienie w odniesieniu do sieci hydraulicznych, ze względu na różniczkowalność funkcji celu i tym samym możliwość wykorzystania przy optymalizacji dodatkowych informacji, jakie daje znajomość jej gradientu, zadanie zredukowane rozwiązano przy pomocy metody kierunków sprzężonych opracowanej przez Fletchera-Reevesa [17]. Do wyznaczenia nowego kierunku poszukiwań posłużono się wzorem Polaka-Ribièrego [15], natomiast w celu znalezienia optimum w danym kierunku, wykorzysta-

no metodę aproksymacji funkcji celu wielomianem drugiego stopnia.

Zamieszczone w [19] wyniki, w sposób jednoznaczny wykazują dużą efektywność obliczeń tej metody, znacznie przewyższającą dotychczas omawiane algorytmy. Wynika to, nie tylko z faktu redukcji zagadnienia wyjściowego lecz także z przyjętych, niezwykle prostych, procedur wyznaczania gradientu i wartości zredukowanej funkcji celu.

Inne podejście spotykane w literaturze polega na rozszerzeniu techniki wyznaczania kolejnych kierunków poprawy. W algorytmie Fletchera-Reevesa nowy kierunek poszukiwań określany jest w oparciu o informację o funkcji celu, jakiej dostarcza jej gradient. Postępowanie takie w przypadku rozpatrywanego zagadnienia, podyktowane było prostotą wyznaczania gradientu, a co za tym idzie również prostotą określania kolejnych kierunków sprzężonych. Analizując możliwość wyznaczania kierunków poprawy w pracy [34] opisano wyniki zastosowania do zadania zredukowanego zmodyfikowanej metody Newtona [15]. Każdy kolejny kierunek wyznaczano wykorzystując zarówno gradient jak i dodatkowe informacje o funkcji celu, jakie daje jej hesjan. Wymagało to każdorazowo rozwiązania układu $(n-w)$ równań liniowych postaci

$$\tilde{H}(x) \cdot d = -\nabla \tilde{f}(x) \quad (2.2.6)$$

gdzie: $\tilde{H}(x)$ - hesjan zredukowanej funkcji celu $\tilde{f}$

$\nabla \tilde{f}(x)$ - gradient zredukowanej funkcji celu $\tilde{f}$

$d = [d_1, d_2, \dots, d_{n-w}]^T$ - wektor kolejnych kierunków poprawy,

co spowodowało wzrost czasu trwania obliczeń jednej iteracji, lecz zmniejszając jednocześnie ich ilość, przyniosło dalsze skrócenie całkowitego czasu trwania obliczeń numerycznych.

2.3. Podsumowanie analizy efektywności omówionych metod

Z przeprowadzonej powyżej analizy wynika, że w teorii optymalizacji nieliniowej znane są metody i algorytmy, których wykorzystanie pozwala na rozwiązanie zadania (1.3.4)-(1.3.7).

Specyfiką tego zagadnienia jest jednak znaczna wymiarowość problemu, jak również duża liczba ograniczeń które to cechy w sposób negatywny [24], [34] wpływają na efektywność procesu obliczeń numerycznych. Fakt ten powoduje, że bezpośrednie zastosowanie, w celu znalezienia rozwiązania optymalnego, znanych metod optyma-

lizacji nieliniowej z ograniczeniami, możliwe jest jedynie w przypadku sieci o niewielkich rozmiarach [24]. Dopiero przeprowadzenie redukcji opartej na specyficznych własnościach zadania wyjściowego, prowadzące do znacznego zmniejszenia wymiarowości eliminują tę niedogodność.

Otrzymany problem zredukowany postaci (2.2.1)-(2.2.3), (2.2.5) rozwiązany być może jedną z wielu metod optymalizacji nieliniowej bez ograniczeń, np. zmodyfikowaną metodą Newtona czy metodą Fletchera-Reevesa. Jednakże nawet w tym przypadku, wymagana zajętość pamięci operacyjnej, jak również osiągane czasy obliczeń [34], nie spełniają wymagań jakie stawiane są metodom wykorzystywanym w procesie automatycznego sterowania sieciami przepływowymi.

Podkreślić należy, że jakkolwiek wzrost efektywności obliczeń algorytmów prezentowanych w pracy [34], osiągnięty drogą redukcji zagadnienia, po uprzednim wykorzystaniu własności macierzy A , jest wyraźny, to jednak nie jest on zadowalający z punktu widzenia zapotrzebowań praktycznych. Potwierdzenie tego faktu zawarte zostanie w rozdziałach następnych, gdzie sformułowane będą warunki konieczne i wystarczające optymalności w taki sposób, który umożliwi opracowanie algorytmów konkurencyjnych w stosunku do metod istniejących, zarówno pod względem czasu obliczeń jak również możliwości zastosowań praktycznych.

3. ZASTOSOWANIE METODY PRĄDÓW SKŁADOWYCH I WYRÓWNAWCZYCH DO ROZWIĄZYWANIA ZADAŃ OPTYMALIZACJI NIELINIOWYCH SIECI PRZEPŁYWOWYCH

W rozdziale tym przedstawiono ideę leżącą u podstaw znanej z literatury metody grafów składowych i wyrównawczych oraz zaprezentowano koncepcję uogólnienia tej idei również na nieliniowe sieci przepływowe typu (1.2.17), (1.2.21)-(1.2.23). Ze względu na nieliniowość rozpatrywanego zagadnienia zrezygnowano z zasady superpozycji, a uogólnienie to osiągnięto drogą rozszerzenia warunków koniecznych i wystarczających (3.1.2)-(3.1.5), (3.1.25) sformułowanych w rozdz. 3.1. Następnie, dysponując warunkami koniecznymi i wystarczającymi optymalności zagadnienia nieliniowego postaci (1.3.4)-(1.3.7), określono zasady jego transformacji do układu równań algebraicznych, umożliwiające redukcję wymiarowości rozpatrywanego zagadnienia.

3.1. Metoda prądów składowych i wyrównawczych - warunki konieczne i wystarczające punktu optymalnego

W rozdz. 1.2. zdefiniowano sieć (1.2.17), (1.2.21)-(1.2.23), która ze względów tradycyjno-historycznych określana jest w literaturze mianem sieci Kirchhoffa. Nazwa ta związana jest z zagadnieniem dotyczącym zasad rozprzysięgu prądu w sieciach elektrycznych, sformułowanym przez tego autora w 1847 r. w pracy [27], postaci

$$P(U, J) = -\sum_{i=1}^n U_i J_i = -U^T \cdot J \longrightarrow \min \quad (3.1.1)$$

przy ograniczeniach

$$A \cdot J - I_w = 0 \quad (3.1.2)$$

$$B \cdot U = 0 \quad (3.1.3)$$

gdzie:

$$U_i = -R_i \cdot J_i; \quad R_i \in R^+; \quad i=1, 2, \dots, n \quad (3.1.4)$$

Zmienne podłużne, to napięcia U_i mierzone wzdłuż łuków sieci, natomiast zmiennym poprzecznym odpowiadają prądy J_i płynące w łukach. Ograniczenia (3.1.2)-(3.1.4) to odpowiednio: I, II prawa Kirchhoffa oraz prawo Ohma, zaś każdy z iloczynów $U_i \cdot J_i$ występujących w funkcji celu (3.1.1), wyraża moc traconą na i -tym elemencie sieci.

Dodatkowo przyjęto, że składowe wektora odpływu $I_w = [I_1, I_2, \dots, I_w]^T$, występującego w zależności (3.1.2), spełniają warunek

$$\sum_{j=1}^w I_j = 0 \quad (3.1.5)$$

zapewniający (zgodnie z przyjmowanym w teorii grafów założeniem, że jedna gałąź wiąże nie więcej niż dwa wierzchołki) niesprzeczność układu równań (3.1.2).

Zagadnienie powyższe, z matematycznego punktu widzenia będące liniowym zadaniem optymalizacji statycznej ze zmiennymi ciągłymi jest szczególną postacią problemu (1.3.4)-(1.3.7), której ze względu na liniowość ograniczeń poświęcono w literaturze szereg gruntownych analiz.

W teorii optymalizacji znanych jest także wiele metod oraz algorytmów obliczeniowych wyznaczających rozwiązanie problemu

(3.1.1)-(3.1.5). Wspólną podstawę dla większości z nich stanowi zaproponowana przez Kirchhoffa metoda prądów składowych i wyrównawczych [28], której idea opiera się na spostrzeżeniu, że wartość zmiennej poprzecznej J_i , płynącej w i -tej gałęzi można przedstawić w formie sumy dwóch składników

$$J_i = J_i^W + J_i^S \quad \text{dla } i=1,2,\dots,n \quad (3.1.6)$$

gdzie: składnik pierwszy J_i^W , nazywany prądem wyrównawczym, zależy od potencjałów źródeł, natomiast prąd składowy J_i^S związany jest z zaspokojeniem potrzeb odbiorców.

Wynik powyższy otrzymano rozpatrując wycinek sieci elektrycznej (rys.3.1), dla którego zgodnie z definicją potencjału i prawem Ohma (3.1.4), mamy

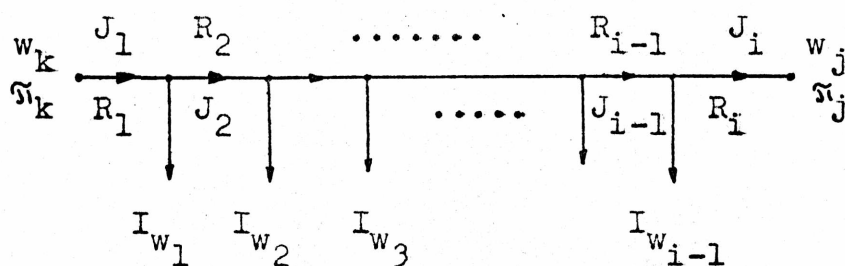
$$\pi_k - \pi_j = \sum_{l=1}^i R_l J_l \quad (3.1.7)$$

a stąd, po uwzględnieniu (3.1.2), otrzymamy równość

$$\pi_k - \pi_j = J_i R_{jk} - \sum_{l=1}^{i-1} R_{lk} I_{wl} \quad (3.1.8)$$

skąd

$$J_i = \frac{\pi_k - \pi_j}{R_{jk}} + \frac{\sum_{l=1}^{i-1} R_{lk} I_{wl}}{R_{jk}} = J_i^W + J_i^S \quad (3.1.9)$$



Rys.3.1. Zamknięty odcinek sieci elektrycznej

Zgodnie z ostatnią zależnością prąd wyrównawczy J_i^W jest funkcją różnicy potencjałów źródeł, natomiast prąd składowy J_i^S zależy jedynie od wartości składowych wektora odpływów I_w . Zatem, jeżeli

dla danego problemu (3.1.1)-(3.1.5), wektor I_w jest ustalony, wtedy wektor prądów składowych $J^S = [J_1^S, J_2^S, \dots, J_n^S]^T$ nie ulega zmianie w czasie procesu optymalizacji. Minimalizacja strat określonych warunkiem (3.1.1) przebiega więc przy ustalonych wartościach składowych wektora J^S i polega na poszukiwaniu optymalnych wartości składowych wektora $J^W = [J_1^W, J_2^W, \dots, J_n^W]^T$, określonych zależnością (3.1.9).

W oparciu o warunek (3.1.9), korzystając z zasady superpozycji, w pracy [27] sformułowano równoważność:

$$(\forall i \in N_1^n)(J_i^W = 0) \iff (w_j, w_k \in W_Z)(\pi_k - \pi_j = 0) \quad (3.1.10)$$

gdzie: $W_Z = \{w_i | i=1, 2, \dots, z\} \neq \emptyset$ jest zbiorem węzłów będących źródłami

stąd

LEMAT 3.1

Na to, żeby para wektorów $(\hat{U}, \hat{J})$ była rozwiązaniem optymalnym zagadnienia (3.1.1)-(3.1.5), potrzeba i wystarcza, aby spełniała ona następujący układ równań

$$A \cdot \hat{J}^S - I_w = 0 \quad (3.1.11)$$

$$B \cdot \hat{U} = 0 \quad (3.1.12)$$

$$\hat{U}_i = -R_i \cdot \hat{J}_i^S; \quad R_i \in R^+; \quad i=1, 2, \dots, n \quad (3.1.13)$$

Dowód; (Część 1 ($\Rightarrow$))

Niech wektory $\hat{U}, \hat{J}$ będą rozwiązaniem optymalnym zagadnienia (3.1.1)-(3.1.5). Funkcja Lagrange'a rozważanego problemu przyjmie postać

$$L(J, \lambda, \beta) = J^T R_d J + \lambda^T (AJ - I_w) + \beta^T (B \cdot R_d \cdot J) \quad (3.1.14)$$

gdzie: R_d $_{[n \times n]}$ - macierz diagonalna o elementach $r_{ii} = -R_i$ oraz $r_{ij} = 0$ dla $i \neq j$

λ, β - wektory mnożników Lagrange'a, przy czym na mocy (3.1.5) mamy

$$(\forall i \in N_1^Z)(\forall w_i \in W)(\lambda_i = 0) \quad (3.1.15)$$

Zgodnie z teorią mnożników Lagrange'a wektor $\hat{J}$ spełnia warunki:

$$\frac{\partial L(\hat{J}, \hat{\lambda}, \hat{\beta})}{\partial \hat{J}} = 2R_d \cdot \hat{J} + A^T \hat{\lambda} + R_d B^T \hat{\beta} = 0 \quad (3.1.16)$$

$$\frac{\partial L(\hat{J}, \hat{\lambda}, \hat{\beta})}{\partial \hat{\lambda}} = A \cdot \hat{J} - I_w = 0 \quad (3.1.17)$$

$$\frac{\partial L(\hat{J}, \hat{\lambda}, \hat{\beta})}{\partial \hat{\beta}} = B \cdot R_d \cdot \hat{J} = 0 \quad (3.1.18)$$

Obliczając $\hat{J}$ z warunku (3.1.16) otrzymamy

$$\hat{J} = 0,5 \cdot R_d^{-1} (A^T \hat{\lambda} + R_d B^T \hat{\beta}) \quad (3.1.19)$$

co wraz z (3.1.18) daje

$$B R_d R_d^{-1} A^T \hat{\lambda} + B R_d R_d^{-1} R_d B^T \hat{\beta} = 0 \quad (3.1.20)$$

a stąd, korzystając ze znanej z teorii grafów [6] własności ortogonalności macierzy A i B , mamy

$$B \cdot R_d B^T \hat{\beta} = 0 \quad (3.1.21)$$

co wobec zależności (3.1.9) (o dodatnich wartościach R_i , $\{i=1,2,\dots,n\}$) pociąga za sobą

$$(\forall j=1,2,\dots,(n-w+1)) (\hat{\beta}_j = 0) \quad (3.1.22)$$

Uwzględnienie powyższej zależności w równaniu (3.1.16) daje

$$2R_d \cdot \hat{J} + A^T \hat{\lambda} = 0 \quad (3.1.23)$$

skąd, wymnażając lewostronnie powyższą zależność przez wektor $E_{j,k}^r$, określający łańcuch prosty (1.2.31), otrzymamy

$$(\forall j,k \in N_1^Z) (\forall w_j, w_k \in W) (E_{j,k}^r \cdot R_d \hat{J} = E_{j,k}^r \cdot \hat{U} = 0) \quad (3.1.24)$$

co wobec zależności (1.3.25) i równoważności (3.1.10), kończy dowód części 1 ($\Rightarrow$).

Część 2 ($\Leftarrow$)

Dowód dostateczności, wobec faktu że funkcja celu (3.1.1) jest wypukła a ograniczenia (3.1.2)-(3.1.5) liniowe, jest oczywiste ([15] str.62)

($\times \times \times$)

Zgodnie z powyższym lematem, każdemu rozwiązaniu optymalnemu problemu (3.1.1)-(3.1.5) odpowiada rozwiązanie układu równań

(3.1.11)–(3.1.13) i odwrotnie. Zatem, w celu otrzymania rozwiązania zagadnienia (3.1.1)–(3.1.5) wystarczy rozwiązać układ równań liniowych, co stanowiąc istotne uproszczenie zagadnienia, pozwala na znaczne skrócenie czasu obliczeń. Warto zauważyć również, że z omawianego lematu wynikają następujące wnioski.

Wniosek 3.1

Zbiór rozwiązań optymalnych problemu (3.1.1)–(3.1.5) jest jednoelementowy ([3] str.166)

Wniosek 3.2

Zbiór warunków koniecznych i wystarczających rozwiązania optymalnego zagadnienia (3.1.1)–(3.1.5) tworzą równania (3.1.2)–(3.1.5) oraz zależność

$$(\forall j, k \in \mathbb{N}_1^Z)(\forall w_j, w_k \in W)(\pi_j - \pi_k = \text{const} = 0) \quad (3.1.25)$$

Dowód:

Z (3.1.6), dla rozwiązania optymalnego, mamy

$$\hat{J}_i = \hat{J}_i^W + \hat{J}_i^S \quad \text{dla } i=1,2,\dots,n$$

skąd na mocy (3.1.13), (3.1.4) otrzymujemy

$$\hat{U}_i = -R_i \hat{J}_i = -R_i \hat{J}_i^W - R_i \hat{J}_i^S = -R_i \hat{J}_i^S \quad i=1,2,\dots,n$$

a stąd

$$\hat{J}_i^W = 0 \quad \text{dla } i=1,2,\dots,n$$

co wraz z równoważnością (3.1.10) kończy dowód

(* * *)

Drugi z przedstawionych wniosków prezentuje ujęcie metody prądów składowych i wyrównawczych odbiegające od przedstawionego w pracach [28], [38], przede wszystkim z powodu zastąpienia pierwszego członu równoważności (3.1.10) warunkiem (3.1.25). Podejście takie związane jest z faktem, iż w nieliniowych sieciach przepływowych (1.3.17), (1.2.21)–(1.2.23) nie obowiązuje zasada superpozycji i podział na przepływy: składowy i wyrównawczy ma znaczenie jedynie teoretyczne. Natomiast, jak zostanie to wykazane w rozdz. 3.2, poszukiwane rozwiązanie zagadnienia (1.3.1)–(1.3.7) optymalizacji sieci nieliniowych, zawsze spełnia warunek będący naturalnym rozszerzeniem zależności (3.1.25).

3.2. Uogólnienie metody prądów składowych i wyrównawczych w celu redukcji problemu wyjściowego

Jest rzeczą oczywistą, iż ze względu na nieliniowość rozpatrywanego problemu, zastosowanie zasady superpozycji w celu otrzymania rozwiązania optymalnego jest wykluczone. W tej sytuacji również wyodrębnianie przepływów składowych i wyrównawczych, tak ułatwiające otrzymanie rozwiązania optymalnego w przypadku sieci liniowych, praktycznie jest mało efektywne i posiada jedynie znaczenie teoretyczne. Jednakże, jeśli powrócić do Lematu 3.1 i dokładnie przeanalizować Część 1 ($\Rightarrow$) jego dowodu, to nietrudno zauważyć, iż zasada superpozycji w sposób istotny wykorzystywana była przede wszystkim w jego ostatniej fazie (poprzez skorzystanie z równoważności (3.1.10)). Tak więc zależność (3.1.25) otrzymano zakładając jedynie, że para wektorów $(\hat{U}, \hat{J})$ jest optymalnym rozwiązaniem zagadnienia (3.1.1)-(3.1.3) będącego zawężeniem problemu (1.3.4)-(1.3.7) do przypadku sieci liniowych.

W związku z powyższym nasuwa się przypuszczenie, że rozpatrywany warunek (3.1.25) jest szczególną postacią zależności bardziej ogólnej, będącej warunkiem koniecznym optymalności rozwiązania zagadnienia (1.3.4)-(1.3.7).

W celu potwierdzenia tej hipotezy utwórzmy funkcję Lagrange'a problemu (1.3.4)-(1.3.7) postaci:

$$L(x, y, \lambda, \beta) = f_1(x, y) + \lambda^T(Ax - q) + \beta^T(By) \quad (3.2.1)$$

gdzie: $\lambda \in \mathbb{R}^w$, $\beta \in \mathbb{R}^p$ - wektory mnożników Lagrange'a, przy czym wektor λ jest postaci

$$\lambda = [0, 0, \dots, 0, \lambda_{z+1}, \dots, \lambda_w]^T \quad (3.2.2)$$

Zgodnie z teorią mnożników Lagrange'a [15] zbiór warunków koniecznych punktu optymalnego wyznaczają zależności:

$$\frac{\partial L(\hat{x}, \hat{y}, \hat{\lambda}, \hat{\beta})}{\partial x_i} = (k+1)\hat{y}_{i+h_i} + \sum_{z=1}^w a_{zi}\hat{\lambda}_z + \sum_{j=1}^p \hat{\beta}_j b_{ji}\hat{y}_i = 0; \quad (3.2.3)$$

$i=1, 2, \dots, n$

$$\frac{\partial L(\hat{y}, \hat{\lambda}, \hat{\beta}, \hat{x})}{\partial \lambda_z} = \sum_{i=1}^n a_{zi} \cdot \hat{x}_i = 0, \quad z=1, 2, \dots, w \quad (3.2.4)$$

$$\frac{\partial L(\hat{x}, \hat{y}, \hat{\lambda}, \hat{\beta})}{\partial \beta_j} = \sum_{i=1}^n b_{ji} \cdot \hat{y}_i = 0, \quad j=1, 2, \dots, p \quad (3.2.5)$$

przy czym

$$\tilde{y}_i = y_i - h_i \quad \text{oraz} \quad y_i' = \frac{dy_i}{dx_i} \quad i=1, 2, \dots, n \quad (3.2.6)$$

Ponadto, dla rozpatrywanych sieci przepływowych, prawdziwe są następujące uwagi,

Uwaga 3.1

Dla dowolnej wartości przepływu x_i ($i=1, 2, \dots, n$) zachodzi,

$$y_i' = \frac{dy_i}{dx_i} \leq 0, \quad i=1, 2, \dots, n \quad (3.2.7)$$

Dowód:

Na mocy warunku (1.2.20) mamy

$$d_i < 0, \quad i=1, 2, \dots, n \quad (3.2.8)$$

skąd

$$\begin{aligned} y_i' = \frac{dy_i}{dx_i} &= (d_i \cdot |x_i|^k \cdot \text{sgn}(x_i) + h_i)' = \\ &= [d_i \cdot k \cdot |x_i|^{k-1} \text{sgn}(x_i)] \cdot \text{sgn}(x_i) = \\ &= d_i \cdot k \cdot |x_i|^{k-1} \cdot \text{sgn}^2(x_i) \leq 0 \end{aligned}$$

co wobec faktu, iż rozumowanie przeprowadzone zostało dla dowolnego wskaźnika i , kończy dowód.

(* * *)

Uwaga 3.2.

Dla każdego wektora h , o składowych $h_1, h_2, \dots, h_n$ określonych zależnością (1.2.24), niezależnie od wyboru dendrytu D prawdziwym jest warunek:

$$B_D \cdot h = 0 \quad (3.2.9)$$

Dowód: (Nie wprost)

Przyjmijmy, że zależność (3.2.9) nie jest spełniona, tj.

$$B_D \cdot h \neq 0 \quad (3.2.10)$$

Zgodnie z określeniem macierzy obwodów podstawowych B_D , mamy

$$B_D = [c_D^1, c_D^2, \dots, c_D^p]^T \quad (3.2.11)$$

gdzie: c_D^j - jest n -wymiarowym wektorem określającym, zgodnie z (1.2.30) i definicją 1.9, cykl podstawowy zawierający j -tą cięciwę dendrytu D .

Korzystając z ostatniej zależności warunek (3.2.10) zapiszemy następująco:

$$c_D^j \cdot h \neq 0 \quad \text{dla } j=1, 2, \dots, p \quad (3.2.12)$$

czyli

$$\sum_{i=1}^{l_j} h_i \neq 0 \quad j=1, 2, \dots, p \quad (3.2.13)$$

gdzie: $\sum_{i=1}^{l_j}$ - to operator sumowania po wszystkich łukach tworzących cykl j -ty o długości l_j .

W myśl definicji cyklu (zgodnie z którą każdy cykl jest łańcuchem), w oparciu o (1.2.24), warunek (3.2.13) przyjmie postać

$$\sum_{i=1}^{l_j} [\zeta(w_{i-1}) - \zeta(w_i)] \neq 0 \quad \text{dla } j=1, 2, \dots, p \quad (3.2.14)$$

a stąd

$$\zeta(w_0) - \zeta(w_{l_j}) \neq 0 \quad \text{dla } j=1, 2, \dots, p \quad (3.2.15)$$

co po uwzględnieniu faktu, że dla dowolnego cyklu c_D^j zachodzi

$$w_0 = w_{l_j} \quad \text{dla } j=1, 2, \dots, p \quad (3.2.16)$$

prowadzi do sprzeczności, kończąc dowód

($\times \times \times$)

Wyliczając $\hat{y}_i$ z zależności (3.2.9) oraz korzystając z uwagi 3.2, po podstawieniu uzyskanego wyniku do (3.2.5) otrzymamy

$$\frac{1}{k+1} \sum_{i=1}^n b_{ji} \left[\sum_{z=1}^w \hat{\lambda}_z a_{zi} + k \sum_{j=1}^p \hat{\beta}_j b_{ji} \hat{y}_i \right] = 0 \quad \text{dla } j=1, 2, \dots, p \quad (3.2.17)$$

czyli

$$\frac{1}{k+1} \left[\sum_{i=1}^n b_{ji} \sum_{z=1}^w \hat{\lambda}_z a_{zi} \right] + \frac{k}{k+1} \left[\sum_{i=1}^n b_{ji} \sum_{j=1}^p \hat{\beta}_j b_{ji} \hat{y}_i \right] = 0, \quad j=1,2,\dots,p \quad (3.2.18)$$

Zauważmy, że pierwszy składnik zależności (3.2.18) stanowi j -tą składową wyrażenia $B \cdot A^T \cdot \hat{\lambda}$, przy czym z teorii grafów [6] wiadomo, że,

$$B \cdot A^T = 0 \quad (3.2.19)$$

Stąd warunek (3.2.18) przyjmie postać

$$\sum_{i=1}^n b_{ji} \sum_{j=1}^p b_{ji} \hat{\beta}_j \hat{y}_i = 0, \quad j=1,2,\dots,p \quad (3.2.20)$$

Zauważmy, że zależność powyższa określa jednorodny układ równań liniowych względem wektora $\hat{\beta}$, przy czym rząd macierzy tworzącej ten układ jest równy p . Stąd mamy

$$\hat{\beta}_j = 0, \quad j=1,2,\dots,p \quad (3.2.21)$$

Otrzymany warunek pozwala na znaczne uproszczenie zagadnienia (1.3.4)-(1.3.7), drogą eliminacji ograniczeń nieaktywnych (1.3.5), jak to zrobiono w pracy [34], lub w sposób jaki zostanie przedstawiony poniżej.

W tym celu, dla potrzeb dalszych rozważań, wykażemy prawdziwość następujących uwag;

Uwaga 3.3

Niech $\mathbb{E}_{j1}^r$ będzie n -wymiarowym wektorem określającym r -ty łańcuch prosty łączący dowolne dwa wierzchołki w_j, w_1 będące źródłami. Wtedy

$$(\forall w_j, w_1 \in W_Z) (\forall r \in S_{j1}) (\mathbb{E}_{j1}^r \cdot h = \sum_{\mathbb{E}_{j1}}^r h_i = \zeta(w_j) - \zeta(w_1)) \quad (3.2.22)$$

gdzie: $\sum_{\mathbb{E}_{j1}}^r$ - jest operatorem sumowania po wszystkich łukach tworzących r -ty łańcuch prosty łączący wierzchołki w_j z wierzchołkiem w_1 ,

S_{j1} - zbiór wszystkich łańcuchów prostych łączących wierzchołki w_j oraz w_1 .

Dowód:

Rozpatrzmy dowolny łańcuch $\mathbb{E}_{j1}^r$ łączący źródła w_j oraz w_1 .
Na mocy definicji 1.4 oraz w oparciu o zależność (1.2.24) zachodzi

$$\mathbb{E}_{j1}^r \cdot h = \sum_{\mathbb{E}_{j1}}^r h_i = \sum_{\mathbb{E}_{j1}}^r \left[\zeta(w_{i-1}) - \zeta(w_i) \right] = \zeta(w_p) - \zeta(w_k) \quad (3.2.23)$$

Ponieważ jednak

$$w_p = w_j, \quad w_k = w_1 \quad (3.2.24)$$

stąd

$$\mathbb{E}_{j1}^r \cdot h = \sum_{\mathbb{E}_{j1}}^r h_i = \zeta(w_j) - \zeta(w_1) \quad (3.2.25)$$

a więc Uwaga 3.3 jest udowodniona

(***)

Uwaga 3.4

Dla każdego łańcucha $r \in \mathbb{S}_{j1}$ określonego wektorem $\mathbb{E}_{j1}^r$, łączącego dowolne dwa źródła w_j oraz w_1 , spełniony jest warunek

$$(\forall w_j, w_1 \in W_Z)(\forall r \in \mathbb{S}_{j1})(\mathbb{E}_{j1}^r \cdot A^T \lambda = \sum_{\mathbb{E}_{j1}}^r \sum_{z=1}^W \lambda_z a_{zi} = 0) \quad (3.2.26)$$

Dowód:

Zgodnie z przyjętym założeniem (1.2.16), że jeden łuk łączy nie więcej niż dwa wierzchołki, kolumna - a_i macierzy A , odpowiadająca i -tej gałęzi zawiera tylko dwa elementy niezerowe (tj. 1, -1) na tych pozycjach, które odpowiadają wierzchołkom w_k i w_p tej gałęzi. Stąd

$$\mathbb{E}_{j1}^r \cdot A^T \cdot \lambda = \sum_{\mathbb{E}_{j1}}^r \sum_{z=1}^W \lambda_z a_{zi} = \sum_{\mathbb{E}_{r1}}^r (\lambda_p^i - \lambda_k^i) \quad (3.2.27)$$

Powracając do definicji łańcucha prostego $\mathbb{E}_{j1}^r$, otrzymamy

$$\sum_{\mathbb{E}_{j1}}^r (\lambda_p^i - \lambda_k^i) = \sum_{\mathbb{E}_{j1}}^r (\lambda_{i-1} - \lambda_i) = \lambda_j - \lambda_1 \quad (3.2.28)$$

co wobec warunku (3.2.2) daje zależność (3.2.26) i kończy dowód.

(***)

Obecnie, mając na uwadze warunek (3.2.21) oraz korzystając z zależności (3.2.3), mamy

$$(\forall w_j, w_1 \in W_z)(\forall r \in S_{j1}) \left(\sum_{i=1}^r \frac{\partial \hat{L}}{\partial x_i} = \sum_{i=1}^r \left[(k+1) \hat{y}_i + h_i + \sum_{z=1}^w \hat{\lambda}_z a_{zi} \right] = 0 \right) \quad (3.2.29)$$

czyli

$$(\forall w_j, w_1 \in W_z)(\forall r \in S_{j1}) \left(\sum_{i=1}^r (k+1) y_i + \sum_{i=1}^r h_i + \sum_{i=1}^r \sum_{z=1}^w \hat{\lambda}_z a_{zi} = 0 \right) \quad (3.2.30)$$

stąd zgodnie z Uwagą 3.1 oraz Uwagą 3.2, otrzymamy

$$(\forall w_j, w_1 \in W_z)(\forall r \in S_{j1}) \left((k+1) \sum_{i=1}^r \hat{y}_i = \zeta(w_j) - \zeta(w_1) \right) \quad (3.2.31)$$

oraz

$$(\forall w_j, w_1 \in W_z)(\forall r \in S_{j1}) \left(\sum_{i=1}^r \hat{y}_i = \frac{\zeta(w_j) - \zeta(w_1)}{k+1} \right) \quad (3.2.32)$$

Jeżeli teraz skorzystamy z określenia potencjału węzłowego π_w , to na mocy warunku (1.2.25)

$$(\forall w_j, w_1 \in W_z)(\forall r \in S_{j1}) \left(\hat{\pi}_{w_1} - \hat{\pi}_{w_j} = \frac{\zeta(w_j) - \zeta(w_1)}{k+1} = \text{const} \right) \quad (3.2.33)$$

gdzie

$$\hat{\pi}(w_i) = \pi(w_i) + \zeta(w_i), \quad i=1, 2, \dots, w \quad (3.2.34)$$

nazywany będzie względnym potencjałem węzłowym.

Otrzymany warunek (3.2.33) jest poszukiwaną prawidłowością, o której wspomniano na początku tej części rozdziału. Nietrudno dostrzec, że zależność (3.1.25) jest szczególną postacią tego warunku, powstałą przez przyjęcie założenia

$$(\forall i \in N_1^w)(\forall w_i \in W)(\zeta(w_i) = 0) \quad (3.2.35)$$

co rzeczywiście miało miejsce w przypadku problemu (3.1.1)-(3.1.4) omawianego w trakcie prezentacji metody prądów składowych i wyrównawczych. Fakt ten, jak również przedstawione powyżej rozumowanie, będące bezpośrednim potwierdzeniem hipotezy odnośnie możliwości rozszerzenia Wniosku 3.2 także na nieliniowe sieci przepływowe, prowadzą do następującego twierdzenia;

Twierdzenie 3.1

Dla każdej pary wektorów $(\hat{x}, \hat{y})$ będącej optymalnym rozwiązaniem zagadnienia (1.3.4)-(1.3.7), różnica względnych potencjałów węzłowych $\hat{\pi}_w$, wyrażona sumą algebraiczną zmiennych podłużnych y_1 , liczoną wzdłuż dowolnego łańcucha łączącego każde dwa źródła, jest wielkością stałą, tzn.

$$(\forall w_j, w_1 \in W_z) (\forall r \in S_{j1}) (\hat{\pi}_{w_1} - \hat{\pi}_{w_j} = L_{j1}^r \hat{y} = \frac{\zeta(w_j) - \zeta(w_1)}{k+1} = \text{const}) \quad (3.2.36)$$

Dowód:

Wynika bezpośrednio z rozumowania przedstawionego powyżej.

($\times \times \times$)

Otrzymane twierdzenie dotyczy jedynie warunku koniecznego istnienia punktu optymalnego zagadnienia (1.3.4)-(1.3.7). Jak wiadomo spełnienie zależności (3.2.36) przez daną parę wektorów (x, y) , w ogólnym przypadku nie jest równoznaczne z optymalnością danego rozwiązania, a także nie daje podstaw do wnioskowania, że rozwiązanie takie istnieje. Niezbędnym jest więc sformułowanie warunków wystarczających, których spełnienie zapewniałoby otrzymanie rozwiązania optymalnego.

Wykażemy teraz, że warunek (3.2.36) tworzy wraz z zależnościami (1.3.4)-(1.3.6) zbiór warunków koniecznych i wystarczających optymalności rozwiązania zagadnienia (1.3.4)-(1.3.7). W tym celu skorzystamy z następującego lematu zaczerpniętego z pracy [15].

Lemat 3.2

Niech $f: X \rightarrow R$ posiada drugie pochodne cząstkowe $\frac{\partial^2 f(x)}{\partial x_i \partial x_j}$ ($i, j=1, 2, \dots, n$). Warunkiem wystarczającym istnienia minimum funkcji $f(x)$ jest to, aby dla wartości x spełniających warunki konieczne istnienia minimum, hesjan

$$H(x) = \frac{\partial^2 f(x)}{\partial x^T \partial x} = \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j} \right]_{n \times n} \quad (3.2.37)$$

był dodatnio określony ([15] str.36), tzn.

$$(\forall x \in R^n; x \neq 0) (x^T H(x) x > 0) \quad (3.2.38)$$

Podstawiając do funkcji celu (1.3.7) równania (1.3.6), określające wzajemne powiązania pomiędzy odpowiednimi składowymi wekto-

rów, zmiennych poprzecznych - x oraz podłużnych - y , otrzymujemy

$$f(x) = - \sum_{i=1}^n (d_i \cdot |x_i|^k \cdot \text{sgn}(x_i) + h_i) \cdot x_i \quad (3.2.39)$$

a po przekształceniu

$$f(x) = \sum_{i=1}^n (-d_i |x_i|^{k+1} - h_i x_i) \quad (3.2.40)$$

Stąd hesjan funkcji (3.2.40) zawiera elementy postaci

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \begin{cases} -d_i \cdot k \cdot (k+1) \cdot |x_i|^{k-1} & \text{dla } i=j \\ 0 & \text{dla } i \neq j \end{cases} \quad (3.2.41)$$

czyli jest macierzą posiadającą elementy różne od zera jedynie na głównej przekątnej. Oznacza to, że wyrażenie $x^T H x$ dla funkcji (3.2.40) przyjmie postać

$$x^T H(x) x = \sum_{i=1}^n -d_i \cdot k(k+1) \cdot |x_i|^{k+1} \quad (3.2.42)$$

Zgodnie z powyższym, warunek (3.2.38) dodatniej określoności hesjanu (3.2.41) będzie spełniony, jeżeli wszystkie wartości parametrów d_i ($i=1,2,\dots,n$) są ujemne, a to zapewnione jest zależnością (1.2.20). Stąd

TWIERDZENIE 3.2

Funkcja celu (1.3.7) zagadnienia optymalizacji (1.3.4)-(1.3.7) osiąga minimum dla tych wartości wektorów $(\hat{x}, \hat{y})$, które spełniają układ równań (1.3.4)-(1.3.6), (3.2.36).

Dowód:

Ponieważ wszystkie wartości parametrów d_i są niedodatnie, stąd

$$(\forall x \in \mathbb{R}^n) (x^T H(x) x = \sum_{i=1}^n [(-d_i) k(k+1) |x_i|^{k+1}] \geq 0) \quad (3.2.43)$$

W szczególności więc, pomijając przypadek trywialny mamy

$$(\forall x \in \mathbb{R}^n; x \neq 0) (x^T H(\hat{x}) x > 0) \quad (3.2.44)$$

co w świetle Lematu 3.2 kończy dowód.

(***)

Twierdzenie powyższe nie dostarcza informacji o tym, które z uzyskanych rozwiązań $(\hat{x}, \hat{y})$ układu równań (1.3.4)-(1.3.6), (3.2.36) stanowią minima lokalne, a które są globalnymi. Nieprecyzuje również liczby tych rozwiązań. Jednakże w literaturze sformułowane zostały warunki przesadzające te wątpliwości. Pamiętając o nieaktywności ograniczeń (1.3.5), przytoczymy za pracą [15] lemat rozstrzygający zagadnienie pierwsze;

Lemat 3.3

Dany jest problem minimalizacji

$$f(x) \longrightarrow \min \quad (3.2.45)$$

przy ograniczeniach

$$g_i(x) = 0 \quad i = 1, 2, \dots, m \quad (3.2.46)$$

Jeżeli ograniczenia (3.2.46) są liniowe, a funkcja (3.2.45) wypukła, to w punktach, dla których spełnione są warunki;

$$\frac{\partial L(x, \lambda)}{\partial x_i} = 0, \quad i = 1, 2, \dots, n \quad (3.2.47)$$

$$\frac{\partial L(x, \lambda)}{\partial \lambda_i} = 0, \quad i = 1, 2, \dots, m \quad (3.2.48)$$

$$\lambda_j \geq 0, \quad i = 1, 2, \dots, m \quad (3.2.49)$$

osiągane jest minimum globalne funkcji (3.2.45) w rozpatrywanym obszarze.

W oparciu o lemat 3.3, wobec zależności (3.2.21) mamy

Wniosek 3.3

Funkcja celu (1.3.7) zagadnienia optymalizacji (1.3.4)-(1.3.7) osiąga minimum globalne dla tych wartości wektorów $(\hat{x}, \hat{y})$, które spełniają układ równań (1.3.4)-(1.3.6), (3.2.36).

Dowód:

Niech wektory $\hat{x}, \hat{y}$ będą rozwiązaniem układu równań (1.3.4)-(1.3.6), (3.2.36), wtedy wobec nieaktywności ograniczeń (1.3.5), problem (1.3.4)-(1.3.7) spełnia założenia lematu 3.3, a więc funkcja celu (1.3.7) osiąga w punktach $(\hat{x}, \hat{y})$ minimum globalne, co kończy dowód.

(x x x)

W ten sposób, uogólniając zasady leżące u podstaw metody superpozycji prądów, otrzymano warunki będące rozszerzeniem Lematu 3.1, oraz Wniosków 3.1, 3.2 na przypadek nieliniowych sieci przepływowych (1.2.17), (1.2.21)-(1.2.23).

Zauważmy, że Twierdzenie 3.2 wraz z wnioskami z niego wypływającymi umożliwia sprowadzenie zadania optymalizacji (1.3.4)-(1.3.7) do zagadnienia rozwiązywania układu równań algebraicznych (1.3.4)-(1.3.6), (3.2.36). Tego typu postępowanie znane jest w teorii optymalizacji, a typowym jego przykładem może być metoda mnożników Lagrange'a. Niestety, w przypadku zagadnienia (1.3.4)-(1.3.7) metoda ta okazuje się zawodna, bowiem ze względu na dużą wymiarowość problemu i znaczną ilość ograniczeń, rozwiązanie układu $2n$ równań (3.2.3)-(3.2.5), pomimo istnienia szeregu metod iteracyjnych, jest uciążliwe.

Z tego punktu widzenia, sprowadzenie nieliniowego zagadnienia (1.3.4)-(1.3.7) do problemu rozwiązania układu równań postaci (1.3.4)-(1.3.6), (3.2.36) należy uznać za korzystne, bowiem w wyniku takiego postępowania następuje podwójna redukcja ilości równań koniecznych do rozwiązania, tak iż ostatecznie stajemy przed problemem rozwiązywania $(n+z-w)$ równań nieliniowych, który rozpatrzony zostanie w rozdziale 4.

3.3. Zasady optymalizacji nieliniowych sieci przepływowych w oparciu o transformację zadania wyjściowego do układu równań algebraicznych

W rozdziale poprzednim uzyskano warunek (3.2.36) umożliwiający sprowadzenie zagadnienia optymalizacji (1.3.4)-(1.3.7) do problemu rozwiązywania układu równań nieliniowych (1.3.4)-(1.3.6) określających strukturę rozpatrywanej sieci, przy jednoczesnym spełnieniu s -równań nieliniowych wynikających z zależności (3.2.36). Z utilitarnego punktu widzenia postępowanie takie jest uzasadnione jedynie wtedy, gdy możliwe jest zadawalająco szybkie i efektywne znalezienie rozwiązania nowo powstałego układu. Jednakże ilość równań wynikających z zależności (3.2.36) równa jest sumie wszystkich, różnych dróg łączących każde dwa źródła $w_j, w_i \in W_z$, czyli

$$s = \sum_{i=1}^Z \sum_{\substack{j=1 \\ j \neq i}}^Z \text{card } S_{ij} \quad (3.3.1)$$

gdzie: $\text{card } S_{ij}$ określa moc zbioru S_{ij} ,

a jak wiadomo z teorii grafów skierowanych, moc zbioru wszystkich dróg łączących dwa dowolne wierzchołki, w przypadku grafów wielooczkowych przyjmuje duże wartości.

Z tej przyczyny, wykorzystanie w procesie optymalizacji warunków koniecznych i wystarczających w postaci (1.3.4)-(1.3.6), (3.2.36) jest niecelowe. Powodem tego jest także duża ilość równań nieliniowych, których rozwiązywanie, jeśli nawet nie wymaga się nadmiernej dokładności, zawsze związane jest ze znacznymi stratami czasu. Dlatego, sprawą niezwykle istotną jest problem redukcji ilości równań występujących w rozpatrywanym układzie.

Obecnie, zostaną określone reguły transformacji zależności (1.3.4)-(1.3.6), (3.2.36) do układu równań typu (1.2.21)-(1.2.23), a następnie opiszemy metodę jego redukcji opartej na specyficznych własnościach macierzy obwodów podstawowych B oraz węzłowo-gałęziowej macierzy incydencji A .

Procedura transformacji polega na zastąpieniu równań wynikających z warunku (3.2.36) układem $(z-1)$ zależności typu (1.2.22), przy czym w procesie tym wykorzystywane jest następujące spostrzeżenie;

Uwaga 3.5

Oznaczmy przez analogię do (1.2.24)

$$h_{ij} = \frac{\zeta(w_i) - \zeta(w_j)}{k+1} \quad ; \quad w_i, w_j \in W_Z \quad (3.3.2)$$

Jeżeli S jest nieliniową siecią przepływową (1.2.17), (1.2.21)-(1.2.23), wtedy zależność (3.2.36) oraz warunek

$$(\forall w_i, w_j \in W_Z) (\exists r \in S_{ij}) (x_{ij}^r = h_{ij}) \quad (3.3.3)$$

są sobie równoważne.

Dowód:

Przypomnijmy, że obwody podstawowe względem dendrytu D określone są n -wymiarowymi, liniowo niezależnymi wektorami C_D^i i tworzą macierz

$$B_D = [c_D^1, c_D^2, \dots, c_D^i, \dots, c_D^p]^T \quad (3.3.4)$$

stąd korzystając z (1.2.22) mamy

$$(\forall i \in N_1^p)(c_D^i \cdot y = 0) \quad (3.3.5)$$

Ogólnie w sieci S istnieje $l \geq p$ różnych obwodów, z których każdy jest obwodem podstawowym, bądź kombinacją liniową obwodów podstawowych. Z tego powodu warunek (3.3.5) pociąga za sobą również prawdziwość zależności

$$(\forall i \in N_1^l)(c_D^i \cdot y = 0) \quad (3.3.6)$$

Przyjmijmy, że istnieje łańcuch E_{ij}^1 , taki że

$$E_{ij}^1 \cdot \hat{y} - h'_{ij} = 0 \quad (3.3.7)$$

Rozpatrując dowolny obwód C_D^i zawierający łańcuch E_{ij}^1 , zgodnie z (3.3.6) i definicją łańcucha, mamy

$$C_D^i \cdot \hat{y} = 0 \quad \text{oraz} \quad E_{ij}^r = -E_{ji}^r \quad (3.3.8)$$

Rozbicie obwodu C_D^i na łańcuchy E_{ij}^1 oraz E_{ij}^2 , gdzie E_{ij}^2 określa jedną z dróg r S_{ij} , daje nam;

$$(E_{ij}^1 + E_{ji}^2) \cdot \hat{y} = E_{ij}^1 \cdot \hat{y} + E_{ji}^2 \cdot \hat{y} = 0 \quad (3.3.9)$$

a stąd, wobec zależności (3.3.7) otrzymujemy

$$E_{ij}^2 \cdot \hat{y} - h'_{ij} = 0 \quad (3.3.10)$$

co, wobec faktu iż E_{ij}^2 jest dowolnym elementem zbioru S_{ij} , kończy dowód.

($\times \times \times$)

Na mocy powyższej uwagi spełnienie warunku (3.2.36) wymaga rozwiązania układu równań, z których każde odpowiada jednej z dróg łączących dwa dowolne elementy zbioru W_Z . Naturalnie ilość takich równań wynosi

$$s = \binom{z}{2} = \frac{z!}{2!(z-2)!} = \frac{z(z-1)}{2}; \quad z = \text{card } W_Z \quad (3.3.11)$$

lecz zgodnie z (1.2.8), (3.3.2) oraz (3.2.9) tylko $(z-1)$ tworzy układ równań niezależnych. Jak wiadomo, odpowiadają one gałęziom

dowolnego dendrytu D łączącego wszystkie źródła $w_1, w_2, \dots, w_z \in W_z$. W szczególności więc, dla dendrytu powstałego przez połączenie źródła pierwszego z pozostałymi $(z-1)$ źródłami, z (3.2.36) otrzymamy

$$L_{1j} \hat{y} - h'_{1j} = 0, \quad j=2,3,\dots,z \quad (3.3.12)$$

czyli

$$\begin{bmatrix} \mathcal{L} & | & 1_d \end{bmatrix} \cdot \begin{bmatrix} \hat{y} \\ y_{n+1} \\ \vdots \\ y_{n+e} \end{bmatrix} = 0 \quad (3.3.13)$$

$$\text{gdzie: } \mathcal{L}_{[exn]} = [L_{1,2}, L_{1,3}, L_{1,4}, \dots, L_{1,z}]^T \quad y_{n+i} = h'_{1,i+1} \quad (3.3.14)$$

$$i=1,2,\dots,e$$

natomiast macierz 1_d jest $[exe]$ wymiarową, diagonalną macierzą jedynekową ($e=z-1$).

Twierdzenie 3.3

Niech $\hat{x}' = [\hat{x} | x_{n+1}, \dots, x_{n+e}]^T$ oraz $\hat{y}' = [\hat{y} | y_{n+1}, \dots, y_{n+e}]^T$, gdzie $\hat{x}, \hat{y}$ określone zależnościami (1.3.4)-(1.3.6), (3.2.36) są rozwiązaniem optymalnym zagadnienia (1.3.4)-(1.3.7). Wektory $\hat{x}', \hat{y}'$ spełniają układ równań

$$A' x' - q' = 0 \quad (3.3.15)$$

$$B' y' = 0 \quad (3.3.16)$$

$$y'_i = d_i \cdot |x'_i|^k \cdot \text{sgn}(x'_i) + h_i, \quad i=1,2,\dots,n+e \quad (3.3.17)$$

wtedy i tylko wtedy, gdy macierz A' oraz B' są postaci;

$$A' = \begin{bmatrix} A_{[w \times n]} & \begin{bmatrix} -1^T [1 \times e] \\ 1_d \\ 0 \end{bmatrix} \end{bmatrix} \quad B' = \begin{bmatrix} B_{[p \times n]} & 0 \\ \mathcal{L}_{[exn]} & 1_d [exe] \end{bmatrix} \quad (3.3.18)$$

Dowód:

Wynika z rozumowania poprzedzającego twierdzenie. Zauważmy jedynie, że rozpisując w oparciu o (3.3.18), zależność (3.3.15), otrzymamy wektor

$$q = [q_1 + \sum_{i=1}^z x_{n+i}, q_2 - x_{n+1}, q_3 - x_{n+2}, \dots, q_z - x_{n+e}, q_{z+1}, \dots, q_w]^T \quad (3.3.19)$$

występujący w zależności (1.3.4).

(x x x)

Przedstawione twierdzenie, określające zasady transformacji układu równań (1.3.4)-(1.3.6), (3.2.36) do postaci (1.2.21)-(1.2.23), posłuży jako podstawa do algorytmizacji tego procesu. Wykorzystanie zależności (3.3.18) wymaga jednak opracowania metody wyznaczania macierzy $\mathcal{L}$ (3.3.14), co wiąże się z dodatkowymi nakładami czasowymi. Z tego powodu opracowano procedurę transformacji pozwalającą na wyznaczanie macierzy A', B' bez konieczności wyznaczania macierzy łańcuchów (3.3.14). Opiera się ona na spostrzeżeniu, że określanie macierzy A', B' w oparciu o wcześniej zbudowane macierze A, B (zgodnie z (3.3.18)), zastąpić można procedurą wyznaczania tych macierzy drogą transformacji struktury sieci S .

W tym celu należy:

PROCEDURA 1. (Transformacja)

Krok 1. Wyznacz zbiory (1.2.14), (1.2.15) wierzchołków i łuków optymalizowanej sieci przepływowej S oraz określ relację (1.2.16) i funkcje $\xi(w), \psi(\tau), \pi(w)$

Krok 2. Rozszerz zbiór (1.2.15) łuków rozpatrywanej sieci o podzbiór

$$E' = \{ \tau_j \mid j=n+1, n+2, \dots, n+e \} \quad (3.3.20)$$

przy czym spełniony powinien być warunek

$$(\forall \tau \in E') (\exists ! i \in N_2^Z) (P(\tau) = \langle w_1, w_i \rangle) \quad (3.3.21)$$

Krok 3. Każdemu nowemu łukowi $\tau \in E'$ przyporządkuj wielkości h_τ oraz d_τ , zgodnie z zależnością

$$(\forall \tau \in E' \mid P(\tau) = \langle w_1, w_i \rangle) (h_\tau = h_{1i}' = \frac{\xi(w_1) - \xi(w_i)}{k+1} \wedge d_\tau = 0) \quad (3.3.22)$$

Krok 4. Koniec procedury transformacji.

Korzystając z otrzymanego, rozszerzonego w Procedurze 1, modelu struktury przestrzennej sieci S' , można wyznaczyć macierz obwodów podstawowych B' oraz macierz incydencji A' , występujące w zależnościach (3.3.15)-(3.3.16). Przestrzeganie powyższych reguł transformacji gwarantuje otrzymanie macierzy A', B' o strukturach określonych zależnością (3.3.18) ponieważ:

- warunek (3.3.21) zapewnia, że wszystkie elementy $\tau \in E'$ tworzą drzewo identyczne z tym, które stanowiło podstawę sformułowania zależności (3.3.13),

- na mocy zależności (3.3.22), (3.3.17) mamy

$$y'_{n+i} = h'_{1,i+1} \quad \text{dla } i=1,2,\dots,e \quad (3.3.23)$$

co odpowiada warunkowi (3.3.14).

Zauważmy, że otrzymamy w wyniku transformacji, układ (3.3.15)-(3.3.17) posiada niezmiennie istotną własność związaną z ortogonalnością macierzy A', B' , która umożliwia dalszą redukcję liczby równań, które należy rozwiązać.

Lemat 3.4

Przyjmijmy, że macierze A, B są odpowiednio: $[w \times n]$ -wymiarową, węzłowo-gałęziową macierzą incydencji oraz $[p \times n]$ -wymiarową macierzą obwodów podstawowych pewnego grafu G . Jeżeli wektor x^0 jest rozwiązaniem układu równań

$$Ax - q = 0 \quad (3.3.24)$$

to dla dowolnego wektora $\Delta^i \in \mathbb{R}^p$, zachodzi

$$A(x^0 + B^T \Delta^i) - q = 0 \quad (3.3.25)$$

Dowód:

Z założenia mamy

$$Ax^0 - q = 0 \quad (3.3.26)$$

a stąd także

$$Ax^0 - q + \tilde{O} \Delta^i = 0 \quad (3.3.27)$$

gdzie $\tilde{O}$ jest macierzą zerową wymiaru $[w \times p]$.

Korzystając z ortogonalności macierzy A, B [22], z warunku (3.3.27) otrzymamy

$$Ax^0 - q + AB^T \Delta^i = 0 \quad (3.3.28)$$

a więc

$$A(x^0 + B^T \Delta^i) - q = 0 \quad (3.3.29)$$

co kończy dowód.

(* * *)

Przed przystąpieniem do omówienia wniosków wpływających z powyższego lematu, przypomnijmy, że rozwiązanie układu równań nieliniowych posiada zwykle wiele pierwiastków i w przypadku ogólnym znalezienie ich wymaga żmudnych obliczeń. Często określenie

pełnego zbioru rozwiązań jest wręcz praktycznie niemożliwe nawet przy wykorzystaniu maszyn cyfrowych. W pracy [3] wykazano, że mimo nieliniowości równań tworzących układ (3.3.15)-(3.3.17), prawdziwym jest następujący lemat:

Lemat 3.5

Oznaczmy przez R zbiór rozwiązań układu równań nieliniowych (3.3.15)-(3.3.17), tj.

$$R = \{ (\hat{x}, \hat{y})_i \mid i=1, 2, \dots, t \} \quad (3.3.30)$$

Jeżeli układ równań (3.3.15)-(3.3.17) jest niesprzeczny, wtedy ([3], str.166)

$$\text{card } R = 1 \quad (3.3.31)$$

Istnienie co najwyżej jednego rozwiązania układu (3.3.15)-(3.3.17) znacznie upraszcza rozpatrywane zagadnienie, umożliwiając wykorzystanie w procesie obliczeń szeregu znanych w literaturze metod iteracyjnego rozwiązywania układów równań nieliniowych (np. metody Newtona-Raphsona). Jednakże wiadomo [51], że maksymalna efektywność obliczeń uzyskana zostanie jedynie wtedy, gdy procedury te zastosowane zostaną do rozwiązywania układu złożonego z możliwie najmniejszej ilości równań. Dlatego też, zauważmy że podstawiając równanie (3.3.17), określające zależności pomiędzy składowymi wektorów zmiennych poprzecznych - x i podłużnych - y , do równania (3.3.16) otrzymamy, odpowiadający (3.3.15)-(3.3.17), układ $(n+e)$ równań liniowych z niewiadomymi $x_1, x_2, \dots, x_n, x_{n+1}, \dots, x_{n+e}$, postaci

$$\sum_{i=1}^{n+e} a'_{zi} x'_i - q'_i = 0, \quad z = 1, 2, \dots, (w-1) \quad (3.3.32)$$

$$\sum_{i=1}^{n+e} (b'_{ji} \cdot d_i \cdot |x'_i|^k \cdot \text{sgn } x'_i - h'_i) = 0, \quad j=1, 2, \dots, p' \quad (3.3.33)$$

Jeżeli teraz, w celu rozwiązania powyższego układu zastosujemy metodę iteracyjną, w której nowe wartości przybliżeń obliczane są wg zależności

$$x^{(n+1)} = x^{(0)} + B'^T \Delta^{(n)} \quad (3.3.34)$$

gdzie $\Delta^{(n)} \in R^{(p+e)}$ - wektor poprawek po n -tej iteracji,

to przyjmując za punkt startowy wektor $x^{(0)}$, spełniający równania (3.3.32), na mocy lematu 1.4, otrzymamy problem zredukowany z niewiadomymi $\Delta_1, \dots, \Delta_{p+e}$, postaci

$$\begin{aligned} \sum_{i=1}^{n+e} b_{1i} \left[\pm d_i(x_i^{(0)}) + \sum_{j=1}^{p+e} b_{ji} \Delta_j \right]^k + h_1 &= 0 \\ \dots\dots\dots & \\ \sum_{i=1}^{n+e} b_{pi} \left[\pm d_i(x_i^{(0)}) + \sum_{j=1}^{p+e} b_{ji} \Delta_j \right]^k + h_i &= 0 \end{aligned} \quad (3.3.35)$$

który po przekształceniach przyjmie formę

$$\begin{aligned} g_1(\Delta) &= \sum_{j=1}^{p+e} \sum_{i=0}^k G_{1j}^{(i)} (\Delta_j)^i = 0 \\ \dots\dots\dots & \\ g_p(\Delta) &= \sum_{j=1}^{p+e} \sum_{i=0}^k G_{pj}^{(i)} (\Delta_j)^i = 0 \end{aligned} \quad (3.3.36)$$

gdzie; $G_{lj}^{(i)} \in \mathbb{R}$, $p' = p+e$

Z Lematu 3.4 oraz Lematu 3.5 wynika ponadto,

Wniosek 3.4

Układ równań nieliniowych (3.3.36) posiada (w przypadku istnienia) dokładnie jedno rozwiązanie $\hat{\Delta}$; przy czym jeżeli $x^{(0)} \in \mathbb{R}^{n+e}$ jest dowolnym wektorem spełniającym (3.3.15), to rozwiązanie układu (3.3.15)-(3.3.17) określone jest warunkiem

$$(\forall x^{(0)} \in \mathbb{R}^{n+e}) (\exists \hat{\Delta} \in \mathbb{R}^{p+e}) (\hat{x} = x^{(0)} + B^T \hat{\Delta}) \quad (3.3.37)$$

Dowód:

Niech $\hat{\Delta}$ będzie rozwiązaniem układu (3.3.36), a więc także (3.3.35). Stąd wektor $\hat{y}$ określony zależnością (3.3.17) (po uwzględnieniu (3.3.34)) spełnia równania (3.3.16).

Ponieważ $x^{(0)}$ spełnia (3.3.15), stąd na mocy Lematu 3.4 również $\hat{x}$ jest jego rozwiązaniem, co dowodzi prawdziwości warunku (3.3.37).

Istnienie co najwyżej jednego rozwiązania wynika natomiast bezpośrednio z Lematu 3.5, co kończy dowód.

(* * *)

Przedstawione lematy i wnioski dowodzą, że wykorzystanie warunków koniecznych i wystarczających (3.3.15)-(3.3.17) optymalności zagadnienia (1.3.4)-(1.3.7) znacznie upraszcza proces optymalizacji, pod warunkiem, że w algorytmie obliczeniowym uwzględniona zostanie procedura transformacji umożliwiająca otrzymanie układu równań (3.3.36).

Sformułowany w rozdz. 1.3, problem nieliniowej optymalizacji statystycznej (1.3.4)-(1.3.7) z "n" ograniczeniami, sprowadzono do zagadnienia rozwiązywania układu $(n-w+z)$ równań nieliniowych (3.3.36). W ten sposób znacznemu zmniejszeniu uległa wymiarowość rozpatrywanego zadania, jak również skrócony został czas trwania obliczeń, który jak wiadomo, zależny jest w sposób istotny od ilości równań tworzących rozwiązywany układ.

Opisana procedura transformacji charakteryzuje się tak dużą prostotą, że wykonanie jej bez pomocy maszyny cyfrowej, nawet w przypadku bardzo dużych sieci przepływowych, nie nastrocza większych trudności.

Nie bez znaczenia jest również fakt, że otrzymany w wyniku transformacji układ równań, umożliwia wykorzystanie w procesie optymalizacji znanych i oprogramowanych metod rozwiązywania układów równań nieliniowych, bez potrzeby wprowadzania zmian w istniejących programach obliczeniowych.

4. ALGORYTMY MINIMALNO-ENERGETYCZNEJ OPTYMALIZACJI SIECI PRZEPŁYWOWYCH OPARTE NA POSTACI ZREDUKOWANEJ PROBLEMU WYJŚCIOWEGO

W rozdziale poprzednim dokonano transformacji i redukcji zadania optymalizacji nieliniowej (1.3.4)-(1.3.7) do układu nieliniowych równań algebraicznych (3.3.36). Postępowanie takie wymaga jednak, aby dla przyjętej metody rozwiązywania układu równań tej postaci spełniona była zależność (3.3.34). Obecnie przedstawiona zostanie metoda Crossa-Łobaczewa iteracyjnego rozwiązywania układów równań, spełniająca ten warunek, a następnie opisane zostaną: procedura redukcji i procedura przyspieszająca otrzymanie rozwiązania optymalnego, zwiększająca efektywność obliczeń omawianej metody. Po przedstawieniu procedur budowy macierzy A, B zaprezentowane zostaną dwa, opracowane przez autora, algorytmy obliczeniowe przeznaczone do rozwiązywania zagadnień postaci (1.3.4)-(1.3.7). Pierwszy, zalecany w sytuacji, gdy nie dysponujemy modelem danej sieci i konieczna jest jej symulacja. Drugi dostosowany do wymogów automatycznego sterowania w trybie on-line, rzeczywistą siecią przepływową. W rozdz. 4.4 sformułowane

zostały warunki zbieżności, a rozdz. 4.5 poświęcono omówieniu wyników przeprowadzonych badań testowych opracowanych algorytmów. Na zakończenie w rozdz. 4.6 podano przykłady obliczeniowe prezentujące działanie obydwu algorytmów.

4.1. Metoda Crossa-Łobaczewa iteracyjnego rozwiązywania układów równań nieliniowych

Zgodnie z zależnościami przedstawionymi w rozdz. 3.3, poszukując rozwiązań n -wymiarowego problemu optymalizacji (1.3.4)-(1.3.7) należy rozwiązać układ $(n-w+z)$ równań postaci (3.3.36).

Ze względu na warunek (3.3.34), w celu rozwiązania otrzymanego w wyniku transformacji układu algebraicznych równań nieliniowych, wykorzystany zostanie algorytm opracowany przez Crossa [10] i Łobaczewa [36]. O wyborze tej właśnie metody zadecydowała duża prostota algorytmu, połączona z zadowalającą efektywnością obliczeń oraz fakt, iż uwzględniając zasady redukcji, umożliwia ona bezpośrednie otrzymanie rozwiązania problemu (1.3.4)-(1.3.7). Ponadto metoda ta jest znana i powszechnie stosowaną w dziedzinach, w których omawiany problem optymalizacji jest szczególnie istotny.

Metoda Crossa-Łobaczewa [36] jest jedną z szeregu znanych w literaturze p '-wymiarowych modyfikacji metody Newtona rozwiązywania równań nieliniowych [51], stąd kolejne przybliżenia rozwiązania układu równań (3.3.36) obliczane są (podobnie jak w metodzie Newtona) za pomocą jednopunktowych wzorów iteracyjnych

$$\Delta^{(r+1)} = \Delta^{(r)} - \varphi_d^{(r)}(\Delta^{(r)}) \cdot g^{(r)}(\Delta^{(r)}) = \psi^{(r)}(\Delta^{(r)}) \quad (4.1.1)$$

gdzie: $\varphi_d^{(r)}$ - jest macierzą diagonalną postaci

$$\varphi_d^{(r)} = \begin{bmatrix} \varphi_1^{(r)} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \varphi_p^{(r)} \end{bmatrix} \quad (4.1.2)$$

przy czym zakłada się, że elementy $\varphi_j^{(r)}$ są funkcjami analitycznymi w pewnym otoczeniu poszukiwanego rozwiązania $\hat{\Delta}$. Natomiast $g^{(r)}(\Delta^{(r)})$ jest wektorem o składowych określonych zależnością (3.3.36).

Korzystając z ostatniej zależności, rozwiązanie $\hat{\Delta}$ obliczamy jako

$$\hat{\Delta} = \lim_{r \rightarrow \infty} \Delta^{(r)} = \Delta_0 + \lim_{r \rightarrow \infty} \sum_{i=1}^r \delta^{(i)} \quad (4.1.3)$$

podstawiając

$$\delta^{(i)} = \Delta^{(i)} - \Delta^{(i-1)} \quad (4.1.4)$$

Oczywiście zgodnie z (3.3.34) mamy również,

$$x_i^{(r)} = x_i^{(0)} + \sum_{j=1}^{p+e} b_{ji} \Delta_j^{(r-1)}, \text{ dla } i=1, 2, \dots, (n+e) \quad (4.1.5)$$

oraz

$$\hat{x} = x_0 + B^T \hat{\Delta} \quad (4.1.6)$$

gdzie $\hat{x}$ jest rozwiązaniem problemu optymalizacji (1.3.4)-(1.3.7).

Metoda Crossa-Łobaczewa polega więc na iteracyjnym rozwiązywaniu równań nieliniowych (3.3.36) przy wykorzystaniu zależności (4.1.1) i po przyjęciu za funkcję

$$\varphi_j^{(r)}(\Delta^r) = \left[k \cdot \sum_{i=1}^{p+e} |b_{ji}| \cdot d_i |x_i^{(r)}|^{(k-1)} \right]^{-1} \quad j=1, 2, \dots, (p+e) \quad (4.1.7)$$

a stąd, wzór iteracyjny (4.1.1) przyjmie postać

$$\Delta_j^{(r+1)} = \Delta_j^{(r)} + \delta_j^{(r)} = \varphi_j^{(r)}(\Delta^{(r)}), \quad j=1, 2, \dots, (p+4) \quad (4.1.8)$$

gdzie

$$\delta_j^{(r)} = - \frac{\sum_{i=1}^{p'} b_{ji} [d_i |x_i^{(r)}|^k \cdot \text{sgn}(x_i^{(r)}) + h_i]}{k \cdot \sum_{i=1}^{p'} d_i |b_{ji}| \cdot |x_i^{(r)}|^{k-1}}, \quad j=1, 2, \dots, (p+e) \quad (4.1.9)$$

Naturalnie otrzymane zależności są funkcjami wektora $\Delta^{(r)}$ wyrażonego za pomocą $x^{(r)}$, a w celu otrzymania (4.1.7), (4.1.8) jako jawnej funkcji wektora $\Delta^{(r)}$ należy w zależnościach tych uwzględnić wzór (4.1.5). Zauważmy również, że ostatnią zależność otrzymujemy, pomijając wzajemne oddziaływania składowych wektora a oraz rozwijając równania układu (3.3.36) w szereg Taylora, w otoczeniu punktu rozwiązania i odrzuceniu reszty zawierającej wyrazy z Δ w stopniu wyższym niż pierwszy.

Oznaczmy przez $\xi^{(r)}$ p' -wymiarowy wektor niedokładności przybliżenia rozwiązania $\hat{\Delta}$ w r -tej iteracji, wtedy

$$\xi^{(r)} = \hat{\Delta} - \mathcal{U}^{(r-1)}(\hat{\Delta}^{(r-1)}) = \hat{\Delta} - \mathcal{U}^{(r-1)}(\hat{\Delta} - \xi^{(r-1)}) \quad (4.1.10)$$

Rozwijając prawą stronę powyższego warunku w szereg Maclaurina w otoczeniu punktu $\hat{\Delta}$ i pomijając wyrazy w stopniu wyższym niż pierwszy, otrzymamy

$$\xi^{(r)} = \hat{\Delta} - \mathcal{U}^{(r-1)}(\hat{\Delta}) + \Phi'(\hat{\Delta}) \xi^{(r-1)} \quad (4.1.11)$$

gdzie $\Phi'(\hat{\Delta})_{[p' \times p']}$ jest macierzą pochodnych cząstkowych liczoną w punkcie rozwiązania, o elementach

$$\Phi'_{ij}(\hat{\Delta}) = \left. \frac{\partial \mathcal{U}_i}{\partial \Delta_j} \right|_{\Delta = \hat{\Delta}} \quad i, j = 1, 2, \dots, (p+e) \quad (4.1.12)$$

Korzystając z (4.1.1), dla punktu rozwiązania $\hat{\Delta}$, mamy

$$\mathcal{U}^{(r)}(\Delta) \big|_{\Delta = \hat{\Delta}} = \hat{\Delta} \quad \text{dla } r \in \mathbb{N} \quad (4.1.13)$$

stąd, zależność (4.1.11) przyjmuje postać

$$\xi^{(r)} = \Phi'(\hat{\Delta}) \cdot \xi^{(r-1)} \quad r \in \mathbb{N} \quad (4.1.14)$$

Wartości elementów $\Phi'_{ij}(\hat{\Delta})$ obliczone na mocy (4.1.1) określone są wzorem

$$\Phi'_{ij}(\Delta) = \delta_{ij} - \varphi'_{ij}(\Delta) g_i(\Delta) - \varphi_i(\Delta) g'_{ij}(\Delta) \quad i, j = 1, 2, \dots, (p+e) \quad (4.1.15)$$

gdzie: δ_{ij} -delta Kroneckera, natomiast

$$\varphi'_{ij} = \frac{\partial \varphi_i}{\partial \Delta_j} \quad \text{i} \quad g'_{ij} = \frac{\partial g_i}{\partial \Delta_j} \quad i, j = 1, 2, \dots, (p+e) \quad (4.1.16)$$

Wyrażenie (4.1.15) dla $\Delta = \hat{\Delta}$ przyjmie więc postać

$$\Phi'_{ij}(\hat{\Delta}) = \delta_{ij} - \varphi_i(\hat{\Delta}) g'_{ij}(\hat{\Delta}) \quad i, j = 1, 2, \dots, (p+e) \quad (4.1.17)$$

a stąd, zgodnie z (4.1.7), (3.3.35), (3.3.36)

$$\Phi'_{ij}(\Delta) = \delta_{ij} - \frac{\sum_{s=1}^{p+e} b_{is} \cdot b_{js} \cdot d_s \cdot |\hat{x}_s|^{k-1}}{\sum_{s=1}^{p+e} b_{is} \cdot d_s \cdot |\hat{x}_s|^{k-1}} \quad i, j = 1, 2, \dots, (p+e) \quad (4.1.18)$$

Dysponując zależnościami (4.1.11), (4.1.18) oszacujmy stałą asymptotyczną błędu C metody Crossa-Łobaczewa oraz określmy rząd tej metody. W tym celu przyjmijmy, że istnieje granica (4.1.3), a rząd zbieżności metody określony jest zależnością [15]

$$\lim_{r \rightarrow \infty} \frac{\|\Delta^{(r+1)} - \hat{\Delta}\|}{\|\Delta^{(r)} - \hat{\Delta}\|^p} = \lim_{r \rightarrow \infty} \frac{\|\xi^{(r+1)}\|}{\|\xi^{(r)}\|^p} = C \neq 0 \quad (4.1.19)$$

W przypadku metody Crossa-Łobaczewa mamy, dla $p=1$

$$\lim_{r \rightarrow \infty} \frac{\|\xi^{(r+1)}\|}{\|\xi^{(r)}\|} = \lim_{r \rightarrow \infty} \frac{\|\xi^{(r)} \Phi'(\hat{\Delta})\|}{\|\xi^{(r)}\|} = C \quad (4.1.20)$$

stąd, w oparciu o nierówność Schwarza, otrzymujemy

$$C = \lim_{r \rightarrow \infty} \frac{\|\xi^{(r+1)}\|}{\|\xi^{(r)}\|} \leq \lim_{r \rightarrow \infty} \|\Phi'(\hat{\Delta})\| = \|\Phi'(\hat{\Delta})\| \quad (4.1.21)$$

przy czym, za pracę [51] przyjmijmy, że norma macierzy $\Phi'(\hat{\Delta})$ występująca w ostatniej zależności, rozumiana będzie w sensie normy spektralnej określonej następująco;

DEFINICJA 4.1

Niech A będzie macierzą prostokątną o wymiarach $[n \times n]$. Normą spektralną macierzy A nazywamy liczbę rzedzywistą

$$\|A\| = \max_i \{ \lambda_i(AA^T) \mid i=1, 2, \dots, n \} \quad (4.1.22)$$

gdzie zapis $\lambda_i(AA^T)$ oznacza i -tą wartość własną macierzy AA^T .

Uzasadnienie użycia normy spektralnej, jako oszacowania stałej asymptotycznej błędu metody rozwiązywania układów równań, zawarte zostało w pracy [51], jak również poruszane będzie w rozdz. 4.3 przy okazji formułowania warunków zbieżności metody Crossa-Łobaczewa.

Powracając do zagadnienia szybkości zbieżności metody Crossa-Łobaczewa, zauważmy, że algorytm ten jest zbieżny liniowo, o stałej asymptotycznej błędu C nie większej niż norma spektralna macierzy (4.1.10). Efektywność rozpatrywanej metody, ze względu na liniową zbieżność, jest odwrotnie proporcjonalna do wymaganej dokładności obliczeń, jednak może zostać znacznie zwiększona przez zastosowanie prostych procedur przyspieszających, które omówione zostaną w rozdziale następnym.

4.2. Opis procedury redukcji i procedury przyspieszającej otrzymanie rozwiązania optymalnego

W rozdz. 3 dokonano transformacji zadania optymalizacji nieliniowej typu (1.3.4)-(1.3.7) do układu nieliniowych równań algebraicznych (3.3.15)-(3.3.17) a następnie przy dodatkowym warunku (3.3.34), wykorzystując specyficzną postać otrzymanego układu, przeprowadzono jego redukcję do postaci (3.3.36).

Sposób transformacji oraz warunki jej realizowalności przedstawiono w rozdziale 3.3, poniżej natomiast omówione zostaną: procedura redukcji oraz procedura przyspieszająca otrzymanie rozwiązania optymalnego, umożliwiającą wykorzystanie w procesie optymalizacji metody Crossa-Łobaczewa omawianej w rozdziale poprzednim.

PROCEDURA 2 (Redukcja)

- Krok 1. Podstaw $W := \text{card } W$; $N := \text{card } E$; $SK := 1$; $J := 1$; $I := 2$; $D(1) := 1$; $ZE := \text{crd}^2(P(\tau_1))$; $A(1,1) := 0$ i przejdź do kroku 2.
- Krok 2. Sprawdź, czy $A(J,1) \neq 0$? Jeżeli tak, przyjmij $K := A(J,1)$ i przejdź do kroku następnego. Gdy nie - przejdź do kroku 4.
- Krok 3. Sprawdź, czy $ZE = \text{crd}^{SK} P(\tau_K)$? Gdy nie, przejdź do kroku 4. W przypadku przeciwnym podstaw $D(I+1) := \text{crd}^{(3-SK)} [P(\tau_K)]$; $ZE := D(I+1)$; $A(J,1) := 0$; $D := I+2$; $J := J+2$; $KZ := 3$ i wróć do kroku 2.
- Krok 4. Sprawdź, czy $J < W-1$? Jeśli tak, $J := J+1$ i powrót do kroku 2. Gdy nie, przejdź do kroku następnego.
- Krok 5. Sprawdź, czy $SK = 1$? Jeżeli tak, podstaw $SK := 2$; $J := 1$ i wykonaj krok 2. W przypadku, gdy nie, sprawdź wartość przełącznika KZ . Gdy $KZ = 3$, wykonaj krok następny. Jeżeli $KZ = 1$, to podstaw $KY := KY+2$, $I := I-1$ i przejdź do kroku 6.
- Krok 6. Sprawdź, czy $I < KY$? Jeżeli nie, to podstaw $D(I) := 999$; $KZ, SK, J := 1$, $D(I+1) := D(I-KY)$, $ZE := D(I+1)$; powrót do kroku 2. W przypadku przeciwnym podstaw $I := I-2$ i wykonaj krok 7.
- Krok 7. Sprawdź, czy $D(I) = 999$? Jeśli tak, to podstaw $I := I-2$ i wykonaj ponownie krok 7. Gdy nie, $L := |D(I)|$, $Q(D(I-1)) := Q(D(I-1)) + Q(D(I+1))$; $X(L) := Q(D(I+1) \text{SGN}(D(I)))$ i przejdź do kroku następnego.

Krok 8. Sprawdź, czy $I \leq 2$? Jeżeli nie, to podstaw $I := I - 2$ i wykonaj ponownie krok 7. Gdy tak, przejdź do kroku 9.

Krok 9. Oblicz wartości współczynników $G_{jj}^{(0)}, G_{jj}^{(1)}$; $j=1, 2, \dots, p$, występujących w układzie równań (3.3.36) w oparciu o zależność

$$G_{jj}^{(0)} := \sum_{I=1}^N \bullet(I, J) \cdot R(I) |X(I)|^K \cdot \text{SGN}(X(I))$$

$j=1, 2, \dots, (p+e) \quad (4.2.1)$

$$G_{jj}^{(1)} := K \sum_{I=1}^N B(I, J) \cdot R(I) \cdot |X(I)|^{K-1}$$

i przejdź do kroku następnego.

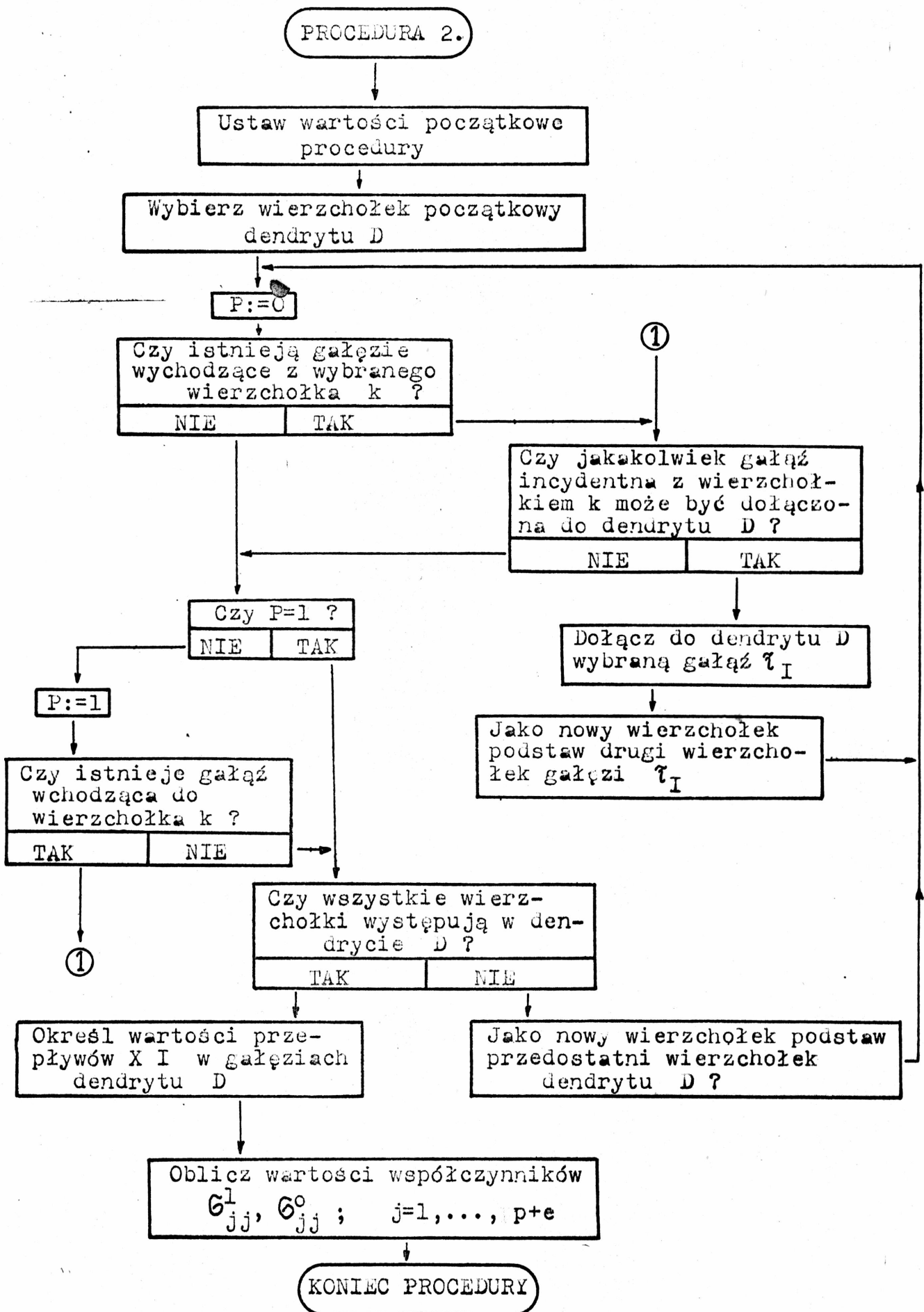
Krok 10. Koniec procedury redukcji.

W przedstawionej procedurze wyszczególnić można dwa etapy: generacji punktu startowego - x_0 (Kroki 1-8) oraz redukcji właściwej (Krok 9).

Generacja punktu x_0 polega na wyznaczeniu maszynowej reprezentacji dendrytu D , a następnie w oparciu o otrzymaną strukturę tego drzewa, określeniu takich wartości przepływów $X(I)$ w jego gałęziach, dla których spełniony jest układ (3.3.15). Po wykonaniu opisanych czynności syponujemy punktem startowym $(X(1), X(2), \dots, X(N))$ spełniającym powyższe wymaganie, przy czym różne od zera wartości przepływów $X(I)$ odpowiadają gałęziom dendrytu D . Wydajność źródła pierwszego określa natomiast całkowite zapotrzebowanie odbiorców.

Etap redukcji właściwej sprowadza się do określenia współczynników $G_{ij}^{(k)}$ występujących w układzie równań (3.3.36), jednakże z uwagi na przyjętą metodę rozwiązywania tego układu (rozdz.4.1) wyznaczane są jedynie te współczynniki $G_{ij}^{(k)}$, dla których $k \leq 1$ oraz $i=j$. Pozostałe natomiast w myśl założeń metody Crossa-Łobaczewa, porównywane są do zera. Z chwilą wykonania Kroku 9 opisanej procedury dysponujemy układem równań postaci (3.3.36) oraz punktem startowym X_0 umożliwiającym wyznaczenie rozwiązania w oparciu o opisaną w rozdziale poprzednim metodę Crossa-Łobaczewa.

Procedura powyższa, której schemat blokowy przedstawiono na Rys.4.1, odznacza się niskim stopniem złożoności oraz umożliwia zredukowanie ilości koniecznych do rozwiązania równań, przy czym



Rys.4.1. Schemat ogólny sieci działania procedury redukcji

stopień redukcji jest w przybliżeniu wprost proporcjonalny do mocy zbioru W. Należy przy tym zaznaczyć, że wybór metody Crossa-Łobaczewa tak znacznie ułatwiający redukcję i upraszczający dalszy proces obliczeń, pociąga za sobą wzrost ilości iteracji koniecznych do otrzymania wymaganej dokładności. Powodem tego jest wolna (liniowa) zbieżność metody Crossa-Łobaczewa, obniżająca efektywność obliczeń tym znacznie, im bliżej punktu optymalnego się znajdujemy. Wpływ tej negatywnej własności na efektywność obliczeń może zostać jednak ograniczony poprzez zastosowanie procedur przyspieszających [51].

Poniżej przedstawiony zostanie algorytm zbliżony do procesu δ^2 -Aitkena [51], który wykorzystując fakt liniowej zbieżności metody Crossa-Łobaczewa, zapewnia zwiększenie efektywności jej działania.

Z własności (4.1.20) metod zbieżnych liniowo wynika, że dla dostatecznie bliskiego otoczenia rozwiązania $\hat{\Delta}$, wartości

$$(\Delta^k - \hat{\Delta}); (\Delta^{k+1} - \hat{\Delta}); (\Delta^{k+2} - \hat{\Delta}); \dots; (\Delta^n - \hat{\Delta}) \quad (4.2.2)$$

tworzą ciąg geometryczny o ilorazie C, określonym zależnością (4.1.21). Oczywiście wielkości $\delta^k; \delta^{k+1}; \dots; \delta^n$, gdzie zgodnie z (4.1.4)

$$\delta^k = \Delta^k - \Delta^{k-1} = (\Delta^k - \hat{\Delta}) - (\Delta^{k-1} - \hat{\Delta}) \quad (4.2.3)$$

tworzą także ciąg geometryczny, o tej samej wartości ilorazu C i postaci

$$\{\delta^n\} = \{[(\Delta^k - \hat{\Delta})(C-1)] C^{n-2}; n \in \mathbb{N}_2^{\infty}\} \quad (4.2.4)$$

W szczególności więc mamy

$$\|\delta^{k+1}\| = C \|\delta^k\| = \sqrt{(C \cdot \delta_1^k)^2 + (C \cdot \delta_2^k)^2 + \dots + (C \cdot \delta_p^k)^2} \quad (4.2.5)$$

$$\text{a stąd } (\exists k \in \mathbb{N})(\forall i = 1, 2, \dots, p)(\forall n \in \mathbb{N}_k)(\delta_i^n = C \cdot \delta_i^{n-1}) \quad (4.2.6)$$

więc składowe wektorów $\delta^k, \delta^{k+1}, \dots, \delta^n$ tworzą również ciąg geometryczny o ilorazie C.

Jednocześnie, zgodnie z (4.1.3) mamy

$$\hat{\Delta}_i = \Delta_i^0 + \lim_{n \rightarrow \infty} \sum_{k=1}^n \delta_i^k, \quad \text{dla } i=1, 2, \dots, p \quad (4.2.7)$$

a stąd, korzystając ze wzoru na sumę wyrazów ciągu geometrycznego

$$\hat{\Delta}_i = \Delta_i^0 + \lim_{n \rightarrow \infty} \left[\frac{1-C^n}{1-C} \delta_i^1 \right] = \Delta_i^0 + \frac{(\delta_i^1)^2}{\delta_i^1 - \delta_i^2} \quad i=1,2,\dots,p \quad (4.2.8)$$

Otrzymana zależność (4.2.8) znacznie polepsza zbieżność metody Crossa-Łobaczewa pod warunkiem, że prawdziwe są poczynione powyżej założenia. Dlatego procedura, która zostanie teraz przedstawiona, uruchamiana jest po uprzednim stwierdzeniu, że zależności (4.1.20), (4.2.6) są spełnione.

Zauważmy jednak, że otrzymana ze wzoru (4.1.14) zależność

$$\xi^{(r)} = [\Phi'(\hat{\Delta})]^r \cdot \xi^{(0)} ; r \in \mathbb{N} \quad (4.2.9)$$

określający r-ty wektor iterowany, stanowi podstawę znanej z literatury [51] metody potęgowej obliczania wartości własnej macierzy $\Phi'(\hat{\Delta})$ o największym module.

Stąd na podstawie twierdzenia Von Misesa [51] mamy

$$(\exists k \in \mathbb{N})(\forall r \geq k)(\xi^{(r+1)} = \Phi'(\hat{\Delta}) \xi^{(r)} = \lambda_c \cdot \xi^{(r)}) \quad (4.2.10)$$

gdzie λ_c - wartość własna macierzy $\Phi'(\hat{\Delta})$ o największym module co przy podstawieniu $\lambda_c := C$ daje (4.2.6).

PROCEDURA 3. (Akceleracja)

Krok 1. Podstaw $I:=1$; $K:=1$; $C,D,E:=0$; $J:=0$ i przejdź do kroku 2.

Krok 2. Oblicz wartości wyrażeń;

$$A := \frac{DE(4,I)}{DE(3,I)} \quad B := \frac{DE(2,I)}{DE(1,I)} \quad C := C + DE(1,I)^2$$

$$D := D + DE(2,I)^2 \quad E := E + DE(3,I)^2$$

Sprawdź, czy $|A-B| < 10^{-2}$? Jeżeli tak, $J:=J+1$ i przejdź do kroku 3. W przypadku, gdy nie - przejdź do kroku następnego.

Krok 3. Sprawdź, czy $I \geq P$? W przypadku, gdy tak - przejdź do kroku 4. Jeżeli nie - podstaw $I:=I+1$ i powrót do kroku 2.

Krok 4. Podstaw $C:=\sqrt{C}$; $D:=\sqrt{D}$; $E:=\sqrt{E}$ wyznacz wartości $D:=D/E$ $C:=C/D$ oraz sprawdź czy $|C-D| < 10^{-2}$? jeżeli nie, to przejdź do kroku 7. W przypadku, gdy tak - sprawdź wartość $(J-P)$. Jeżeli $J-P < 0$, to przejdź do kroku 7. Gdy $J-P=0$, podstaw $I:=0$ i wykonaj krok następny.

Krok 5. Podstaw $D:=I+1$, $L:=1$ oraz oblicz:

$$\delta_I := \frac{DE(4, I)^2}{DE(3, I) - DE(4, I)}$$

i przejdź do kroku następnego.

Krok 6. Podstaw $J:=|B(I, L)|$ i oblicz nowe wartości przybliżenia rozwiązania wg wzoru

$$X(J)^{k+1} := X(J)^k + \delta_I \cdot \text{SGN}(B(I, L))$$

Sprawdź, czy $L < SP(I)$? Jeżeli tak, $L:=L+1$ i powtórz krok 6. Gdy nie, sprawdź czy $I \leq N-W+1$? Gdy tak, wróć do kroku 5. W przypadku gdy nie, wykonaj krok 7.

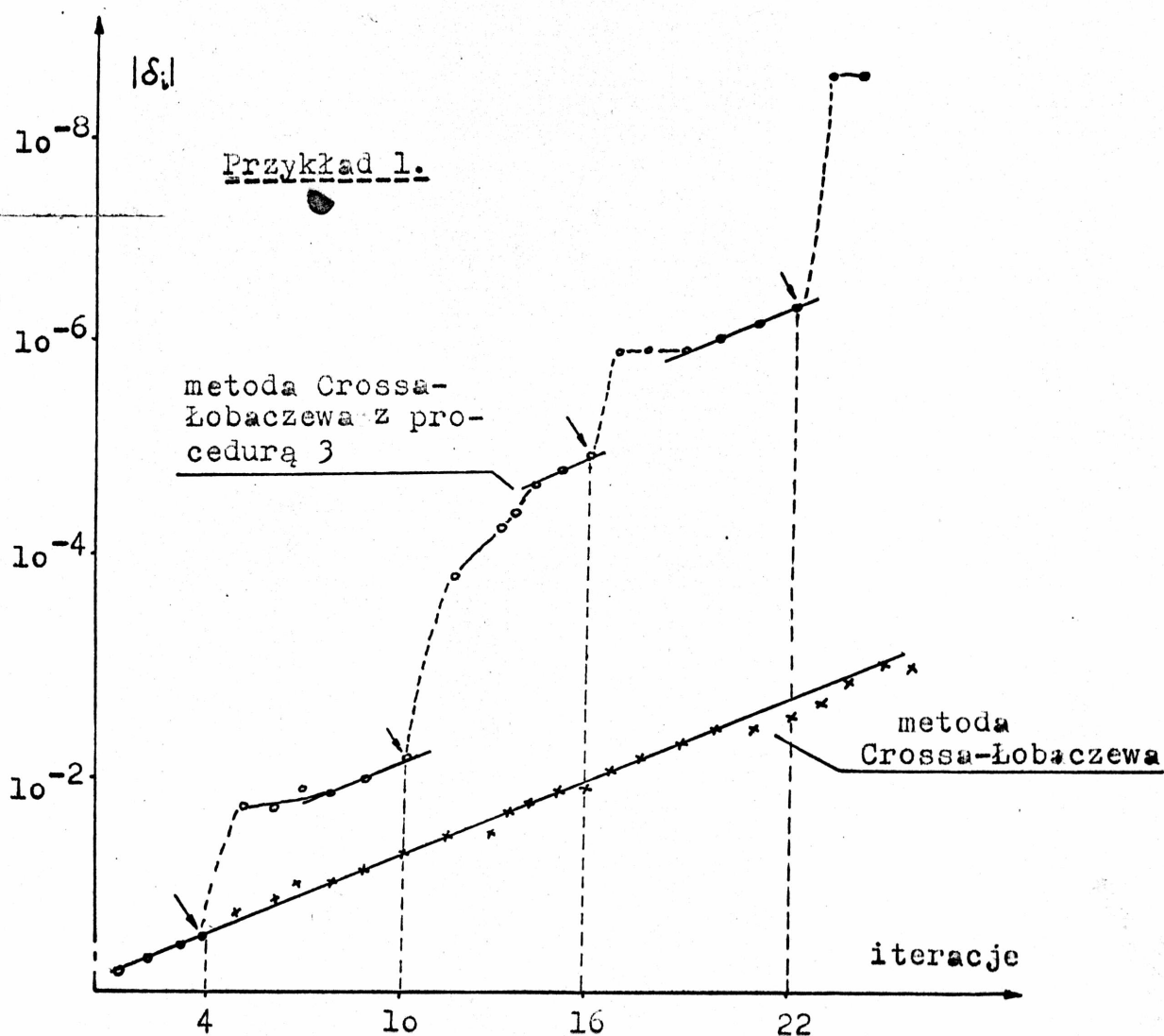
Krok 7. Koniec procedury przyspieszającej.

Celem pełniejszego opisu zasad działania powyższej procedury, na rys.4.2 przedstawiono w skali logarytmicznej wielkości jednej ze składowych wektora δ , wyznaczonej w kolejnych iteracjach w oparciu o metodę Crossa-Łobaczewa uzupełnioną procedurą 3 i bez niej, w czasie rozwiązywania Przykładu 1.

Strzałkami zaznaczono moment każdorazowego uruchomienia procedury, której działanie rozpoczyna się od sprawdzenia, czy spełnione są warunki jej stosowalności (kroki 1-4). W przypadku, gdy cztery ostatnie punkty wyznaczają, z przyjętą dokładnością, prostą równoległą do prostej określonej przez wyniki metody oryginalnej, wyznaczone są poprawki przyspieszające δ_I (krok 5). Następnie obliczane są nowe wartości przybliżeń (Krok 6) i następuje powrót do oryginalnej metody Crossa-Łobaczewa, aż do momentu kolejnego wywołania procedury przyspieszającej.

Dokładna analiza wyników badań porównawczych przeprowadzona zostanie w rozdz. 4.5, tutaj podkreślimy jedynie:

- dużą efektywność procedury przyspieszającej,
- empiryczne potwierdzenie liniowej zbieżności metody Crossa-Łobaczewa,
- konieczność rozwiązania problemu częstości i wyboru chwili początkowej uruchamiania procedury przyspieszającej.



Rys.4.2. Wielkości poprawek δ_i obliczanych w kolejnych iteracjach w oparciu o oryginalną i zmodyfikowaną przez autora metodę Crossa-Łobaczewa

4.3. Algorytmy optymalizacji nieliniowych sieci przepływowych z wykorzystaniem postaci zredukowanej problemu wyjściowego

Poniżej przedstawione zostaną dwa algorytmy przeznaczone do rozwiązywania zagadnień optymalizacji (1.3.4)-(1.3.7).

Pierwszy (I), którego koncepcja budowy oparta jest na zasadach transformacji i redukcji problemu wyjściowego, przedstawionych w rozdz. 3.3 oraz rozdz. 4.2, zawiera między innymi procedurę roz-

wiązywania układów równań nieliniowych zmodyfikowaną metodą Crossa-Łobaczewa.

Drugi (II) - posiadający strukturę związaną bezpośrednio z Twierdzeniem 3.2 i Wnioskiem 3.3, oparty jest na metodzie Crossa [10], przy czym różnice pomiędzy wspomnianymi metodami sprowadzają się do innych postaci funkcji φ_j występujących w zależności (4.1.1).

Dwoistość ta podyktowana została względami praktycznymi. Jak wykazane zostanie w rozdz. 4.5, użycie algorytmu I daje szczególnie dobre rezultaty w przypadku, gdy nie dysponujemy modelem (1.2.21)-(1.2.23) danej sieci i konieczna jest jej symulacja. Natomiast użycie algorytmu II zalecane jest w procesach automatycznego sterowania sieciami przepływowymi, gdzie pomiar wartości potencjałów węzłowych π_k umożliwia wyznaczenie minimalno-energetycznej strategii sterowania systemem.

Dążąc do zwiększenia przejrzystości algorytmów wyodrębniono w nich grupę pięciu procedur (zasady trzech z nich; transformacji, redukcji i procedury przyspiszającej, opisane zostały w rozdz. 3.3, rozdz. 4.2). Stąd przed przystąpieniem do opisu algorytmów omówione będą kolejno; procedura budowy węzłowo-gałęziowej macierzy incydencji A oraz procedura określania macierzy oczkowej B.

Ogólnie znaną własnością macierzy incydencji jest fakt, że każda jej kolumna zawiera jedynie dwa elementy różne od zera. Wskaźnik zapełniania macierzy definiowany jako iloraz ilości elementów niezerowych do elementów wszystkich, będzie więc równy:

$$\eta_A = \frac{2n}{n \cdot w} = 2/w \quad (4.3.1)$$

Ponadto wykonywanie działań arytmetycznych na elementach macierzy A o tak niskim wskaźniku zapełnienia, charakteryzuje się małą efektywnością, ponieważ konieczność wielokrotnego mnożenia i dodawania wartości zerowych lub wcześniejszego sprawdzania elementów a_{ij} , znacznie zwiększa czas trwania obliczeń. W konsekwencji tradycyjną postać macierzy incydencji należy uznać za tym mniej przydatną do obliczeń numerycznych, im większą sieć ona opisuje.

Z tych powodów, procedura wyznaczania macierzy incydencji, która zostanie podana poniżej, wykorzystuje jeden ze sposobów zapisu macierzy rzadkich, zapewniający wielkość wskaźnika zapełnienia (4.3.1) na poziomie nie niższym niż

$$\eta_A = \frac{2n}{2n+w} = \frac{2}{2 + \frac{w}{n}} \geq 0,5 \quad (4.3.2)$$

PROCEDURA 4. (Wyznaczanie macierzy A)

- Krok 1. Wprowadź parametry $W := \text{card } W$, $N := \text{card } E$ oraz przyjmij $J := 1$; $SK := 1$; $I := 1$; $K := 1$. Przejdź do kroku 2.
- Krok 2. Sprawdź, czy $\text{crd}^{SK} P(\tilde{\gamma}_K) = D$? Jeżeli tak, wykonaj krok następny. W przypadku przeciwnym, przejdź do kroku 4.
- Krok 3. Określ element $A(J, K)$ podstawiając;
 $A(J, K) := (-1)^{SK} \cdot I$
 oraz zwiększ wskaźnik rzędu wierzchołka J-tego tj,
 podstaw $K := K + 1$ i przejdź do kroku 4.
- Krok 4. Sprawdź, czy $I < N$? Jeśli tak, wykonaj krok następny, w przeciwnym razie przejdź do kroku 6.
- Krok 5. Podstaw $I := I + 1$ i powrót do kroku 2.
- Krok 6. Sprawdź, czy $SK = 1$? W przypadku gdy tak, podstaw $SK := SK + 1$, $I := 1$ i powrót do kroku 2. Jeżeli nie, przejdź do kroku następnego.
- Krok 7. Podstaw $S(J) := K - 1$. Sprawdź, czy $J < W$? Jeżeli tak, podstaw $J := J + 1$, $I := 1$, $K := 1$ i ponownie wykonaj krok 2. Gdy nie, wykonaj krok następny.
- Krok 8. Koniec procedury określania macierzy A.

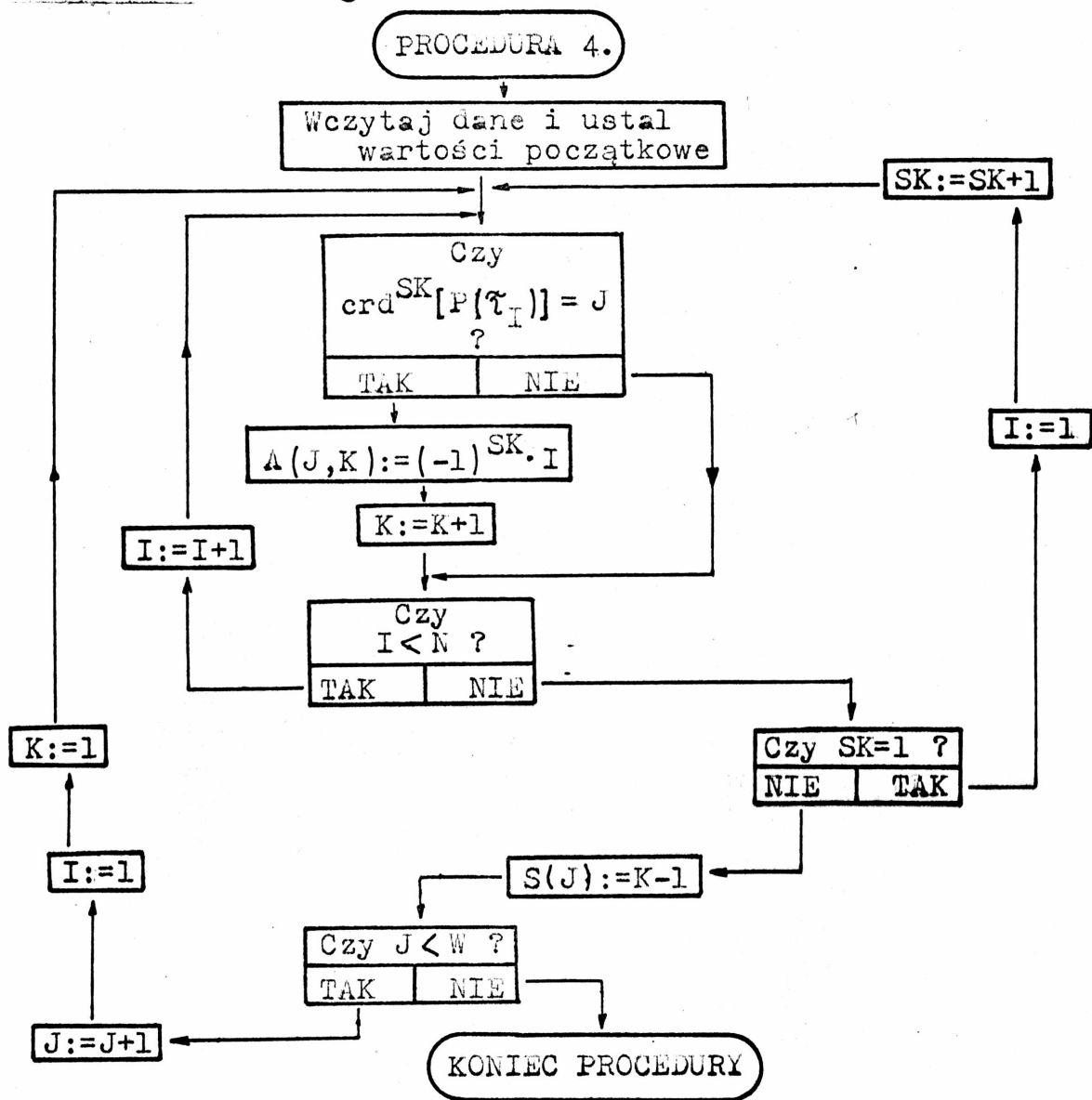
Powyższa procedura umożliwia wyznaczenie macierzy incydencji A, przy czym pierwsze $W = \text{card } W$ wierszy macierzy A odpowiada kolejnym wierzchołkom grafu (1.2.13). W każdym wierszu - J znajduje się $S(J)$ elementów niezerowych, określających numery łuków incydentnych z danym węzłem. Jeżeli łuk wchodzi do węzła, wtedy odpowiedni element macierzy jest dodatni, w przypadku przeciwnym - jest ujemny.

Dysponując macierzą A, otrzymaną w wyniku realizacji powyższej procedury, możemy przystąpić do budowy macierzy obwodów podstawowych B. Oczywiście również i w tym przypadku wybrano taki sposób komputerowej reprezentacji macierzy B, który zapewnia dużą efektywność wykonywanych obliczeń tzn. pamiętane są jedynie elementy niezerowe macierzy B.

W teorii grafów [11],[20] znane są dwie podstawowe własności umożliwiające określenie macierzy obwodów podstawowych. Jedną z nich jest ortogonalność macierzy A, B , drugą natomiast określa zależność

$$B \cdot C^T = 0 \quad (4.3.3)$$

gdzie: C - macierz przekrojów podstawowych wymiaru $[(w-1) \times n]$.



Rys.4.3. Schemat blokowy procedury wyznaczania macierzy A

Jeżeli przyjąć

$$B = [1_B | B_1] \quad \text{oraz} \quad C = [C_1 | 1_C] \quad (4.3.4)$$

gdzie:

1_B - macierz jednostkowa odpowiadająca cięciwom dendrytu, względem którego budowana jest macierz B ,

B_1 - odpowiednia podmacierz dla gałęzi dendrytu,

C_1 - macierz przekrojów złożona z kolumn odpowiadających cięciwom

1_C - macierz jednostkowa dla $(w-1)$ gałęzi dendrytu,

wtedy z (4.33) otrzymamy

$$1_B \cdot C_1^T + B_1 \cdot 1_C = 0 \quad (4.3.5)$$

skąd

$$B_1 = -C_1^T \quad (4.3.6)$$

Warunek powyższy stanowi podstawę działania opracowanej procedury budowy macierzy B. Wybór taki wynika z faktu, że określenie macierzy B w oparciu o zależność (3.2.19) wymaga działań na macierzach odwrotnych, co zgodnie ze wzorem Strassena [63] pociąga za sobą konieczność wykonania w^{lg}2⁷ operacji mnożenia i dodawania. Zastosowanie zaś procedury opisanej poniżej, sprowadza się do wykonania co najwyżej tej samej ilości operacji dodawania, w rezultacie skracając czas obliczeń.

PROCEDURA 5. (Wyznaczanie macierzy B)

Krok 1. Przyjmij $W := \text{card } W$; $N := \text{card } E$; $J := 1$; $I := 1$; $K := 1$ i przejdź do kroku następnego.

Krok 2. Sprawdź, czy $I=J$? Jeśli tak, podstaw $I := I+1$ i przejdź do kroku następnego. W przypadku przeciwnym wykonaj krok 3.

Krok 3. Oblicz wartość wyrażenia $Z := |A(I,1) - A(I,K)|$. Jeżeli $Z=0$, wykonaj krok 7. Jeśli jest inaczej, przejdź do kroku następnego.

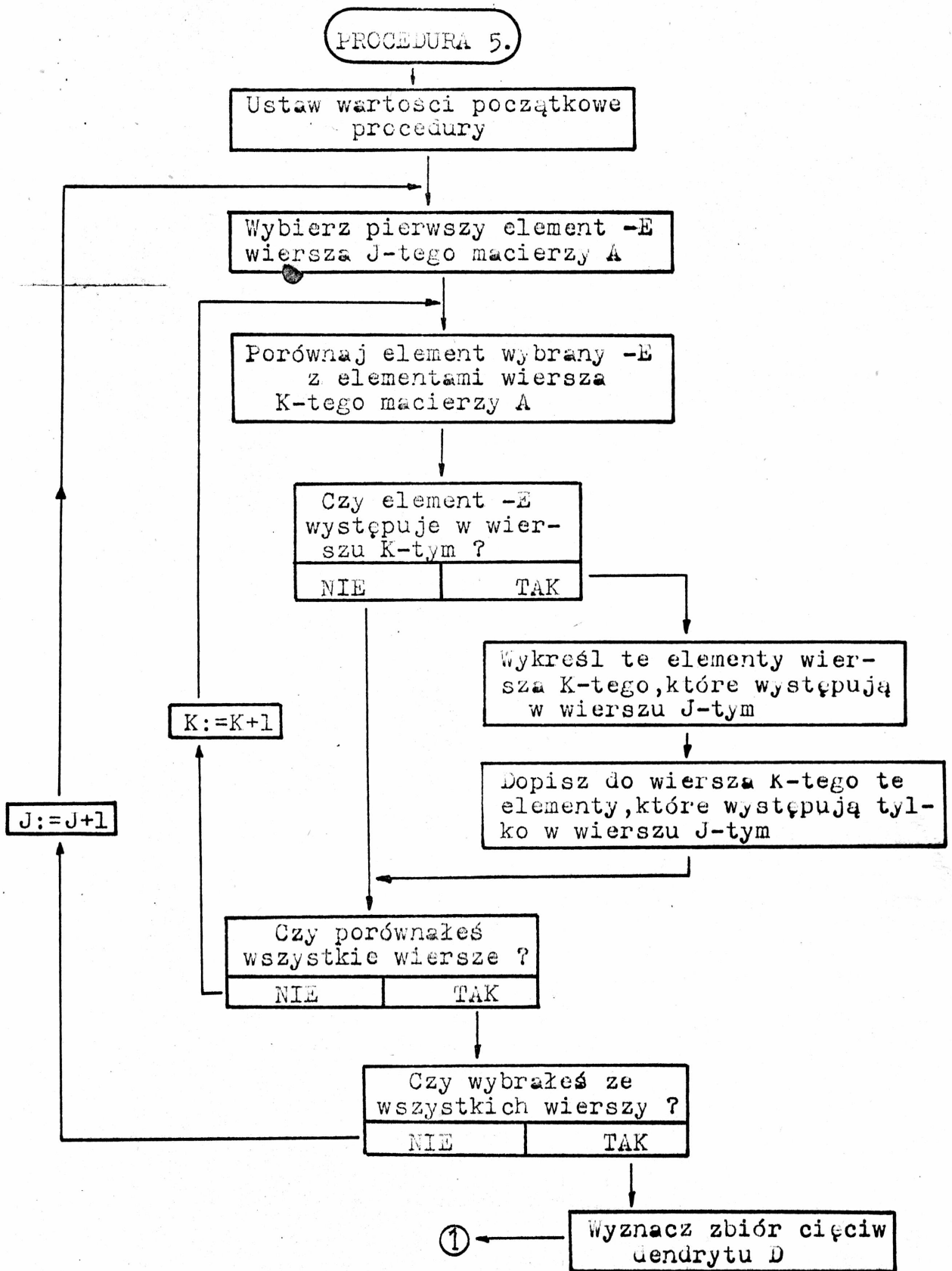
Krok 4. Sprawdź, czy $K < S(J)$? Gdy tak, podstaw $K := K+1$ i powtórz krok 3. W przypadku, gdy nie, wykonaj krok następny.

Krok 5. Sprawdź, czy $J < W$? Jeżeli tak, podstaw $J := J+1$, $K := 1$ i wróć do kroku 2. Gdy nie, wykonaj krok 6.

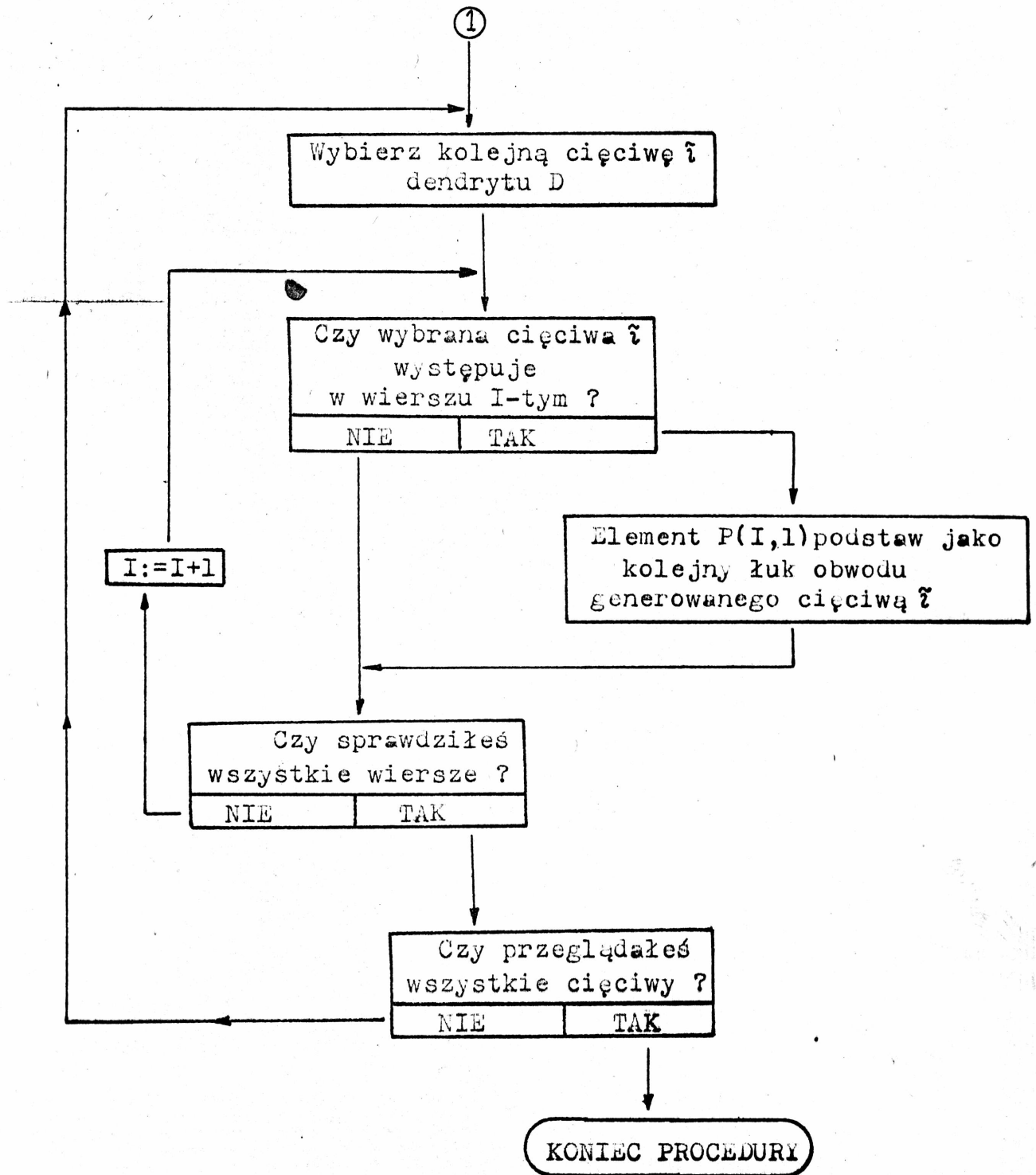
Krok 6. Porównaj wielkości: I oraz W . Gdy $I < W-1$, podstaw $I := I+1$, $K := 1$, $J := 1$ i cofnij się do kroku 2. W przypadku, gdy $I = W-1$, przejdź do kroku 9.

Krok 7. Porównaj każdy z elementów wiersza I -tego z każdym elementem wiersza J -tego macierzy A , przy czym jeżeli

$$(\exists L \in S(I)) (\exists N \in S(J)) (|A(I,L)| = |A(J,N)|)$$



Rys.4.4. Ogólna sieć działania procedury wyznaczania macierzy B (część 1)



Rys.4.4. Ogólna sieć działania procedury wyznaczania macierzy B (część 2)

wtedy podstaw $A(J,N):=0$ i przejdź do kroku 8. Natomiast gdy,

$$(\exists L \in S(I))(\forall N \in S(J))(|A(I,L)| \neq |A(J,N)|)$$

to przyjmij $S(J):=S(J)+1$ oraz $A(J,S(J)):=A(I,L)$ i przejdź do kroku 8.

Krok 8. Sprawdź, czy istnieje taka wartość $L=1,2,\dots,S(J)$, dla której $A(J,L)=0$? Jeżeli tak, to dla każdego $N=L,L+1,\dots,S(J)$ podstaw $A(J,N):=A(J,N+1)$ oraz $S(J):=S(J)-1$ i powtórz krok 8. Jeżeli nie, powrót do kroku 5.

Krok 9. Wyznacz zbiór cięciw dendrytu $D = \{\tilde{\tau}_I \mid I=1,2,\dots,N-W+1\}$, dla którego $(\sim \exists I \in \{1,\dots,N\})(|A(J,1)|=I)$ oraz przyjmij $J:=1$, $I:=1$, $SP(J):=1$ i przejdź do kroku następnego.

Krok 10. Podstaw $B(J,SP(J)):=\tilde{\tau}_I$ i przyjmując $SP(J):=SP(J)+1$, dołącz do wiersza J -tego macierzy B , te elementy $A(L,1)$; $L=1,2,\dots,(W-1)$, dla których

$$(\exists L,N \in \mathbb{N})(|A(L,N)|=\tilde{\tau}_I)$$

Krok 11. Sprawdź, czy $I < N-W+1$? Jeżeli tak, to podstaw $J:=J+1$, $I:=I+1$, $SP(J):=1$ i wróć do kroku 10. W przypadku przeciwnym przejdź do kroku następnego.

Krok 12. Koniec procedury wyznaczania macierzy B .

Postępowanie opisane w Procedurze 4 kończy się z chwilą otrzymania macierzy B , zbudowanej z $(n-w+1)$ wierszy określających obwo-
dy podstawowe, z których każdy zawiera $SP(J)$ (J -wskaźnik wiersza) elementów niezerowych. Pierwszym elementem każdego wiersza jest numer cięciwy, następnymi - numery gałęzi dendrytu tworzących dany obwód. Jeżeli element J -tego wiersza jest ujemny, oznacza to, że odpowiednia gałąź grafu wchodzi do J -tego obwodu ze zwrotem przeciwnym do jego orientacji, która pokrywa się z orientacją cięciwy generującej ten obwód. W przypadku przeciwnym zwroty gałęzi i obwodu są zgodne.

Zostanie teraz przedstawiony algorytm I rozwiązywania zagadnienia optymalizacji (1.3.4)-(1.3.7) oparty na zasadach transformacji i redukcji opisanych w poprzednich rozdziałach. W algorytmie tym wykorzystano zmodyfikowaną przez autora metodę Crossa-Łobaczewa, co umożliwiło uzyskanie lepszej efektywności obliczeń.

Wprowadzone zmiany dotyczą dwóch zagadnień. Po pierwsze - zredukowano czas trwania iteracji poprzez obliczanie jedynie tych wartości $\delta_j^{(r)}$, które posiadają największy wpływ na otrzymaną dokładność rozwiązania. Po drugie - korzystając z liniowej zbieżności metody Crossa-Łobaczewa zastosowano proces przyspieszający jej zbieżność.

ALGORYTM I.

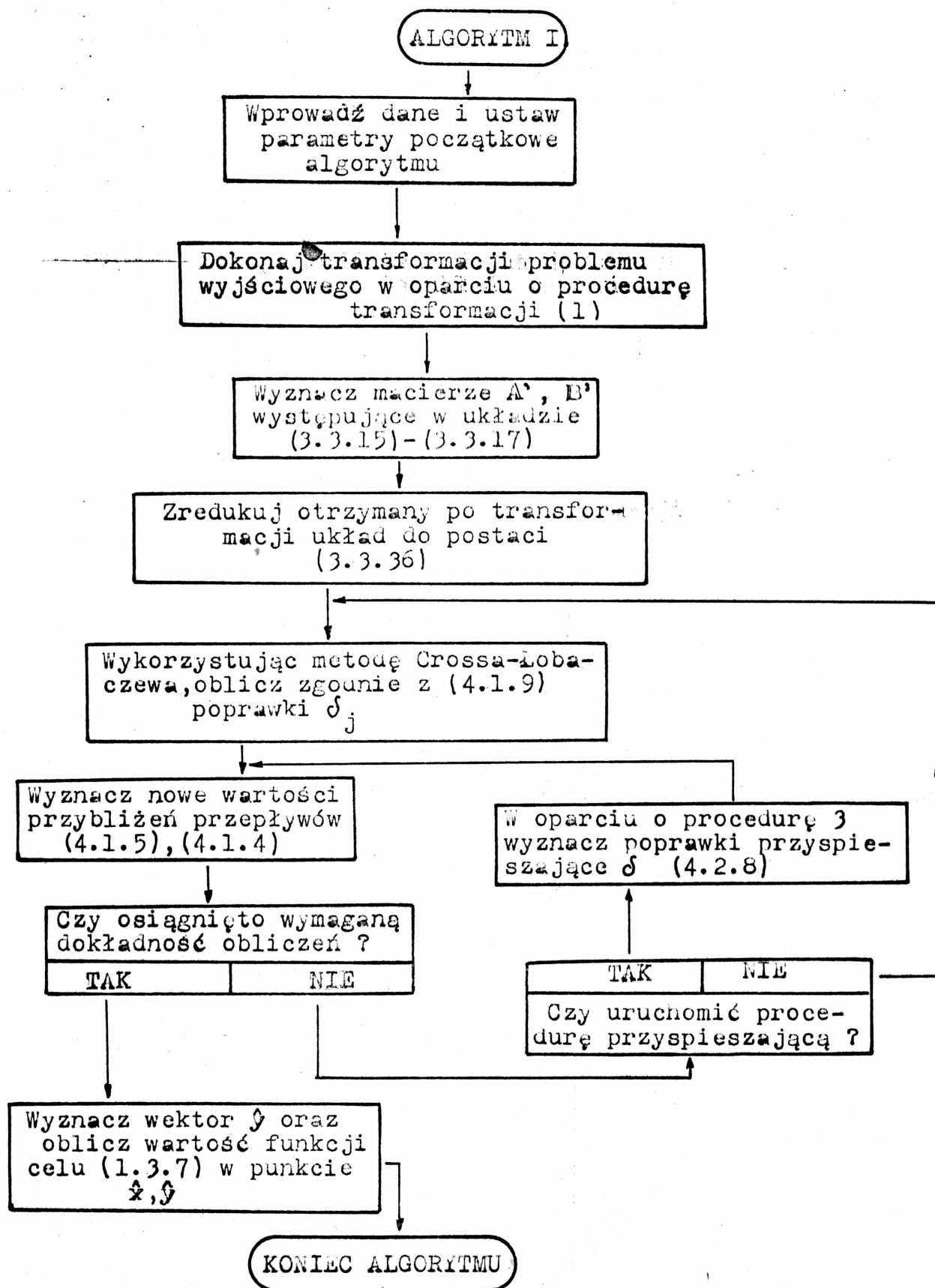
- Krok 1. Wprowadź parametry $N:=\text{card } E$; $W:=\text{card } W$, $LI:=$ liczba iteracji, $\epsilon:=$ wymagana dokładność rozwiązania, $L1:=$ numer iteracji, od której zacznie działać procedura przyspieszająca, $L2:=$ ilość iteracji pomiędzy kolejnymi włączeniami procedury 3, oraz podstaw $J:=0$, $PK:=1$, $R:=1$, $I:=1$ i przejdź do kroku 2.
- Krok 2. W oparciu o zasady transformacji (3.3.20)-(3.3.22) rozszerz strukturę przestrzenną rozpatrywanej sieci i przejdź do kroku 3.
- Krok 3. Wyznacz według procedury 4, węzłowo-gałęziową macierz incydencji A' i przejdź do kroku następnego.
- Krok 4. Zbuduj macierz obwodów podstawowych B' zgodnie z procedurą 5 i przejdź do kroku 5.
- Krok 5. Korzystając z procedury 2, przeprowadź redukcję układu równań opisanego macierzami A', B' , wyznacz punkt startowy obliczeń i przejdź do kroku 6.
- Krok 6. Podstaw $J:=J+1$; $L:=1$; $\varphi_J^R:=0$; $g_J^R:=0$; $F:=-1$ i przejdź do kroku następnego.
- Krok 7. Sprawdź, czy $M_J=0$? Jeżeli tak, przejdź do kroku następnego. W przypadku, gdy nie, podstaw $M_J:=0$; $G:=G+G/J$ i przejdź do kroku 11.
- Krok 8. Przyjmij $I:=|B'(J,L)|$ oraz oblicz wartości wyrażeń

$$g_J^R = g_J^R + d_I \cdot |X(I)^{(R)}| \cdot K \cdot \text{SGN}(X(I)^{(R)}) \cdot \text{SGN}(B'(J,L))$$

$$\varphi_J^R = \varphi_J^R + d_I \cdot |X(I)^{(R)}|^{k-1}$$
 Sprawdź, czy $L < SP(J)$? Jeśli tak, podstaw $L:=L+1$ i wróć do kroku 8. W przypadku gdy nie, wykonaj krok następny.

- Krok 9. Przyjmij $L:=1$; $\delta_J^{(R+1)} := \frac{-\varepsilon_J^R}{k \cdot \varphi_J^R}$; $G:=G+\delta_J^{(R+1)}$;
 $DE(T,J):=\delta_J^{(R+1)}$ i sprawdź czy $\delta_J^{(R+1)} < \frac{G-\delta_J^{(R+1)}}{J}$?
 Jeżeli tak, podstaw $M_J:=777$ i wykonaj krok następny.
 Jeżeli nie, przejdź do kroku 10.
- Krok 10. Podstaw $I:=|B'(J,L)|$ i oblicz nowe wartości przybliżeń
 $X(I)^{(R+1)}:=X(I)^R + \delta_J^{(R+1)} \cdot \text{SGN}(B'(J,L))$ oraz sprawdź,
 czy $L < SP(J)$? Jeżeli tak, $L:=L+1$ i powtórz krok 10.
 Gdy nie, sprawdź czy $F=1$? Jeśli nie - wykonaj krok
 następny. Natomiast jeżeli $F=1$, przejdź do kroku 12.
- Krok 11. Sprawdź, czy $J < N-W+1$? Jeżeli tak, wróć do kroku 6.
 W przypadku gdy nie, sprawdź czy $R=PK$? Gdy tak, pod-
 staw $F:=1$, $J:=0$ i przejdź do kroku 13. Jeżeli nie
 ($PK \neq R$), wtedy wykonaj krok 14.
- Krok 12. Sprawdź, czy $J < N-W+1$? Jeśli tak, przejdź do kroku
 następnego. Gdy nie $PK:=PK+L2$ i wykonaj krok 14.
- Krok 13. Przyjmij $J:=J+1$; $\delta_J = \frac{DE(T,J)^2}{DE(T-1,J)-DE(T,J)}$; $L:=1$
 i powrót do kroku 10.
- Krok 14. Sprawdź, czy $G < \varepsilon$? Jeżeli tak to przejdź do kroku 16.
 Gdy nie - wykonaj krok następny.
- Krok 15. Sprawdź, czy $R < II$? Gdy nie, wykonaj krok następny.
 Jeżeli tak, to podstaw $R:=R+1$; $G:=0$; $J:=0$ i sprawdź
 czy $T=4$? Jeśli tak, to $T:=1$, w innym przypadku $T:=T+1$
 i powrót do kroku 6.
- Krok 16. Przyjmij $\hat{x} = x^R$ i w oparciu o zależność (3.3.17) wyz-
 nacz wartość $\hat{y}$ oraz oblicz wartość funkcji celu
 (1.3.7) w punkcie $(\hat{x}, \hat{y})$. W oparciu o procedurę 2
 (wygenerowany dendryt D), korzystając z zależności
 (1.2.25) wyznacz wartości potencjałów wierzchołków π_k .
 Wyprowadź wyniki obliczeń i przejdź do kroku 17.
- Krok 17. Koniec algorytmu I.

Postępowanie opisane przez algorytm I polega na transforma-
 cji problemu wyjściowego (1.3.4)-(1.3.7), a następnie w oparciu
 o procedurę 4 oraz 5, określeniu układu równań (3.3.15)-(3.3.17)
 modelującego rozpatrywaną sieć (1.2.17), (1.2.21)-(1.2.25).



Rys.4.5. Ogólny schemat działania algorytmu I

Otrzymany układ, po przeprowadzeniu jego redukcji do postaci (3.3.36), rozwiązywany jest iteracyjną metodą Crossa-Łobaczewa. Kolejne wartości przybliżeń rozwiązania - x^R obliczane są w krokach 8-10, gdzie sprawdzane są również wartości parametrów kontrolnych M_J, PK .

Wartości parametrów M_J , określając wpływ J-tych poprawek na dokładność wyniku, umożliwiają eliminację obliczeń tych wartości σ_J^R , których wpływ na dokładność rozwiązania jest pomijalnie mały. Natomiast wartości parametru kontrolnego PK decydują o uruchomieniu procedury 3, przyspieszającej zbieżność zastosowanej do obliczeń metody Crossa-Łobaczewa.

Przedstawiony zostanie teraz jeszcze jeden algorytm, który wykazując znaczne podobieństwo w stosunku do algorytmu I, może być wykorzystany w procesie automatycznego sterowania nieliniową siecią przepływową (por. rozdz. 6.2).

Jak nietrudno zauważyć na układ (3.3.36) rozwiązywany w algorytmie I składa się $(p+z-1)$ równań, z których p związanych jest z modelem rozpatrywanej sieci, natomiast $(z-1)$ wynika z konieczności spełnienia warunku (3.2.36).

Z uwagi na fakt, że w praktyce $(z-1) \ll p$, naturalnym jest dążenie aby w sytuacji, gdy posiadamy możliwość pomiaru stanu sieci, bądź dysponujemy jej modelem (zapewniającym spełnienie wszystkich " p " równań), ograniczyć się jedynie do zagadnienia rozwiązania pozostałych $(z-1)$ równań układu (3.3.36).

Idea prezentowanego algorytmu II opiera się na spostrzeżeniu, że spełnienie zależności (1.3.4)-(1.3.6) wymuszone zostaje przez samą sieć przepływową S , natomiast proces optymalizacji powinien zostać ograniczony do takich działań, które zapewniłyby spełnienie warunku (3.2.36).

W omawianym algorytmie II, w części symulującej sieć przepływową S wykorzystano metodę Crossa [10] rozwiązywania układów równań nieliniowych typu (1.2.21)-(1.2.23). Charakteryzuje się ona gorszą efektywnością obliczeń niż metoda Crossa-Łobaczewa, znana z algorytmu I, jednakże umożliwia bezpośrednie wykorzystanie warunków koniecznych (3.2.36) optymalności rozwiązania. Metoda ta polega bowiem na iteracyjnym obliczaniu kolejnych przybliżeń wartości potencjałów węzłowych π_w , za pomocą jednopunktowych wzorów iteracyjnych postaci;

$$\pi^{(R+1)} = \pi^{(R)} - \varphi^{(R)}(\pi^{(R)}) g^{(R)}(\pi^{(R)}) \quad (4.3.6)$$

tak długo, aż otrzymane wartości $\pi_w^{(m)}$ ($w=1,2,\dots,W$) nie spełnią (po uwzględnieniu zależności (1.2.25)) układu równań (1.3.4)-(1.3.6) modelującego sieć nieliniową S.

Ponieważ, zgodnie z Twierdzeniem 3.2 oraz Wn. 3.3 i Wn.3.4, otrzymane rozwiązanie $\hat{\pi} = [\hat{\pi}_1, \hat{\pi}_2, \dots, \hat{\pi}_W]^T$ powinno spełniać warunek

$$(\forall w_i, w_j \in W_Z) (\hat{\pi}_{w_i} - \hat{\pi}_{w_j} = \frac{\xi(w_j) - \xi(w_i)}{k+1} = \text{const}) \quad (4.3.7)$$

stąd wzór iteracyjny (4.3.6), w myśl metody Crossa [10], przyjmie postać

$$\pi_w^{(R+1)} = \frac{\xi(w)}{k+1} \quad \text{dla } w=1,2,\dots,z$$

$$\pi_w^{(R+1)} = \pi_w^{(R)} - \frac{\sum_{i=1}^{n+e} a_{wi} \cdot N_i \cdot \left[|\pi_w^R - \pi_i^R|^{1/k} \cdot \text{SGN}(\pi_w^R - \pi_i^R) \right] + q_w}{1/k \cdot \sum_{i=1}^{n+e} a_{wi} \cdot \frac{N_i}{|\pi_w^R - \pi_i^R|^{1/k}}} \quad (4.3.8)$$

$$w=(z+1), \dots, W$$

gdzie; N_i - przewodność łuku i-tego, określona zależnością

$$N_i = 1 / \sqrt[k]{d_i} \quad i=1,2,\dots,(n+e) \quad (4.3.9)$$

natomiast licznik ułamka występującego w wyrażeniu (4.2.13) odpowiada składowej $g_i^{(R)}(\pi^{(R)})$, zaś mianownik - $\varphi_i^{(R)}(\pi^{(R)})$.

ALGORYTM II.

Krok 1. Wprowadź dane wejściowe $W:=\text{card } W$, $LI:=$ liczba iteracji, $\varepsilon:=$ wymagana dokładność rozwiązania dla metody Crossa, $\varepsilon_J:=$ wymagana dokładność rozwiązania końcowego, $\pi_{\min}:=$ wymagana minimalna wartość potencjału węzłowego, $LZ:=\text{card } W_Z$ oraz podstaw $R:=0$, $J:=Z$, $S:=0$ i przejdź do kroku 2.

Krok 2. Wyznacz, zgodnie z procedurą 4, węzłowo-gałęziową macierz incydencji A. Podstaw

$$\pi_w := \frac{\xi(w)}{k+1} \quad \text{dla } w=1,2,\dots,LZ \text{ i przejdź do kroku następnego.}$$

Krok 3. Podstaw $J:=J+1$, $L:=1$, $\varphi_J^R:=0$, $g_J^R:=0$ i przejdź do kroku 4.

Krok 4. Przyjmij $I:=|A, J, L|$, $N_I:=A(J, L+1)$ oraz oblicz wartości

$$g_J^R := g_J^R + N_I \cdot |\varpi^R(J) - \varpi^R(I)|^{1/k} \cdot \text{SGN}(\varpi^R(J) - \varpi^R(I))$$

$$\varphi_J^R := \varphi_J^R + \frac{N_I}{|\varpi^R(J) - \varpi^R(I)|^{1/k}}$$

Sprawdź, czy $L < S(J)$? Jeżeli tak, podstaw $L:=L+2$ i powtórz krok 4. W przypadku gdy nie, wykonaj krok następny.

Krok 5. Podstaw

$$\delta_J^{(R+1)} := \frac{-g_J^R}{\frac{1}{k} \cdot \varphi_J^R} + QW(J); \quad S := S + |\delta_J^{(R+1)}|;$$

$$\varpi^{(R+1)}(J) := \varpi^{(R)}(J) + \delta_J^{(R+1)}$$

Sprawdź, czy $J < W$? Jeżeli tak, przejdź do kroku 3.

Gdy nie, wykonaj krok następny.

Krok 6. Sprawdź, czy $S < \varepsilon$? Jeśli tak, koniec obliczeń dla metody Crossa, przejdź do kroku 8. W innym przypadku wykonaj krok 7.

Krok 7. Sprawdź, czy $R > LI$? Jeżeli nie, podstaw $J:=Z$; $R:=R+1$ i powrót do kroku 3. Gdy tak, przejdź do kroku 8.

Krok 8. Podstaw $J:=Z$ i oblicz wartość wyrażenia

$$\delta_J := \varpi(1) - \varpi(J) - \frac{\xi(w_J) - \xi(w_1)}{k+1}$$

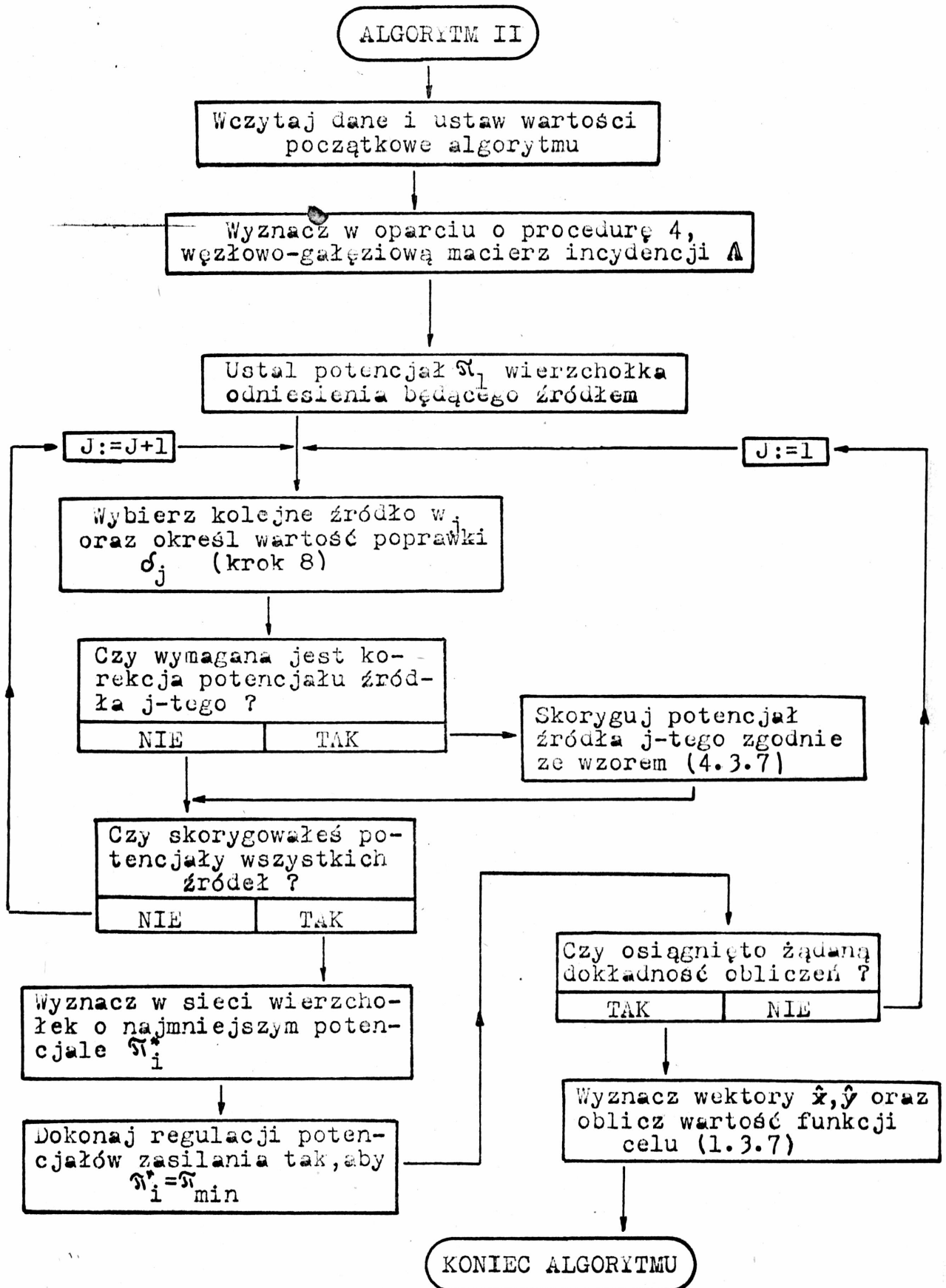
Sprawdź, czy $\delta_J < \varepsilon_J$? Jeśli tak, podstaw $K:=0$ i przejdź do kroku 9. W innym przypadku $K:=1$; $\varpi(J) := \varpi(J) + \delta_J$ i wykonaj krok następny.

Krok 9. Sprawdź, czy $J < Z$? Jeżeli tak, podstaw $J:=J+1$ i powtórz krok 8. Gdy nie, sprawdź czy $K=0$? Jeśli tak, wykonaj krok następny, gdy nie $J:=Z$ i powrót do kroku 3.

Krok 10. Podstaw $\hat{\varpi} = \varpi^{(R)}$ oraz wyznacz i -tą składową wektora potencjałów węzłowych $\hat{\varpi}$ tak, aby

$$\hat{\varpi}_i^{\#} = \min_j \{ \hat{\varpi}_j \mid j=1, 2, \dots, W \}$$

Oblicz wartość wyrażenia $\delta := \hat{\varpi}_i^{\#} - \hat{\varpi}_{\min}$. Następnie podstaw $\hat{\varpi}_w := \hat{\varpi}_w - \delta$ ($w=1, 2, \dots, W$) i przejdź do kroku 11.



Rys.4.6. Schemat działania algorytmu II

Krok 11. W oparciu o zależności (1.2.25), (1.2.23) wyznacz wektory i oblicz wartość funkcji celu (1.3.7). Przejdź do kroku następnego.

Krok 12. Koniec obliczeń algorytmu II.

Działanie przedstawionego algorytmu II sprowadza się do wyliczania w oparciu o metodę Crossa, aktualnych wartości potencjałów wierzchołków π_w (kroki 2-7) oraz takiej ich korekty, aby spełniały one warunek (3.2.36) (kroki 8-9). Z chwilą otrzymania poszukiwanych wartości $\hat{\pi}_w$ ($w=1,2,\dots,W$) wybierana jest z nich wartość najmniejsza $\hat{\pi}_i^*$, która porównywana jest z wymaganą minimalną wartością potencjału węzłowego $\pi_{\min}$. W wyniku tego porównania otrzymujemy wartość poprawki δ (krok 10), o jaką należy zmienić wartości wszystkich potencjałów węzłowych, tak aby $(\forall w \in \{1,2,\dots,W\})(\hat{\pi}_w \geq \pi_{\min})$.

Algorytm II został przedstawiony w wersji opartej na modelu (1.2.21)-(1.2.25) sieci przepływowej, z wykorzystaniem metody Crossa [10]. Przystosowanie powyższego algorytmu do sytuacji, gdy dysponujemy możliwością pomiaru potencjałów węzłowych w sieci rzeczywistej, polega na zastąpieniu kroków 2+7, odczytem aktualnych wartości π_w . Postępowanie takie (rozdz.6.2) czyni algorytm II prostszym i szybszym, umożliwiając uzyskanie optymalnych wartości potencjałów źródeł, w czasie w pełni dopuszczalnym z punktu widzenia wymagań automatycznego sterowania w czasie rzeczywistym.

4.4. Sformułowanie i analiza warunków zbieżności

Przedstawiając w rozdziale poprzednim algorytmy I, II umożliwiające rozwiązanie zagadnienia optymalizacji (1.3.4)-(1.3.7), nie określono warunków zapewniających zbieżność stosowanych procedur obliczeniowych. Z uwagi na fakt, że algorytmy te należą do tej samej grupy metod iteracyjnych o podobnych własnościach i schematach obliczeniowych, w dalszej części poniższego rozdziału rozpatrzmy jedynie algorytm I, pamiętając, że wszystko co zostanie powiedziane o warunkach zbieżności algorytmu I jest także prawdziwe dla algorytmu II. Występowanie ewentualnych różnic zostanie natomiast wyraźnie podkreślone na końcu tego rozdziału.

Rozważmy problem zbieżności algorytmu I oraz jej charakter, które związane są ściśle ze zbieżnością wykorzystywanej w nim iteracyjnej metody Crossa-Lobaczewa, rozwiązywania układów równań nieliniowych postaci (1.3.4)-(1.3.7).

To ostatnie zagadnienie rozpatrywane było w pracach [10],[7], [24], w których wykazano, że jeżeli metoda ta jest zbieżna, to jest to zbieżność rzędu pierwszego (liniowa) o stałej asymptotycznej błęd

$$C = \max_i \{ \lambda_i | i=1,2,\dots,p \} \quad (4.4.1)$$

gdzie $\lambda_i \in \mathbb{R}$ - wartości własne macierzy (4.1.12).

Zgodnie z rekurencyjną zależnością (4.1.14) mamy

$$\xi^{(r)} = [\Phi'(\hat{\Delta})]^r \cdot \xi^{(0)} \quad (4.4.2)$$

zatem ciąg $\xi^{(0)}; \xi^{(1)}; \xi^{(2)}; \dots; \xi^{(r)}$ jest zbieżny do zera, jeżeli

$$\lim_{r \rightarrow \infty} [\Phi'(\hat{\Delta})]^r = 0 \quad (4.4.3)$$

Zapiszmy macierz $\Phi'(\hat{\Delta})$ w postaci

$$\Phi'(\hat{\Delta}) = \sum_{i=1}^p \lambda_i x_i y_i^T \quad (4.4.4)$$

przy czym $x_i, y_i \in \mathbb{R}^p$ to odpowiednio; znormalizowane wektory własne kolumn i wierszy macierzy $\Phi'(\hat{\Delta})$, oraz skorzystajmy ze znanej z algebry [51] własności

Uwaga 4.1

Dla dowolnej macierzy A wymiaru $[n \times n]$, o wartościach własnych $\lambda_1, \lambda_2, \dots, \lambda_n$, wartości własne $\lambda_1^k, \lambda_2^k, \dots, \lambda_n^k$ k -tej potęgi tej macierzy A^k równe są k -tym potęgom wartości własnych $\lambda_1, \lambda_2, \dots, \lambda_n$ ([51], str.475) tj.

$$\lambda_i^k = (\lambda_i)^k \quad i=1,2,\dots,n \quad (4.4.5)$$

Na mocy uwagi 4.1 i zależności (4.4.4) warunek (4.4.3) przyjmie postać

$$\lim_{r \rightarrow \infty} [\Phi'(\hat{\Delta})]^r = \lim_{r \rightarrow \infty} \sum_{i=1}^p \lambda_i^r x_i y_i^T = 0 \quad (4.4.6)$$

a stąd,

$$(\lim_{r \rightarrow \infty} [\Phi'(\hat{\Delta})]^r = 0) \Leftrightarrow ((\forall i \in \mathbb{N}_1^p) (\forall \lambda \in \mathbb{R}^p : \det[\Phi'(\hat{\Delta}) - \lambda I] = 0) (|\lambda_i| < 1)) \quad (4.4.7)$$

Warunek zbieżności (I) [7]

Niech $\lambda_i \in \mathbb{R}$, $i=1,2,\dots,p$ będą wartościami własnymi macierzy $\Phi'(\hat{\Delta})$. Dla zbieżności algorytmu I potrzeba i wystarcza, by wszystkie wartości własne λ_i macierzy (4.1.12) były, co do modułu, mniejsze od jedności tj.

$$(\forall i \in \mathbb{N}_1^p) (\forall \lambda = [\lambda_1; \lambda_2; \dots; \lambda_p]^T | \det[\Phi'(\hat{\Delta}) - \lambda I] = 0) (|\lambda_i| < 1) \quad (4.4.8)$$

Dowód:

Wynika bezpośrednio z przeprowadzonego powyżej rozumowania
(# # #)

Powyższy warunek zbieżności (I) wymaga znajomości punktu optymalnego Δ , określa więc jedynie sposób weryfikacji optymalności otrzymanego rozwiązania. W praktyce, ponieważ $\hat{\Delta}$ jest nieznane, zgodnie z sugestiami prac [7], [51], rozpatrywana jest nie macierz $\Phi'(\hat{\Delta})$, lecz $\Phi'(\Delta^k)$. Oczywiście postępowanie takie jest dopuszczalne, jeżeli istnieje otoczenie $(\hat{\Delta}, \delta)$ punktu $\hat{\Delta}$, takie że każdy wektor Δ należący do tego otoczenia spełnia warunek (4.4.8). Ponadto musi być zapewnione, że przybliżenie Δ^k , otrzymane po wykonaniu skończonej liczby iteracji, należy będzie do wspomnianego otoczenia.

Warunek zbieżności (II) [7]

Niech λ_i , $i=1,2,\dots,p$ będą wartościami własnymi macierzy $\Phi'(\Delta)$, gdzie $\|\hat{\Delta} - \Delta\| < \delta$; $\delta \in \mathbb{R}^+$.

Algorytm I jest zbieżny, jeżeli spełnione są warunki

$$(\exists \delta \in \mathbb{R}^+) (\forall \Delta \in \mathbb{R}^p; \|\hat{\Delta} - \Delta\| < \delta) (|\lambda_i| < 1) \quad (4.4.9)$$

$$(\forall \delta \in \mathbb{R}^+) (\exists k \in \mathbb{N}) (\forall i \geq k) (\forall \Delta^i \in \mathbb{R}^p) (\|\hat{\Delta} - \Delta^i\| < \delta) \quad (4.4.10)$$

Dowód: [7]

Założmy, że w wyniku obliczeń algorytmu I otrzymaliśmy ciąg przybliżeń $\Delta^{(k)}$, $\Delta^{(k+1)}$, ..., $\Delta^{(R)}$ oraz odpowiednio $\Phi'(\Delta^k)$; $\Phi'(\Delta^{(k+1)})$; ..., $\Phi'(\Delta^R)$, przy czym Δ^k spełnia warunek (4.4.10), a więc moduły wartości własnych macierzy $\Phi'(\Delta^k)$ są, zgodnie z (4.4.9), mniejsze od jedności. Ponieważ błąd $\varepsilon^{(i)}$ na mocy (4.1.14) wyrażony jest wzorem

$$\xi^{(i)} = \Phi'(\Delta^k) \cdot \Phi'(\Delta^{k+1}) \dots \Phi'(\Delta^{(i-1)}) \cdot \xi^{(k)}, \quad i=(k+1), \dots, r \quad (4.4.11)$$

stąd, jeżeli Δ^k należy do otoczenia $\mathcal{U}(\Delta, \delta)$, to z (4.1.21)

$$\|\xi^{(k+1)}\| \leq \|\Phi'(\Delta^k)\| \cdot \|\xi^{(k)}\| \leq \|\xi^{(k)}\| \leq \|\hat{\Delta} - \Delta^k\|$$

a więc również $\Delta^{(k+1)}$ oraz pozostałe przybliżenia $\Delta^{(k+2)}, \dots, \Delta^{(r)}$ należą także do tego otoczenia.

Natomiast z (4.4.9) mamy

$$(\forall i \in \{k, (k+1), \dots, r\}) (\forall j \in \{1, \dots, p\}) (|\lambda_j^i| < 1)$$

gdzie: λ_j^i - j-ta wartość własna macierzy $\Phi'(\Delta^i)$.

Niech $\Phi'(\Delta^n)$ będzie macierzą o największej wartości własnej λ_j^n , tj.

$$\Phi'(\Delta^n) = \max \{ \|\Phi'(\Delta^i)\| \mid i=k, (k+1), \dots, r \}$$

oraz

$$\|\Phi'(\Delta^n)\| = \lambda_j^n = \lambda \wedge |\lambda| < 1 \quad (4.4.12)$$

Korzystając z (4.4.11), (4.4.12) oszacujmy wartość graniczną błędu $\xi^{(i)}$

$$\lim_{i \rightarrow \infty} \xi^{(i)} \leq \lim_{i \rightarrow \infty} [\|\Phi'(\Delta^n)\|]^{i-k} \cdot \xi^{(k)} = \xi^{(k)} \cdot \lim_{i \rightarrow \infty} [\|\Phi'(\Delta^n)\|]^i = 0$$

co oznacza zbieżność algorytmu I, kończąc dowód.

(* * *)

Obydwa przedstawione warunki zbieżności algorytmu I związane są ze skomplikowanym zagadnieniem numerycznego wyznaczania wartości własnych λ_j^i . Problem ten, nawet w przypadku niedużych rozmiarów macierzy $\Phi'(\Delta)$, pociąga za sobą znaczny wzrost czasu obliczeń. Niedogodność ta może być wyeliminowana, jeżeli skorzystamy z twierdzenia Gerszgorina [51] dotyczącego lokalizacji i szacowania wartości własnych macierzy.

Twierdzenie Gerszgorina [51]

Niech $\Phi'(\Delta) = [\varphi'_{ji}]_{p \times p}$ natomiast c_i, c_j ($i=1, 2, \dots, p$; $j=1, \dots, p$) będą kołami o środkach φ'_{ii} , φ'_{jj} i promieniach

$$r_j = \sum_{\substack{k=1 \\ k \neq j}}^p |\varphi'_{jk}| \quad r_i = \sum_{\substack{k=1 \\ k \neq i}}^p |\varphi'_{ki}| \quad i, j=1, 2, \dots, p \quad (4.4.13)$$

oraz

$$D_j = \bigcup_{j=1}^p C_j \quad ; \quad D_i = \bigcup_{i=1}^p C_i \quad (4.4.14)$$

Wtedy wszystkie wartości własne λ_i ($i=1,2,\dots,p$) macierzy $\Phi'(\Delta)$ leżą w obszarze ([51], str.478)

$$\bigcup_{j=1}^p C_j = D_j \quad \wedge \quad \bigcup_{i=1}^p C_i = D_i = D \quad (4.4.15)$$

Na mocy powyższego twierdzenia oraz w związku z faktem, że (zgodnie z (4.1.15)) macierz $\Phi'(\Delta)$ posiada na głównej przekątnej elementy zerowe, obszar D będzie kołem o środku w początku układu i promieniu r , równym

$$r = \min \{ R_1, R_2 \} \quad (4.4.16)$$

gdzie $R_1 = \max_i \{ r_i | i=1,2,\dots,p \}$ $R_2 = \max_j \{ r_j | j=1,2,\dots,p \}$

Pozwala to na sformułowanie prostego kryterium zbieżności algorytmu I. W celu określenia czy ciąg wartości $\varepsilon^{(1)}, \varepsilon^{(2)}, \dots, \varepsilon^{(k)}$ dąży do zera, należy wyliczyć wartość promienia r , określoną wzorem (4.4.16) i sprawdzić, czy $r < 1$?

Spełnienie tej nierówności świadczy o tym, że wartości własne macierzy $\Phi'(\Delta)$ są mniejsze od jedności, natomiast sytuacja przeciwna nie daje podstaw do rozstrzygnięcia zbieżności algorytmu I.

Warunek zbieżności (III) [7]

Oznaczmy przez $\Phi'(\Delta)$ macierz (4.1.12) określoną w punkcie Δ , gdzie $\|\hat{\Delta} - \Delta\| < \delta$; $\delta \in \mathbb{R}^+$.

Jeżeli elementy φ'_{ij} tej macierzy spełniają warunek

$$(\forall \Delta \in \mathcal{U}(\hat{\Delta}, \delta)) (\forall i, j \in \mathbb{N}_1^p) \left(\sum_{k=1}^p |\varphi'_{ik}(\Delta)| < 1 \vee \sum_{k=1}^p |\varphi'_{kj}(\Delta)| < 1 \right) \quad (4.4.17)$$

oraz prawdziwa jest zależność (4.4.10), to algorytm I jest zbieżny liniowo [7] ze stałą asymptotyczną błędu nie większą niż;

$$\min_{i,j} \left\{ \sum_{k=1}^p |\varphi'_{ik}(\Delta)| ; \sum_{k=1}^p |\varphi'_{kj}(\Delta)| \mid i, j=1,2,\dots,p \right\} \geq c \quad (4.4.18)$$

Otrzymana postać warunku zbieżności algorytmu I, poprzez zastąpienie zależności (4.4.9), prostszą do sprawdzenia formą (4.4.17), wydatnie upraszcza rozstrzygnięcie czy w danym przypadku, posługując się tym algorytmem, otrzymamy rozwiązanie optymalne.

Ponadto dla metody Crossa-Lobaczewa wykazano [56], że jeżeli punkt startowy Δ^0 (spełniający warunek (4.4.9)) nie znajduje się w otoczeniu $\mathcal{U}(\Delta, \delta)$, to odpowiedni wybór dendrytu D, stanowiącego podstawę budowy macierzy oczkowej B, zapewnia iż po wykonaniu skończonej liczby iteracji otoczenie to zostanie osiągnięte.

W tej samej pracy stwierdzono również, że wspomnianego wyżej wyboru dendrytu należy dokonać tak, aby składał się on z gałęzi o najmniejszych wartościach parametrów d_i . Stąd, w celu zapewnienia zbieżności algorytmu I, przed przystąpieniem do obliczeń należy, stosując jeden ze znanych algorytmów, wyznaczyć dendryt D^* o minimalnym koszcie, a następnie wybrać taki punkt startowy Δ^0 , który spełnia zależność (4.4.17).

Przedstawiony w rozdz. 4.3 algorytm II jest ilustracją możliwości zastosowań utylitarnych opracowanej metody do potrzeb sterowania rzeczywistymi sieciami przepływowymi. Ponieważ jednak nie dysponowano możliwością dokonywania pomiarów na obiekcie rzeczywistym, konieczna była symulacja jego działania. O zbieżności algorytmu II decyduje więc w tym wypadku, zbieżność metody wybranej do symulacji sieci przepływowej tj. metody Crossa [10].

Natomiast w sytuacji, gdy w algorytmie II wykorzystywane są pomiary na obiekcie rzeczywistym, zbieżność tego algorytmu zapewnia przyjęta metoda regulacji (zapewniająca spełnienie warunku (3.2.36)).

4.5. Badania testowe i analiza własności opracowanych algorytmów

W poprzednich rozdziałach rozprawy określono warunki konieczne i dostateczne optymalności zagadnienia (1.3.4)-(1.3.7). Wykazano również, że wykorzystanie sformułowanych zależności, pozwala na opracowanie nowych algorytmów rozwiązujących rozpatrywany problem optymalizacji.

Opracowane w rozdz. 4.3 algorytmy zostały oprogramowane w języku FORTRAN IV-E i przetestowane na minikomputerze MERA-400.

Poniżej przedstawione zostaną wyniki testów, które umożliwiają ocenę efektywności opracowanych przez autora algorytmów w zależności od typu i struktury sieci nieliniowej (1.2.21)-(1.2.23) oraz specyfiki rozpatrywanego zagadnienia.

Jako podstawowe kryteria oceny przyjęto:

- czas trwania obliczeń numerycznych (T)
- stopień zajętości PAO maszyn cyfrowych (M)
- liczbę iteracji (L) jakie należy wykonać dla osiągnięcia żądanej dokładności
- dokładność (ϵ) osiągana po ustalonym czasie obliczeń.

Przy doborze zadań testowych szczególną uwagę zwrócono na: wpływ wymiarowości problemu optymalizacji (1.3.4)-(1.3.7), wpływ topologii i stopnia nieliniowości sieci (1.2.21)-(1.2.23) oraz wpływ położenia ekstremum, na efektywność testowanych algorytmów. W tym celu rozpatrzono szereg wielowymiarowych problemów optymalizacji postaci (1.3.4)-(1.3.7), z których dwa przedstawione zostaną w rozdziale następnym. Zagadnienia te zostały rozwiązane dla różnych stopni nieliniowości sieci, zarówno przy pomocy algorytmu I jak również algorytmu II, a otrzymane wyniki przedstawione zostały odpowiednio: w Tabeli 4.1 oraz Tabeli 4.3.

W przypadku algorytmu I, porównanie otrzymanych czasów (kolumna 5, Tab.4.1) w odniesieniu do topologii optymalizowanej sieci, pozwala stwierdzić wyraźną zależność czasu trwania obliczeń od ilości obwodów podstawowych oraz struktury połączeń sieci S .

Zwiększenie czasu obliczeń wraz ze wzrostem ilości obwodów podstawowych, związane jest między innymi z większą ilością operacji arytmetycznych składających się na jedną iterację (testy nr VII, VIII, Tab.4.1). Natomiast wpływ struktury połączeń sieci na czas obliczeń wynika ze specyfiki metody Crossa-Łobaczewa [10], zakładającej brak wzajemnych oddziaływań pomiędzy składowymi wektora (4.1.1) w danej iteracji. Siła tych oddziaływań jest proporcjonalna do liczby stabilności wewnętrznej (def.1.10) grafu (1.2.13), opisującego strukturę połączeń sieci. Im liczba stabilności wewnętrznej grafu (1.2.13) jest mniejsza, tym oddziaływania te są słabsze, co w rezultacie zwiększa szybkość zbieżności (testy VIII, XII, Tab.4.1) zmniejszając liczbę iteracji niezbędnych do osiągnięcia żądanej dokładności.

Tabela 4.1. Wyniki badań testowych algorytmu I

Nr testu	Ilość węzłów	Ilość łuków	Stopień nielinowości	Czas obliczeń	Wymagana dokładność	Liczba iteracji	Wymagana zajętość PAO
	W	N	K	T	EPS	L	XFAM
1	2	3	4	5	6	7	8
I	4	4	2	2s	10^{-6}	13	7,0 k
II	4	4	4	2s	10^{-6}	13	7,0 k
III	4	4	6	2s	10^{-6}	11	7,0 k
IV	4	4	8	2s	10^{-6}	13	7,0 k
V	5	8	2	2s	10^{-6}	14	7,2 k
VI	10	20	2	27s	10^{-6}	79	7,3 k
VII	12	27	2	23s	10^{-6}	85	7,4 k
VIII	21	40	2	36s	10^{-6}	82	8,5 k
IX	21	40	4	37s	10^{-6}	84	8,5 k
X	21	40	6	36s	10^{-6}	83	8,5 k
XI	21	40	8	38s	10^{-6}	87	8,5 k
XII	76	94	2	58s	10^{-6}	132	11,5 k

Otrzymane wyniki badań testowych, zamieszczone w Tab.4.1 wykazują ponadto również nieznaczny wpływ stopnia nielinowości problemu (1.3.4)-(1.3.7) na czas trwania obliczeń (testy VIII,XI, Tab. 4.1), co jest naturalną konsekwencją przyjętej metody rozwiązywania zagadnienia. Nie stwierdzono natomiast zależności pomiędzy przyrostem czasu trwania obliczeń spowodowanym wzrostem nielinowości zagadnienia, a wielkości optymalizowanej sieci.

W ramach obliczeń testowych zbadana została także wrażliwość opracowanego algorytmu na dobór punktu startowego. Otrzymane wyniki stanowią potwierdzenie rezultatów zamieszczonych w pracy [56], w której zagadnienie to (w odniesieniu do metody Crossa-Łobaczewa) poddano szczegółowej analizie. Z tego powodu, rezygnując z ich powtarzania, w poniższej pracy ograniczymy się jedynie do przedstawienia tych wniosków z nich wypływających, które w sposób istotny ułatwiają wykorzystanie algorytmu I.

Tabela 4.2. Wyniki badań efektywności procedury przyspieszającej

Nr testu	Cross-Łobaczew		Cross-Łobaczew z procedurą przyspieszającą (L1-L2)																	
			2-2		2-4		2-6		4-2		4-4		4-6		6-8		6-4		4-8	
	T	L	T	L	T	L	T	L	T	L	T	L	T	L	T	L	T	L	T	L
I	4	110	2	18	2	12	2	16	2	22	2	13	2	17	2	16	2	14	2	18
V	2	32	3	51	3	51	2	30	2	24	2	15	2	15	2	15	2	14	2	15
VI	27	79	-	-	45	130	29	84	-	-	-	-	39	139	38	134	54	200	35	30
VII	29	118	-	-	38	193	31	98	-	-	52	179	23	85	26	92	34	122	24	87
VIII	55	123	-	-	110	206	43	93	-	-	89	205	42	97	36	82	42	101	36	89
IX	55	123	-	-	110	206	43	95	-	-	90	205	42	95	36	84	41	99	36	89
X	55	122	-	-	110	209	43	96	-	-	90	210	42	98	36	81	42	105	36	90
XI	55	124	-	-	111	209	43	94	-	-	90	212	41	96	36	80	42	100	36	90
XII	203	480	-	-	130	235	103	213	-	-	86	197	95	216	58	132	91	202	92	208

Uwaga: T - czas obliczeń w sekundach, L - liczba iteracji, 2-4 - wartości parametrów (L1-L2)
 "-" brak danych ze względu na niebezpieczeństwo metody

Niezwykle ważną cechą algorytmu I jest mała wrażliwość na dobór punktu startowego x^0 . Szybkość obliczeń tego algorytmu zależy natomiast od wyboru dendrytu D , jednakże każdorazowe zastoso-
sowanie procedury przyspieszającej, zbieżność tę wyraźnie osłabia. Znaczne skrócenie czasu obliczeń ma miejsce w przypadku, gdy suma parametrów d_i wybranego dendrytu jest najmniejsza. Wtedy bowiem stała asymptotyczna błędu (4.1.21) przyjmuje małe wartości, zwiększając zbieżność algorytmu.

W rozdz. 4.1 i rozdz. 4.4 pokazano, że metoda Crossa-Łobaczewa jest zbieżna według normy, przy czym jest to zbieżność typu liniowego. Stąd, jeżeli czas trwania jednej iteracji jest dla danego zadania wielkością stałą, to jeśli po wykonaniu " k " iteracji zwiększono dokładność rozwiązania o jedną liczbę znaczącą, to w celu przejścia do dwóch liczb znaczących należy zwiększyć czas obliczeń dwukrotnie (rys.4.2). Własność ta powoduje, że szybkość zbieżności algorytmu I, opartego na metodzie Crossa-Łobaczewa maleje wraz ze zbliżaniem się do minimum.

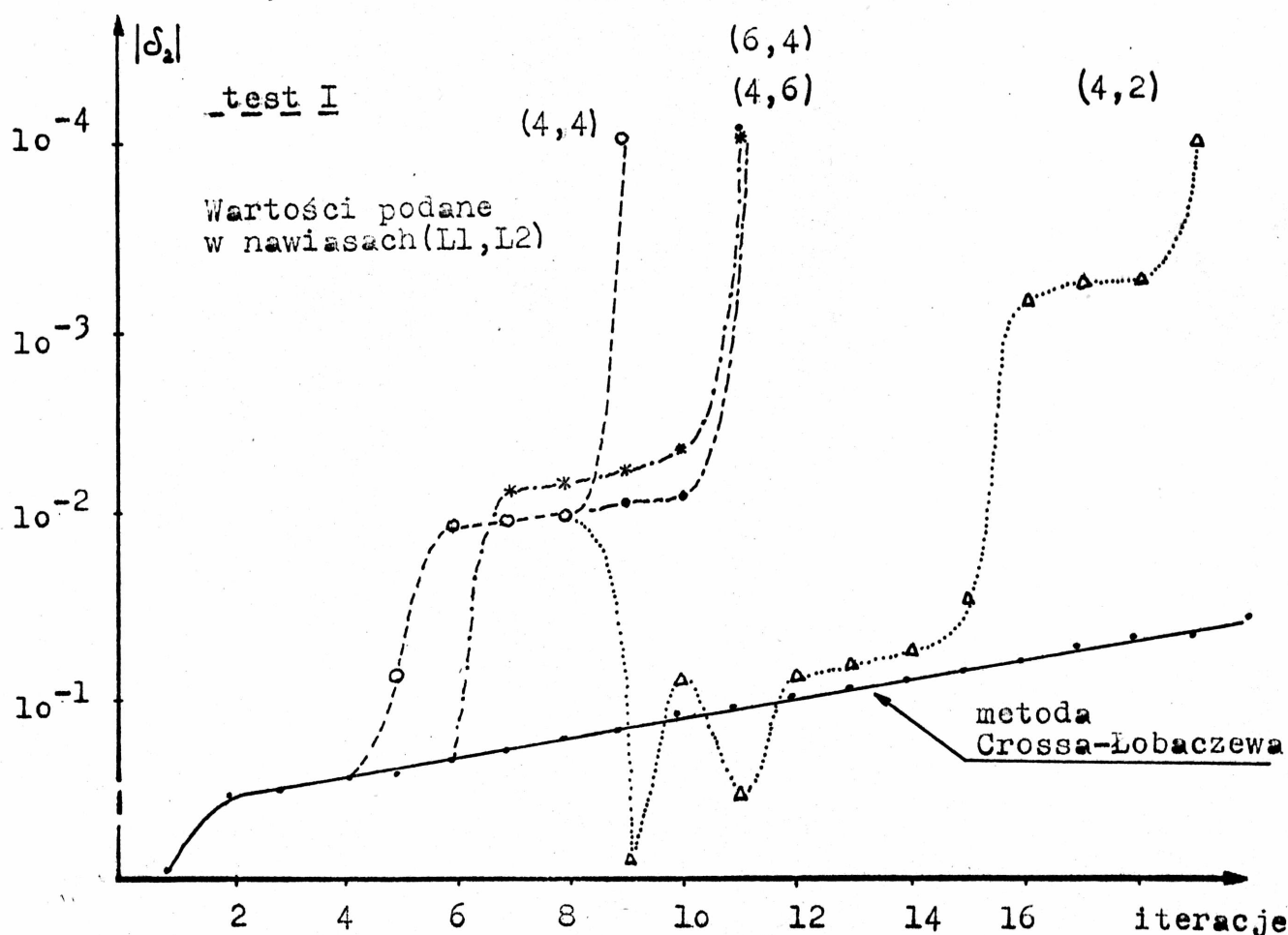
Aby zapobiec temu zjawisku, wykorzystano opracowaną przez autora procedurę przyspieszającą (rozdz.4.2) opartą na liniowej zbieżności metody Crossa-Łobaczewa. Otrzymane wyniki przeprowadzonych badań testowych, zamieszczone w Tab.4.2, dowodzą wysokiej efektywności opracowanej procedury, a także umożliwiają empiryczne określenie chwili początkowej ($L1$) oraz częstości ($L2$) uruchomienia procedury 3.

Podkreślenia wymaga jednak fakt, że wykorzystanie tej procedury możliwe jest jedynie wtedy, gdy spełniony jest warunek (4.2.6). W przypadku przeciwnym, uruchomienie procedury przyspieszającej wyraźnie obniża efektywność przeprowadzanych obliczeń. Sytuacja taka wystąpić może w dwóch przypadkach. Po pierwsze wtedy, gdy w procedurze redukcji wygenerowany zostanie taki dendryt D (względem którego budowana jest macierz obwodów podstawowych B_D), że określona zależnością (4.1.12) macierz $\Phi'(\Delta)$ posiada nieliniowe dzielniki elementarne [51] (Test VI, tab. 4.2). Po drugie w przypadku, jeżeli procedura przyspieszająca uruchomiona zostanie za wcześnie, tj. w chwili gdy bieżący wskaźnik iteracji $n < k$ (por. (4.2.6)) (Testy VI-XII, (2-2), tab.4.2).

Wystąpienie co najmniej jednego z opisanych powyżej przypadków sygnalizowane jest w procedurze 3 (krok 4), a następnie podejmowana jest decyzja o przerwaniu wykonywania procedury przyspieszającej i powrocie do metody Crossa-Łobaczewa.

Analizując zestawione w tabeli 4.2 liczby iteracji (L) oraz czasy trwania obliczeń (T), otrzymane dla oryginalnej metody Crossa-Łobaczewa i metody zmodyfikowanej, o różnych wartościach parametrów sterujących L_1, L_2 , nietrudno dostrzec wpływ właściwego doboru ich wartości na efektywność obliczeń algorytmu I.

W celu zilustrowania omawianego zagadnienia na rys.4.7 przedstawiono wartości jednej ze składowych wektora poprawek δ^k , obliczanych w kolejnych iteracjach w czasie wykonywania Testu I (Tab.4.2), z uwzględnieniem procedury przyspieszającej. Każda z krzywych odpowiada innym wartościom parametrów L_1, L_2 , określających; numer iteracji, od której należy uruchomić procedurę 3 oraz ilość iteracji pomiędzy kolejnymi jej wywołaniami. Dodatkowo linią ciągłą zaznaczono wyniki otrzymane przy wykorzystaniu oryginalnej metody Crossa-Łobaczewa.



Rys.4.7. Zależność dokładności rozwiązania od wartości parametrów L_1, L_2 procedury przyspieszającej

Najlepsze efekty obliczeń uzyskano w przypadku, gdy wartości L_1, L_2 , każdorazowo dobierano do topologii optymalizowanej sieci (dla testu V, tab. 4.2, optymalnego wartości to $L_1=6, L_2=4$, a dla testu XII - $L_1=6, L_2=8$).

Jednakże możliwym jest takie przyjęcie wartości L_1, L_2 dla których efektywność obliczeń algorytmu I jest zadowalająca dla sieci o zróżnicowanych strukturach przestrzennych (przeprowadzone testy wskazują, że wartościami tymi mogą być $L_1=6, L_2=8$). Należy jednak pamiętać, że zbyt małe wartości parametrów L_1, L_2 powodują wczesne uruchomienie procedury przyspieszającej, kiedy to warunek (4.1.20) nie jest jeszcze spełniony (krzywa 4-2, rys.4.7). Tym samym, efekty zastosowań procedury 3 są w tym przypadku niewielkie, a nawet powodują rozbieżność omawianego algorytmu. Z drugiej strony, zbyt duże wartości L_1, L_2 opóźniają uruchomienie procedury przyspieszającej, uniemożliwiając osiągnięcie wszystkich korzyści z tego faktu wynikających.

W przypadku algorytmu II, wyniki badań testowych przedstawione w tabeli 4.3, pozwalają wnioskować o występowaniu istotnej zależności czasu trwania obliczeń od stopnia oraz ilości wierzchołków rozpatrywanej sieci. Ponadto wykazują one istnienie cech negatywnych, wynikających bezpośrednio ze słabej zbieżności metody Crossa, wykorzystywanej w tym algorytmie.

Duża liczba iteracji, pociągająca za sobą wydłużenie się czasu trwania obliczeń oraz mała dokładność otrzymywanego rozwiązania, powodują, że algorytmu II nie można uważać za konkurencyjny w stosunku do algorytmu I, przy rozwiązywaniu zadań optymalizacji postaci (1.3.4)-(1.3.7) w tych przypadkach, kiedy zachodzi konieczność symulacji sieci optymalizowanej.

Główną zaletą algorytmu II jest jednak to, że wszystkie operacje w nim określone wykonywane są nie na wartościach przepływów, a w oparciu o wartości potencjałów wierzchołków (1.2.19). Duża wartość tej własności, wynika stąd, że w przypadku, gdy problem optymalizacji (1.3.4)-(1.3.7) odniesiony jest do sieci rzeczywistej, to proces symulacji można zastąpić pomiarami przeprowadzanymi bezpośrednio na obiekcie. Wtedy zastosowanie algorytmu I, wymagającego znajomości parametrów sieci oraz jej symulacji, jest niecelowe. Natomiast w celu uruchomienia algorytmu II wystarczy znajomość potencjałów wierzchołków będących źródłami, których pomiar (w wyniku, którego otrzymujemy wartości $\pi_1, \pi_2, \dots, \pi_z$) zastępuje kroki 2÷7 algorytmu.

Tabela 4.3. Wyniki badań testowych algorytmu II

Nr testu	Ilość węzłów	Ilość łuków	Stopień nieliniowości	Ocena obliczeń	Wymagana dokładność	Liczba iteracji	Wymagana zajętość PAO
	W	N	K	T	EPS	L	XFAM
1	2	3	4	5	6	7	8
I	4	4	2	4s	10^{-3}	7	5,7k
II	4	4	4	4s	10^{-3}	7	5,7k
III	4	4	6	5s	10^{-3}	9	5,7k
IV	4	4	8	7s	10^{-3}	12	5,7k
V	5	8	2	10s	10^{-3}	15	6,0k
VI	5	8	4	10s	10^{-3}	15	6,0k
VII	10	20	2	73s	10^{-3}	140	6,8k
VIII	21	40	2	83s	10^{-3}	160	7,2k
IX	21	40	4	81s	10^{-3}	158	7,2k
X	21	40	6	87s	10^{-3}	173	7,2k
XI	21	40	8	76s	10^{-3}	147	7,2k
XII	76	94	2	-	10^{-3}	-	11,7k

W świetle przeprowadzonych badań testowych i przedstawionej analizy nasuwają się następujące wnioski:

- ze względu na opisane własności opracowanych algorytmów, w przypadku konieczności symulacji sieci przepływowej, korzystniejszym jest zastosowanie w procesie rozwiązywania zagadnienia (1.3.4)-(1.3.7) algorytmu I,

- ustalając wartości parametrów sterujących procedurą przyspieszającą zalecanym jest przestrzeganie zasady doboru tych stałych (L1, L2) w oparciu o topologię optymalizowanej sieci. W przypadku konieczności określenia wartości parametrów L1, L2, dla serii zagadnień optymalizacji (1.3.4)-(1.3.7), należy unikać zbyt małych wartości L1, L2. (Proponowane wartości minimalne, wynikające z otrzymanych wyników badań testowych wynoszą L1 = 6, L2 = 8)

- jeżeli dysponujemy możliwością przeprowadzenia pomiarów na rzeczywistej sieci przepływowej lub jej modelu, wtedy znacznie lepsze efekty daje zastosowanie mniej skomplikowanego i szybszego algorytmu II.

4.6. Przykłady obliczeniowe

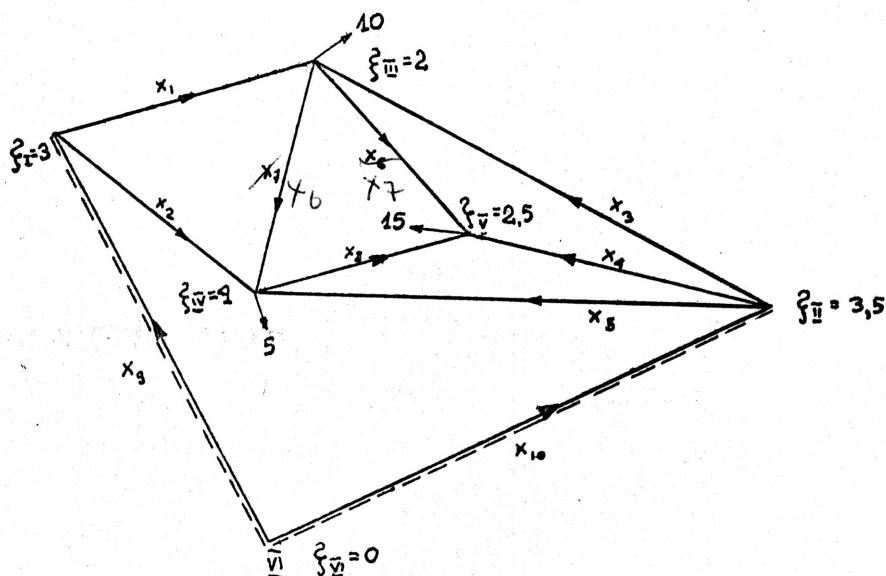
W celu pełniejszego zilustrowania przebiegu obliczeń i metodyki działania opracowanych algorytmów, podane zostaną dwa przykłady obliczeniowe rozwiązywania problemu optymalizacji (1.3.4)-(1.3.7) w oparciu o algorytm I oraz algorytm II.

Przykłady, które zostaną przedstawione, rozwiązane zostały przy użyciu programów #ALG1 oraz #ALG2, których postacie źródłowe zamieszczono w rozdziale 9 (dodatek), gdzie również znajdują się otrzymane tabulogramy wyników.

PRZYKŁAD 1. (Algorytm I)

Dana jest nieliniowa sieć przepływowa stopnia drugiego ($k=2$) przedstawionej na rys.4.9. Wyznaczyć wartości przepływów w poszczególnych gałęziach w sposób zapewniający minimalizację funkcji celu określonej zależnością (4.6.1), wyrażającą straty ponoszone w sieci na dostarczenie wymaganej ilości czynnika od źródeł Z_1 , Z_2 do odbiorców.

gdzie to oznaczono?



Rys.4.8. Struktura przestrzenna sieci przepływowej z przykładu 1 (test V)

Model matematyczny rozpatrywanego zagadnienia optymalizacji
przyjmie więc postać:

$$f(x, y) = \sum_{i=1}^8 x_i \cdot y_i \rightarrow \min \quad (4.6.1)$$

przy ograniczeniach:

$$\begin{aligned} \checkmark x_1 + x_3 - x_6 - x_7 &= 10 \\ 2 x_2 + x_5 - x_8 + x_6 &= 5 & x_2 + x_7 - x_8 + x_5 &= 5 \\ 2 x_7 + x_8 + x_4 &= 15 & x_6 + x_4 + x_8 &= 15 \\ \checkmark x_3 + x_4 + x_5 - q_2 &= 0 \\ \checkmark x_1 + x_2 - q_1 &= 0 \end{aligned} \quad (4.6.2)$$

$$\begin{aligned} y_2 - y_5 + y_4 - y_7 - y_1 &= 0 & y_2 - y_5 + y_4 - y_6 - y_1 &= 0 \\ 2 y_3 + y_7 - y_4 &= 0 & y_3 + y_6 - y_4 &= 0 \\ 2 y_6 + y_5 + y_4 - y_7 &= 0 & y_6 - y_4 + y_5 - y_7 &= 0 \\ \checkmark y_8 - y_4 + y_5 &= 0 \end{aligned} \quad (4.6.3)$$

gdzie

$$y_i = d_i \cdot x_i^2 \cdot \text{sgn } x_i + k_i \quad i=1, 2, \dots, 8 \quad (4.6.4)$$

1. Dane wejściowe do obliczeń (krok 1)

$$N=8, W=5, Z=2, K=2, EPS=10^{-5}, L1=4, L2=6$$

$$P(I, J) = \left[\begin{array}{c|c|c|c|c|c|c|c|c|c} 1 & 1 & 2 & 2 & 2 & 3 & 3 & 4 \\ 3 & 4 & 3 & 5 & 4 & 4 & 5 & 5 \\ \hline 0,01 & 0,02 & 0,04 & 0,05 & 0,04 & 0,02 & 0,03 & 0,01 \end{array} \right]$$

$$Q = [0; 0; 10; 5; 15]^T \quad H = [-1; 1; -1,5; -1; 0,5; 2; 0,5; -1,5]^T$$

2. Problem transformowany (krok 2)

$$N'=10; W'=6; Z=2; K=2; EPS=10^{-5}; L1=4; L2=6$$

$$P'(I, J) = \left[\begin{array}{c|c} P(I, J) & \begin{matrix} 6 & 6 \\ 2 & 1 \\ 0 & 0 \end{matrix} \end{array} \right] \quad Q' = [0, 0, 10, 5; 15; -30]^T$$

$$H' = [-1; 1; -1,5; -1; 0,5; 2; 0,5; -1,5; 1; 1, 16]^T$$

3. Macierz B' (krok 4)

$$B' = \left[\begin{array}{c|c|c|c|c} 2 & -5 & 4 & -7 & -1 \\ 3 & 7 & -4 & 0 & 0 \\ \hline 6 & -5 & 4 & -7 & 0 \\ 8 & -4 & 5 & 0 & 0 \\ 9 & 4 & -7 & -1 & -10 \end{array} \right] \quad SP = \left[\begin{array}{c} 5 \\ 3 \\ 4 \\ 3 \\ 5 \end{array} \right]$$

4. Problem zredukowany (krok 5)

$$B y = 0$$

$$y_i = d_i \cdot x_i^2 \cdot \text{sgn } x_i + h_i \quad i=1,2,\dots,10$$

oraz punkt startowy

$$x_0 = [30; 0; 0; -5; 5; 0; 20; 0; 0; 30]^T$$

5. Wartości pierwszej poprawki (dla problemu zredukowanego) wynoszą (wg (4.1.9))

$$\delta^1 = [1.1110; 1.9443; -3.0378; 3.6210; -2.5849]^T$$

a pierwsze przybliżenie rozwiązania

$$x^1 = [11.6942; 6.9443; 3.8890; 3.9639; 3.5086; -0.0378; -5.6210; 5.4151]^T$$

6. Ponieważ nie osiągnięto wymaganej dokładności (tj. $\|\delta^1\| > 10^{-5}$) ponownie obliczamy poprawki

$$\delta^2 = [-.1761; .3483; -3.6266; 2.1790; -2.3819]^T$$

a stąd

$$x^2 = [10.0705; 7.2027; 4.0651; 4.1669; 4.4049; -3.6644; 7.8000; 3.0332]^T$$

7. Wartości poprawek oraz przybliżeń następnych wynoszą;

$$\delta^3 = [.5952; .5885; -1.3345; 1.0349; -.8434]^T$$

$$\delta^4 = [.1519; .2524; -.6376; .4422; -.3944]^T$$

$$x^4 = [10.3225; 8.1336; 3.3180; 3.9275; 4.2984; -5.6365; 9.2771; 1.7954]^T$$

8. Ponieważ wartość parametru $L1=4$, po zakończeniu czwartej iteracji uruchamiana jest procedura przyspieszająca (krok 11) wyznaczająca poprawki (krok 13)

$$\delta_p^0 = [.0848; .1895; -.5833; .3299; -.3465]^T$$

9. Wartości kolejnych 6-ciu poprawek wynoszą

$$\delta^5 = [.1485; .0536; .0102; .0424; .0089]^T$$

$$\delta^6 = [-.0223; -.0005; -.00096; .0014; -.0014]^T$$

$$\delta^7 = [-.0002; -.0002; -.0007; .00073; -.00079]^T$$

$$\delta^8 = [-.000009; -.00004; -.0004; .00041; -.00042]^T$$

$$\delta^9 = [.000015; .000008; -.00026; .0002; -.00022]^T$$

$$\delta^{10} = [.000016; .00002; -.00014; .0001; -.00011]^T$$

oraz

$$\mathbf{x}^{10} = [10.3002 \mid 8.3760 \mid 3.1399 \mid 3.8928 \mid 4.2911 \mid -6.2122 \mid 9.6523 \mid 1.4549]^T$$

10. Ponieważ $L1+L2=10$, po zakończeniu 10-tej iteracji, ponownie uruchamiana jest procedura przyspieszająca (krok 11), która wyznacza poprawki

$$\delta_p^1 = [-.00001 \mid -.0001 \mid -.0002 \mid .0001 \mid -.0001]^T$$

11. Wartości następnych 3-ch poprawek wynoszą

$$\delta^{11} = [.00019 \mid .00007 \mid .00001 \mid -.000005 \mid .000004]^T$$

$$\delta^{12} = [-.000003 \mid -.000004 \mid .000005 \mid -.000002 \mid .000002]^T$$

$$\delta^{13} = [-.000001 \mid -.000002 \mid .000002 \mid -.0000008 \mid .0000006]^T$$

natomiast

$$\mathbf{x}^{13} = [10.3002 \mid 8.3760 \mid 3.1398 \mid 3.8928 \mid 4.2911 \mid -6.2123 \mid 9.6524 \mid 1.4548]^T$$

Ponieważ osiągnięto wymaganą dokładność rozwiązania (tj. $\|\delta^{13}\| < 10^{-5}$), proces obliczeń zostaje zakończony.

Rozwiązanie problemu (4.6.1)-(4.6.4) stanowią wektory:

$$\hat{\mathbf{x}} = [10.3002 \mid 8.3760 \mid 3.1398 \mid 3.8928 \mid 4.2911 \mid -6.2123 \mid 9.6524 \mid 1.4548]^T$$

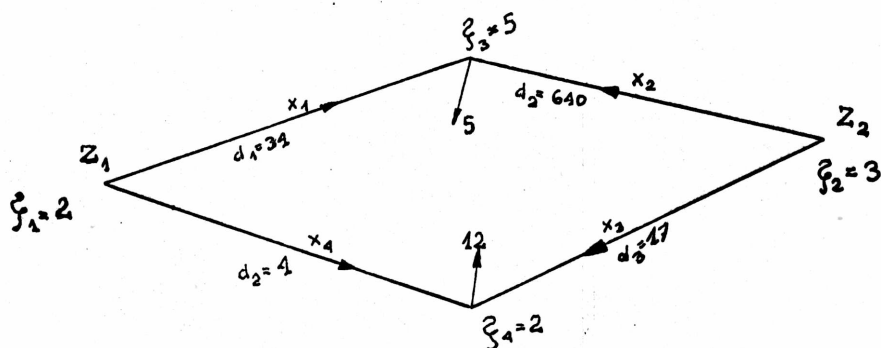
$$\hat{\mathbf{y}} = [-.0609 \mid -2.4032 \mid -1.1057 \mid -.2423 \mid -1.2365 \mid -1.6578 \mid -3.8634 \mid -1.5212]^T$$

dla których funkcja celu (4.6.1) przyjmie wartość

$$f(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = \hat{\mathbf{x}}^T \cdot \hat{\mathbf{y}} = 80.2795$$

PRZYKŁAD 2. (Algorytm II)

Dla danej sieci przepływowej stopnia drugiego przedstawionej na rys.4.9, wyznaczyć wartości przepływów w poszczególnych gałęziach, minimalizując funkcję celu wyrażającą straty energetyczne ponoszone w sieci



Rys.4.9. Struktura przestrzenna sieci przepływowej z przykładu 2 (test I)

Model matematyczny rozpatrywanego zagadnienia optymalizacji przyjmie postać:

$$f(x,y) = \sum_{i=1}^4 x_i \cdot y_i \longrightarrow \min \quad (4.6.5)$$

przy czym

$$\begin{aligned} x_1 + x_2 &= 5 \\ x_3 + x_4 &= 12 \\ q_1 + q_2 &= 17 \end{aligned} \quad (4.6.6)$$

$$y_1 - y_2 + y_3 - y_4 = 0 \quad (4.6.7)$$

$$\text{gdzie } y_i = d_i \cdot x_i^2 \cdot \text{sgn } x_i + h_i \quad i=1,2,3,4 \quad (4.6.8)$$

1. Dane wejściowe do obliczeń (krok 1)

$$N=4; W=4; Z=2; LI=100; K=2; EPS=10^{-4}$$

$$P(I,J) = \begin{bmatrix} 1 & 2 & 2 & 1 \\ 3 & 3 & 4 & 4 \\ 34 & 640 & 17 & 4 \end{bmatrix}$$

$$QW = [0,0,5,12]^T \quad H(I) = [2,3,5,2]^T$$

2. Macierz incydencji A (krok 2)

$$A = \begin{bmatrix} -1 & -4 \\ -2 & -3 \\ 1 & 2 \\ 3 & 4 \end{bmatrix} \quad S(J) = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \end{bmatrix}$$

oraz punkt startowy

$$\pi^0 = [0,6666|1|5|2]^T$$

3. Wartości pierwszej poprawki wynoszą, zgodnie z (4.3.8)

$$\delta^1 = [0|0|-89.1736|-33.0999]^T$$

a pierwsze przybliżenia rozwiązania

$$\pi^1 = [0,666|1|-84.1736|-31.0999]^T$$

4. Ponieważ nie osiągnięto żądanej dokładności ($\|\delta^1\| > EPS$), ponownie obliczamy kolejne poprawki i przybliżenia

$$\delta^2 = [0 \ 0 \ -266.8364 \ -118.7311]^T$$

$$\pi^2 = [0.666 \ 1 \ -351.0100 \ -149.8310]^T$$

$$\delta^3 = [0 \ 0 \ -185.2103 \ -95.4450]^T$$

$$\pi^3 = [0.666 \ 1 \ -536.2203 \ -245.2760]^T$$

$$\delta^4 = [0 \ 0 \ -24.1687 \ -14.8959]^T$$

$$\pi^4 = [0.666 \ 1 \ -560.3890 \ -260.1719]^T$$

$$\delta^5 = [0 \ 0 \ -0.2716 \ -0.2254]^T$$

$$\pi^5 = [0.6666 \ 1 \ -560.6606 \ -260.1949]^T$$

$$\delta^6 = [0 \ 0 \ -0.00002 \ -0.2029]^T$$

$$\pi^6 = [0.6666 \ 1 \ -560.6606 \ -260.3973]^T$$

$$\delta^7 = [0 \ 0 \ -0.00002 \ -0.00008]^T$$

$$\pi^7 = [0.6666 \ 1 \ -560.6606 \ -260.3973]^T = \hat{\pi}$$

5. Ponieważ osiągnięto wymaganą dokładność rozwiązania (tzn. $\|\delta\| \leq \text{EPS}$), obliczenia zostają przerwane i w oparciu o (1.2.25), (1.2.23) wyznaczane są wektory $\hat{x}, \hat{y}$ (krok 11)

$$\hat{x} = [4.0615 \ 0.9385 \ 3.9213 \ 8.0787]^T$$

$$\hat{y} = [-561.3272 \ -561.6606 \ -261.3973 \ -261.0639]^T$$

dla których funkcja celu (4.6.5) przyjmuje wartość

$$f(\hat{x}, \hat{y}) = \hat{x}^T \hat{y} = 5941,0229$$

5. WYNIKI BADAŃ PORÓWNAWCZYCH OPRACOWANEGO ALGORYTMU

Celem porównania proponowanej przez autora metody rozwiązywania zagadnień optymalizacji przepływów (1.3.4)-(1.3.7) oraz oceny efektywności działania opracowanego algorytmu, przeprowadzono serię badań porównawczych. Korzystając z wyników analizy zamieszczonych w rozdz. 2.3, przed przystąpieniem do badań, zredukowano wymiarowość rozpatrywanego zagadnienia poprzez sprowadzenie go do zadania transportowego z nieliniową funkcją kosztów. Spowodowało to wzrost konkurencyjności wybranych do porównania metod w stosunku do metody opracowanej przez autora. Dla potrzeb analizy wybrano trzy metody; Fletchera-Powella-Davidona, zmodyf. metodę Newtona oraz metodę Rosenbrocka. Otrzymane wyniki zamieszczono w rozdz. 5.3, gdzie przeprowadzono także ich analizę oraz ocenę efektywności obliczeń opracowanego algorytmu w świetle przeprowadzonych badań.

5.1. Sprowadzenie zadania optymalizacji do zagadnienia transportowego z nieliniową funkcją kosztów

Analiza literaturowa zamieszczona w rozdz. 2 wykazuje, że główną trudnością napotykaną w procesie rozwiązywania problemu (1.3.4)-(1.3.7) jest duża wymiarowość zagadnienia i znaczna liczba ograniczeń. Stąd przed przystąpieniem do badań porównawczych, problem optymalizacji nieliniowej z ograniczeniami (1.3.4)-(1.3.7) sprowadzony zostanie do zadania transportowego z nieliniową funkcją kosztów [34]. Ponadto, w oparciu o znaną w teorii grafów własność 1.1 (rozdz.1.2), zredukujemy wymiarowość zagadnienia, co spowoduje, że porównywane czasy obliczeń odpowiadać będą zadaniom zredukowanym o identycznej wymiarowości.

W rozdz. 3.2, poszukując rozwiązania problemu (1.3.4)-(1.3.7) wykazano nieaktywność ograniczeń (1.3.5), co umożliwiło sformułowanie warunków koniecznych i wystarczających rozwiązania optymalnego postaci (1.3.4)-(1.3.6), (3.2.36) i stworzyło podstawę do opracowania algorytmów I, II. Nieaktywność ograniczeń (1.3.5), sygnalizowana także w pracach [42,34] pozwala na sprowadzenie rozpatrywanego problemu, do równoważnego mu zagadnienia transportowego z nieliniową funkcją kosztów

$$f(x) = \sum_{i=1}^n [d_i \cdot |x_i|^k \cdot \text{sgn } x_i + h_i] x_i \rightarrow \min \quad (5.1.1)$$

z ograniczeniami

$$Ax - q = 0 \quad (5.1.2)$$

Zbiór ograniczeń (5.1.2) jest układem "w" równań liniowych, z których każde (w-1) równań jest liniowo-niezależnych, a występująca w ograniczeniach macierz A , to węzłowo-gałęziowa macierz incydencji odpowiedniego grafu G , reprezentującego strukturę połączeń sieci nieliniowej.

Korzystając z zamieszczonej w rozdz. 1.2, własności 1.1 macierzy incydencji A , mamy

$$\text{rank } A = w-1 \quad (5.1.3)$$

a zatem (Wn.1.1, rozdz.1.2, zredukowaną macierz incydencji A_Z przedstawić można w postaci

$$A_Z = [A_D \mid A_C] \quad (5.1.4)$$

gdzie

$$A_D = [a_{ij}]_{(w-1) \times (w-1)} \text{ oraz } \det A_D \neq 0 \quad (5.1.5)$$

Stąd, dokonując odpowiedniego rozbitcia wektora x na dwie grupy zmiennych

$$x = [x_D \mid x_C]^T \quad (5.1.6)$$

ograniczenie (5.1.2) przyjmie postać:

$$A_D x_D + A_C x_C - q = 0 \quad (5.1.7)$$

lub

$$x_C = A_C^{-1} q - A_C^{-1} A_D x_D \quad (5.1.8)$$

Posługując się ostatnią zależnością, n-wymiarowe zagadnienie (5.1.1)-(5.1.2) optymalizacji nieliniowej z ograniczeniami sprowadzone być może do (w-n+1)-wymiarowego zadania bez ograniczeń [34]. Jak nietrudno dostrzec, postępowanie takie przynosi znaczne korzyści prowadzące do zmniejszenia czasu obliczeń, bez względu na zastosowaną metodę rozwiązywania zadania bez ograniczeń. Z tego powodu uwzględnione ono zostało we wszystkich algorytmach wykorzystywanych w przeprowadzonych badaniach porównawczych, których omówieniu poświęcony będzie rozdział następny.

5.2. Rozwiązanie nieliniowego zadania transportowego metodami Rosenbrocka, Fletchera-Powella-Davidona i zmodyfikowana metoda Newtona

W literaturze znanych jest szereg metod i algorytmów rozwiązywania zagadnień optymalizacji nieliniowej, które posłużyć mogą do wyznaczenia rozwiązania, otrzymanego w wyniku transformacji zagadnienia (5.1.1)-(5.1.2), problemu optymalizacji nieliniowej bez ograniczeń. Spośród nich, dla potrzeb omawianej analizy wybrano trzy: metodę Fletchera-Powella-Davidona, zmodyfikowaną metodę Newtona oraz metodę Rosenbrocka, reprezentujące odmienne sposoby poszukiwań minimum funkcji nieliniowej. Wybór taki podyktowany został dążeniem jak najpełniejszego porównania opracowanej metody z innymi algorytmami i metodami już istniejącymi, jak również (w przypadku metody F-P-D) tym, że metoda przyjęta do porównania uważana jest w literaturze [15] za jedną z najlepszych.

Pozostałe dwie metody wybrano natomiast w oparciu o wyniki badań zamieszczone w pracy [34] i przeprowadzone testy obliczeniowe, dające podstawę do przypuszczeń, że pomimo iż powszechnie uważa się je za gorsze od pierwszej, to jednak specyfika rozpatrywanego zagadnienia powoduje, że ich zastosowanie również przynieść może zadowalające rezultaty.

Metodę poszukiwań prostych opracowaną przez Rosenbrocka oprogramowano w wersji przedstawionej w pracy [15]. Dla potrzeb realizacji obrotu współrzędnych bazowych wykorzystano procedurę ortogonalizacji Gramma-Schmidta [15]. Natomiast, celem uniknięcia niejednoznaczności przejścia od jednego układu bazowego do drugiego, w przypadku gdy suma kroków (s_k) w danym kierunku była równa zeru, podstawiano $s_k = 10^{-10}$.

Ponadto, przyjęto zaproponowane przez Rosenbrocka wartości parametrów; ekspansji (α) i kontrakcji (β) na poziomie $\alpha=3$, $\beta=0,5$ oraz dążąc do uniknięcia nadmiernej częstotliwości obrotu bazy, każdorazowo po wykonaniu obrotu pozostawiono ostatnie wartości długości kroku.

Warunkiem uruchomienia procedury obrotu układu bazowego było dwukrotne, kolejne otrzymanie wyników negatywnych we wszystkich kierunkach bazowych, zaś jako kryterium końca obliczeń przyjęto odległość rozwiązania bieżącego od znanego rozwiązania optymalnego.

Oprogramowując zmodyfikowaną metodę Newtona [15], należącą do grupy metod kierunków poprawy, celem wyznaczenia macierzy drugich pochodnych funkcji celu, wykorzystano wyniki zamieszczone w pracy [34]. Po określeniu nowego kierunku poszukiwań, minimalizację przeprowadzano w oparciu o bezgradientową metodę aproksymacji kwadratowej, przyjmując za kryterium stopu przekroczenie zadanej liczby obliczeń wartości funkcji celu. Za warunek zakończenia procesu obliczeń, podobnie jak w metodzie poprzedniej, przyjęto aktualną odległość otrzymanego rozwiązania od znanego punktu optymalnego.

Trzecia, ostatnia z metod wykorzystywanych w badaniach porównawczych, to metoda zmiennej metryki Fletchera-Powella-Davidona, w której celem wyznaczenia minimum w kierunku posłużono się ideą aproksymacji funkcji celu formą kwadratową z identycznymi jak w zmodyfikowanej metodzie Newtona kryteriami stopu. Szczegółowy opis programu znajduje się w pracy [54], gdzie również zamieszczono jego postać źródłową.

W oparciu o te trzy metody, oprogramowane w języku FORTRAN, przeprowadzono serię badań porównawczych polegających na rozwiązywaniu przykładowych problemów optymalizacji postaci (1.3.4)-(1.3.7) kolejno; opracowaną metodą (algorytm I) oraz metodami Fletchera-Powella-Davidona, Rosenbrocka i zmodyfikowaną metodą Newtona.

Prezentacji otrzymanych wyników poświęcony będzie rozdział następny, w którym także przeprowadzona zostanie ich analiza oraz ocena użyteczności opracowanego przez autora algorytmu.

5.3. Ocena opracowanego algorytmu optymalizacji w świetle przeprowadzonych badań porównawczych

Poniżej podane zostaną wyniki badań porównawczych opracowanego przez autora algorytmu I, w ramach których rozwiązano 12 testowych zagadnień optymalizacji postaci (1.3.4)-(1.3.7).

Dokonując wyboru problemów testowych, szczególną uwagę zwrócono na zróżnicowanie poszczególnych zadań zarówno pod względem stopnia nieliniowości zagadnienia, wielkości i topologii optymalizowanej sieci, jak również jej struktury przestrzennej. Stąd przyjęte, główne kryteria różnicujące przykłady to:

- stopień nieliniowości zagadnienia optymalizacji (K),

- ilość wierzchołków (W) i łuków (N) optymalizowanej sieci,
- liczba stabilności wewnętrznej grafu G,
- zakres wielkości parametrów d_i, h_i .

Przyjęcie dwóch pierwszych kryteriów wynika z faktu, iż efektywność obliczeń wybranych do porównania algorytmów w sposób istotny zależy od wymiarowości i stopnia nieliniowości rozwiązywanego zagadnienia. Natomiast wyboru trzech ostatnich kryteriów dokonano w oparciu o przedstawione w rozdz. 4.5 wyniki badań testowych algorytmu I, wskazujące na wyraźną zależność jego efektywności obliczeń od topologii optymalizowanej sieci i wielkości parametrów d_i .

Spośród sformułowanych 12 testowych zagadnień optymalizacji postaci (1.3.4)-(1.3.7) dwa, test V oraz test I przedstawione zostały w rozdz. 4.6 (Przykład 1 oraz Przykład 2). Poniżej opisane zostanie zadanie testowe VII, któremu w dalszej części rozdziału poświęcimy szczególną uwagę.

PRZYKŁAD 3 (test VII)

Dana jest nieliniowa sieć przepływowa stopnia drugiego, przedstawiona na rys.5.1. Wyznaczyć wartości przepływów w poszczególnych gałęziach w sposób zapewniający minimalizację funkcji celu określonej zależnością (5.3.1), wyrażającą straty ponoszone w sieci na dostarczenie wymaganej ilości przepływu od źródeł Z_1, Z_2 do odbiorców.

Model matematyczny rozpatrywanego zagadnienia optymalizacji przyjmie więc postać:

$$f(x, y) = \sum_{i=1}^{27} x_i y_i \longrightarrow \min \quad (5.3.1)$$

przy ograniczeniach

$$Ax \cdot q = 0 \quad (5.3.2)$$

$$By = 0 \quad (5.3.3)$$

$$y_i = d_i \cdot |x_i|^2 \operatorname{sgn}(x_i) + k_i \quad i=1, 2, \dots, 27 \quad (5.3.4)$$

gdzie:

$A [12 \times 27]$ - macierz incydencji grafu modelującego rozpatrywaną sieć,

$B [16 \times 27]$ - macierz obwodów podstawowych grafu G

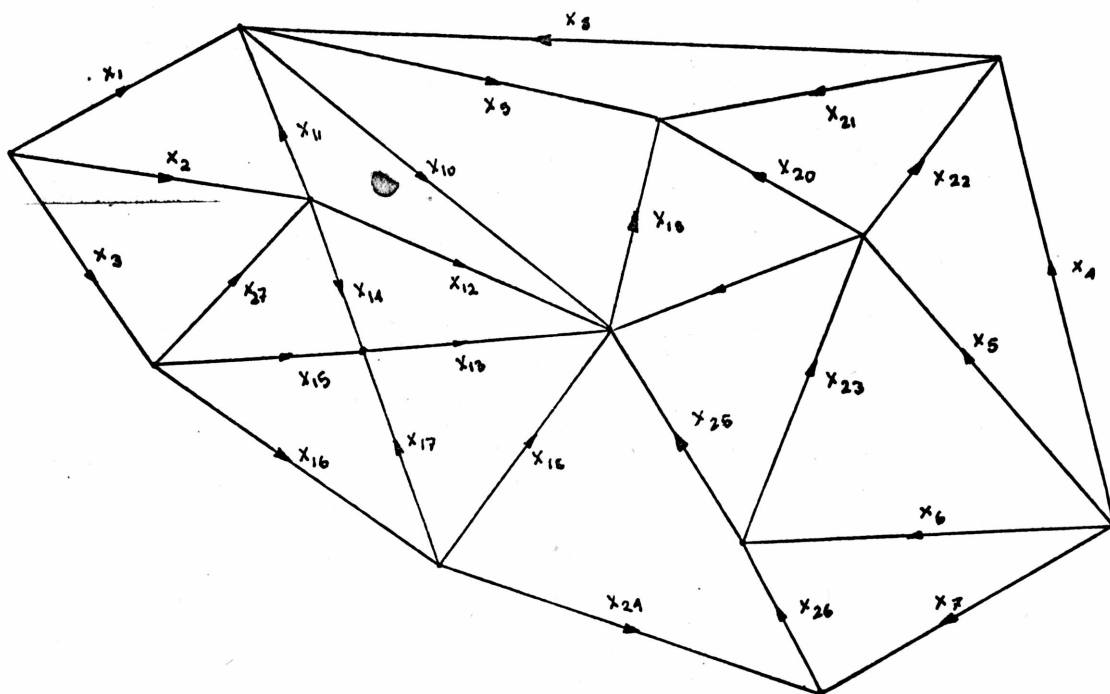
$$q^T = [0 \ 0 \ 15 \ 9 \ 5 \ 10 \ 3 \ 11 \ 6 \ 5 \ 3 \ 0] \quad (5.3.5)$$

Tabela 5.1. Wyniki badań porównawczych algorytmu I

Nr tes- tu	Ilość węzłów	Ilość złuków	Stopień nie- li- niowoś.	Algorytm I				Fletcher-Powell- Dawidon				Zmodyf. metoda Newtona				Rossenbrock			
				10^{-1}	10^{-3}	10^{-6}	PAO	10^{-1}	10^{-3}	10^{-6}	PAO	10^{-1}	10^{-3}	10^{-6}	PAO	10^{-1}	10^{-3}	10^{-6}	PAO
-	W	N	K	EPS1	EPS2	EPS3	XFAM	EPS1	EPS2	EPS3	XFAM	EPS1	EPS2	EPS3	XFAM	EPS1	EPS2	EPS3	XFAM
I	4	4	2	2	2	3	7,0	2	3	4	9,8	2	2	3	0,3	7	11	23	10,0
II	4	4	4	2	2	3	7,0	2	3	4	9,8	2	2	3	10,3	8	14	23	10,0
III	4	4	6	2	2	3	7,0	3	3	4	9,8	2	2	4	10,3	6	20	26	10,0
IV	4	4	8	2	2	3	7,0	2	3	5	9,8	2	2	3	10,3	9	19	26	10,0
V	5	8	2	2	2	4	7,2	7	18	12	11,7	3	5	6	11,8	12	30	71	11,5
VI	10	20	2	18	25	41	7,3	16	21	35	13,2	9	14	22	12,5	26	69	179	13,8
VII	12	27	2	15	29	35	7,4	33	48	69	14,9	28	43	64	14,7	67	109	213	14,1
VIII	21	40	2	27	42	54	8,5	78	85	163	16,3	46	73	148	15,1	-	-	-	-
IX	21	40	4	22	41	54	8,5	78	85	164	16,3	0,6	73	148	15,1	-	-	-	-
X	21	40	6	23	42	54	8,5	79	86	164	16,3	47	73	149	15,1	-	-	-	-
XI	21	40	8	23	42	54	8,5	79	88	166	16,3	46	74	148	15,1	-	-	-	-
XII	76	94	2	42	59	87	11,5	58	245	427	20,0	96	238	358	18,6	-	-	-	-

Uwaga:

Dla poszczególnych metod w pierwszych trzech kolumnach podano czasy trwania obliczeń (w sekundach) do momentu otrzymania rozwiązania z wymaganą dokładnością EPS1, EPS2, EPS3. Natomiast w kolumnie ostatniej zamieszczono wymaganą wielkość obszaru pamięci operacyjnej wyrażoną w K-słwach

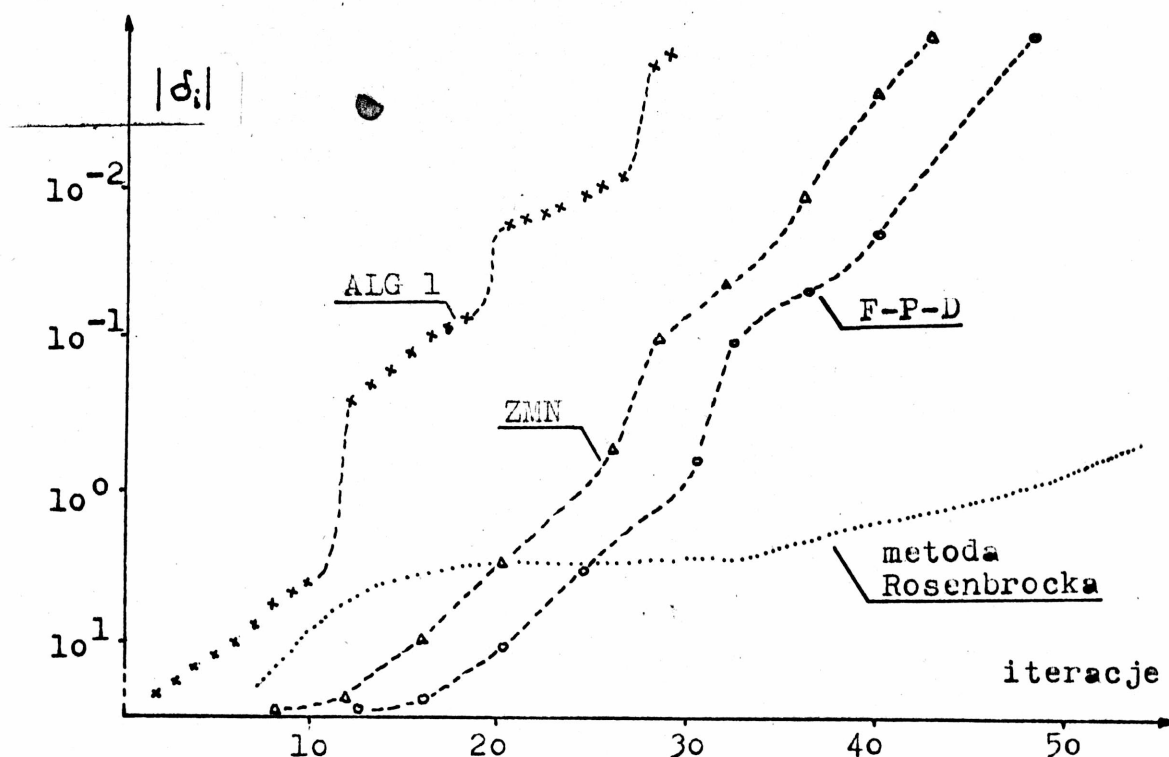


Rys.5.1. Struktura przestrzenna sieci przepływowej z testu VII

W tabeli 5.1 podano czasy trwania obliczeń zadań testowych rozwiązanych w oparciu o opracowany algorytm I oraz algorytmy wybrane do porównania, w funkcji wymaganej dokładności wyniku końcowego. W odróżnieniu od badań testowych przedstawionych w rozdz. 4.5, obliczenia dokonywano na maszynie ODRA-1325, a do pomiaru czasu wykorzystywano podprogram pomocniczy ITIME.

Dla wszystkich metod jako kryterium końca obliczeń przyjęto wielkość EPS, określającą stopień niespełnienia II prawa Kirchhoffa (1.3.5) w dowolnym obwodzie podstawowym rozpatrywanej sieci. Ustalono trzy poziomy dokładności rozwiązań odpowiadające wartościom EPS1, EPS2, EPS3, po osiągnięciu których przerywano obliczenia. Natomiast zamieszczone czasy trwania obliczeń, do momentu osiągnięcia wymaganej dokładności wyniku końcowego, podano w sekundach, przy czym w przypadku, gdy wartości te były większe od 300s obliczenia przerywano. Stąd np. brak czasu rozwiązywania testu VIII w oparciu o metodę Rossenbrocka, rozumieć należy jako nieosiągnięcie wymaganej dokładności ($\text{EPS3}=10^{-5}$) rozwiązania w czasie poniżej 5 min.

Rozwiązując zadanie testowe nr VII (Przykład 3) rejestrowano dodatkowo dokładności osiągnięte po każdej iteracji, a otrzymane wyniki przedstawiono na rys.5.2.



Rys.5.2. Wyniki obliczeń zadania testowego nr VII (tab. 5.1)

Kształt przedstawionych krzywych (rys.5.2), odpowiadających poszczególnym algorytmom, jest oczywiście zależny od postaci rozwiązywanego zagadnienia i nie może być reprezentatywny dla wszystkich przeprowadzonych testów obliczeniowych. Jednakże rys.5.2 uwidacznia pewną, jak się wydaje, pozytywną cechę opracowanego algorytmu I. Polega ona na wyznaczaniu rozwiązania optymalnego drogą wykonywania dużej liczby iteracji, przy czym czasy wykonania pojedynczej iteracji są bardzo krótkie, co powoduje, że również całkowity czas trwania obliczeń jest niewielki.

Odwrotnie jest dla pozostałych metod, gdzie ilość wykonywanych iteracji jest wyraźnie mniejsza (F-P-D, zmodyf. metoda Newtona) lecz ze względu na duże czasy trwania iteracji pojedynczej, sumaryczny czas obliczeń jest większy.

Ten odmienny od pozostałych sposób poszukiwania rozwiązania optymalnego za pomocą małych lecz szybkich kroków w kierunku minimum, posiada szczególne znaczenie z punktu widzenia dokład-

ności otrzymywanego rozwiązania w przypadku, gdy obliczenia zostają przerwane po upływie określonego czasu.

W świetle otrzymanych wyników (Tab.5.1) nasuwają się następujące wnioski:

- proponowany algorytm I, rozwiązywania zadań optymalizacji nieliniowej (1.3.4)-(1.3.7), jest lepszy od pozostałych, gdyż przy ustalonej dokładności wyniku końcowego, pozwala znaleźć rozwiązanie szybciej i przy mniejszej zajętości obszaru pamięci operacyjnej przez program obliczeniowy,

- w przypadku wykonywania dużej ilości obliczeń technicznych (tj. z niewielką dokładnością, rzędu 10^{-1} ÷ 10^{-2}), wykorzystanie opracowanego algorytmu I jest bardziej celowe przede wszystkim ze względu na krótsze czasy trwania obliczeń,

- proponowana metoda (algorytm) wydaje się być dobrym narzędziem do rozwiązywania zagadnień optymalizacji nieliniowej zarówno małej, jak również dużej (około 100-150 łuków) wymiarowości, które opisane są modelem (1.3.4)-(1.3.7).

6. PRZYKŁADOWE ZASTOSOWANIA PRAKTYCZNE OPRACOWANEJ METODY

Jednym z ważnych i istotnych elementów oceny jakości nowej metody jest ilość i znaczenie jej zastosowań użytkowych. Dlatego zostaną teraz przedstawione dwa przykłady, ilustrujące w jaki sposób otrzymane rezultaty posłużyć mogą do rozwiązywania zagadnień praktycznych. Pierwszy, zaprezentowany na III Konferencji Metod i Środków Projektowania Automatycznego - Warszawa 1981, dotyczy wykorzystania algorytmu I w projektowaniu i modernizacji sieci hydraulicznych, gazowych i wentylacyjnych. Drugi, przedstawiony na III Konferencji nt. Zastosowanie Komputerów w Przemysle - Szczecin 1981, ukazuje sposób w jaki otrzymane wyniki umożliwiają automatyzację procesów sterowania sieciami przepływowymi w czasie rzeczywistym.

6.1. Projektowanie i modernizacja sieci hydraulicznych, gazowych i wentylacyjnych

Niezmiernie istotnym i złożonym zagadnieniem charakteryzującym się znaczną liczbą zmiennych decyzyjnych i ograniczeń, jest problem projektowania nowych bądź modernizacji już istniejących sieci przepływowych typu sieci wodociągowych, gazowych czy wentylacyjnych.

Zadanie to w ogólnym przypadku przyjmuje postać zagadnienia 2 typu (1.3.4)-(1.3.6), (1.3.11), opisanego w rozdz. 1.3. Jednakże ze względu na duże trudności napotymane w procesie jego rozwiązywania, sprowadzane jest (najczęściej drogą przyjmowania dodatkowych założeń upraszczających) [35], [57] do znacznie prostszej postaci (1.3.4)-(1.3.6), (1.3.12). W praktyce oznacza to, nieuwzględnianie w fazie projektowania optymalnych warunków eksploatacji nowo-powstałych sieci, ograniczając się jedynie do doboru optymalnych struktur połączeń oraz przepustowości, ze względu na kryterium kosztów inwestycyjnych.

Uwaga powyższa dotyczy powszechnie stosowanych [35] w biurach projektowych pakietów programów optymalizacyjnych i pomocniczych typu systemu programów SIEĆ opracowanego w Centrum ETOB - "Etoprojekt", Warszawa czy pakietów programów PP-1÷PP-7 pochodzących z Wrocławskiego Biura Projektów Budownictwa Przemysłowego [32], [46].

Sytuacja taka, wymagająca od projektantów dodatkowej analizy warunków eksploatacji zaprojektowanych sieci, spowodowała znaczny wzrost zainteresowania budową analogowych i cyfrowych analizatorów pracy sieci nieliniowych. Przykładowymi urządzeniami tego typu mogą być: brytyjski - Water Network Distribution Anylyser, skonstruowany w British Aircraft Corporation Ltd. [58], czy amerykańskie analizatory rozptywu w sieciach, opisane w pracach [9], [12]. Również w Polsce analizatory takie skonstruowano m.in. w Zakładzie Pomiarów i Automatyki Centralnego Laboratorium Gazownictwa [44] oraz w Zakładzie Automatyki i Telemechaniki Politechniki Wrocławskiej na zlecenie Biura Projektów Gazownictwa - Gazoprojekt [4].

Ze względu na stosunkowo dużą ilość niezbędnych do analizy wariantów stanów pracy sieci oraz znaczne koszty zakupu analizatorów rozptywu, ten sposób rozwiązania zagadnienia 2 (rozdz. 1.3) nie znalazł jednak powszechnego zastosowania i aktualnie w większości prac projektowych, kryterium oceny jakości proponowanego rozwiązania oparte na minimalizacji kosztów strat energetycznych ponoszonych w czasie eksploatacji sieci, nie jest uwzględniane. W tej sytuacji, w świetle zwiększonej uwagi na oszczędności energetyczne, idea leżąca u podstaw opracowanej metody wydaje się być szczególnie przydatna. Umożliwia ona bowiem, poprzez modyfikacje istniejących programów obliczeniowych, wyznaczenie w fazie

projektowania przewidywanych, minimalnych kosztów eksploatacji rozpatrywanego rozwiązania.

Spośród stosowanych w pracach projektowych programów numerycznych, wyróżnić można grupę, której cechą wspólną jest wykorzystywanie, w celu wyznaczenia rozwiązania układu równań nieliniowych (1.3.9)-(1.3.11), metody Crossa-Łobaczewa. W algorytmach tych, dla przyjętych rozpiętości, określa się optymalne średnice połączeń węzłów sieci, a następnie w oparciu o metodę Crossa-Łobaczewa obliczane są rozpiętości rzeczywiste, odpowiadające wyznaczonym średnicom. Otrzymane rozpiętości stanowią podstawę do wyznaczania następnych przybliżeń optymalnych średnic połączeń itd. aż do chwili gdy w kolejnych iteracjach wartości średnic nie ulegną zmianie.

W tym przypadku, zastąpienie oryginalnej metody Crossa-Łobaczewa opracowanym przez autora algorytmem I, powoduje że programy te zyskując na szybkości obliczeń (w wyniku zastosowania procedury przyspieszającej), wyznaczają optymalną strukturę połączeń projektowanej sieci, nie tylko w sensie kosztów inwestycyjnych, lecz również w sensie przewidywanych kosztów strat energetycznych ponoszonych w trakcie późniejszej jej eksploatacji.

Zastosowanie algorytmu I umożliwia ponadto określenie wielkości efektów ekonomicznych, uzyskiwanych drogą wprowadzania w sieciach przepływowych układów regulacji automatycznej [35] oraz przeprowadzenie szeregu analiz, dotyczących kosztów i warunków eksploatacji rozpatrywanych sieci [1].

Jak nietrudno dostrzec, dysponując programami opartymi na metodzie Crossa-Łobaczewa, niezbędna ilość modyfikacji, związana z wykorzystaniem algorytmu I, sprowadza się do uzupełnienia programów istniejących procedurami: przyspieszającą i transformacji, co ze względu na ich przystosowanie do metody Crossa-Łobaczewa, nie przedstawia większych trudności.

W przypadku pozostałych programów, w których metoda Crossa-Łobaczewa nie jest wykorzystywana, wymagane modyfikacje polegają natomiast na odpowiednim (zależnie od specyfiki programu) uwzględnieniu w procesie obliczeń, warunków transformacji (3.3.20)-(3.3.22).

Na zakończenie podkreślić należy istotną zaletę proponowanej modyfikacji wynikającą z faktu, że algorytm I opiera się przede wszystkim na znanych, oprogramowanych [32],[35] powszechnie stosowanych w biurach projektowych metodach obliczeniowych, a jego

wprowadzenie do programów istniejących nie wymaga zasadniczych zmian ich struktury. Wydaje się więc, że wykorzystanie proponowanego algorytmu w procesie projektowania sieci przepływowych, pozabawione będzie szeregu niedogodności jakie są udziałem metod nowych, nieznanym projektantom.

6.2. Optymalizacja procesu sterowania siecią wodociagową w czasie rzeczywistym z wykorzystaniem minikomputera MERA-400

Obserwowany od kilku lat zwiększony nacisk na oszczędności energetyczne spowodował wzrost zainteresowania takimi obiektami przemysłowymi i komunalnymi, których wysoka energochłonność stanowi znaczne obciążenie bilansu energetycznego. Ze względu na dużą ilość i stosunkowo wysoką energochłonność, wśród obiektów tych znalazły się także, takie nieliniowe sieci przepływowe jak sieci dystrybucji wody w aglomeracjach miejsko-przemysłowych.

Prowadzone od szeregu lat, na zlecenie Instytutu Inżynierii Środowiska Politechniki Warszawskiej, w ramach programu rządowego PR-7, badania dotyczące metod i algorytmów automatycznego sterowania sieciami dystrybucji wody [47],[48],[49], w pełni potwierdziły zasadność poszukiwania takich metod sterowania, które uwzględniając wysoką energochłonność sieci wodociagowych, minimalizowałyby koszty strat energetycznych ponoszone w czasie ich eksploatacji.

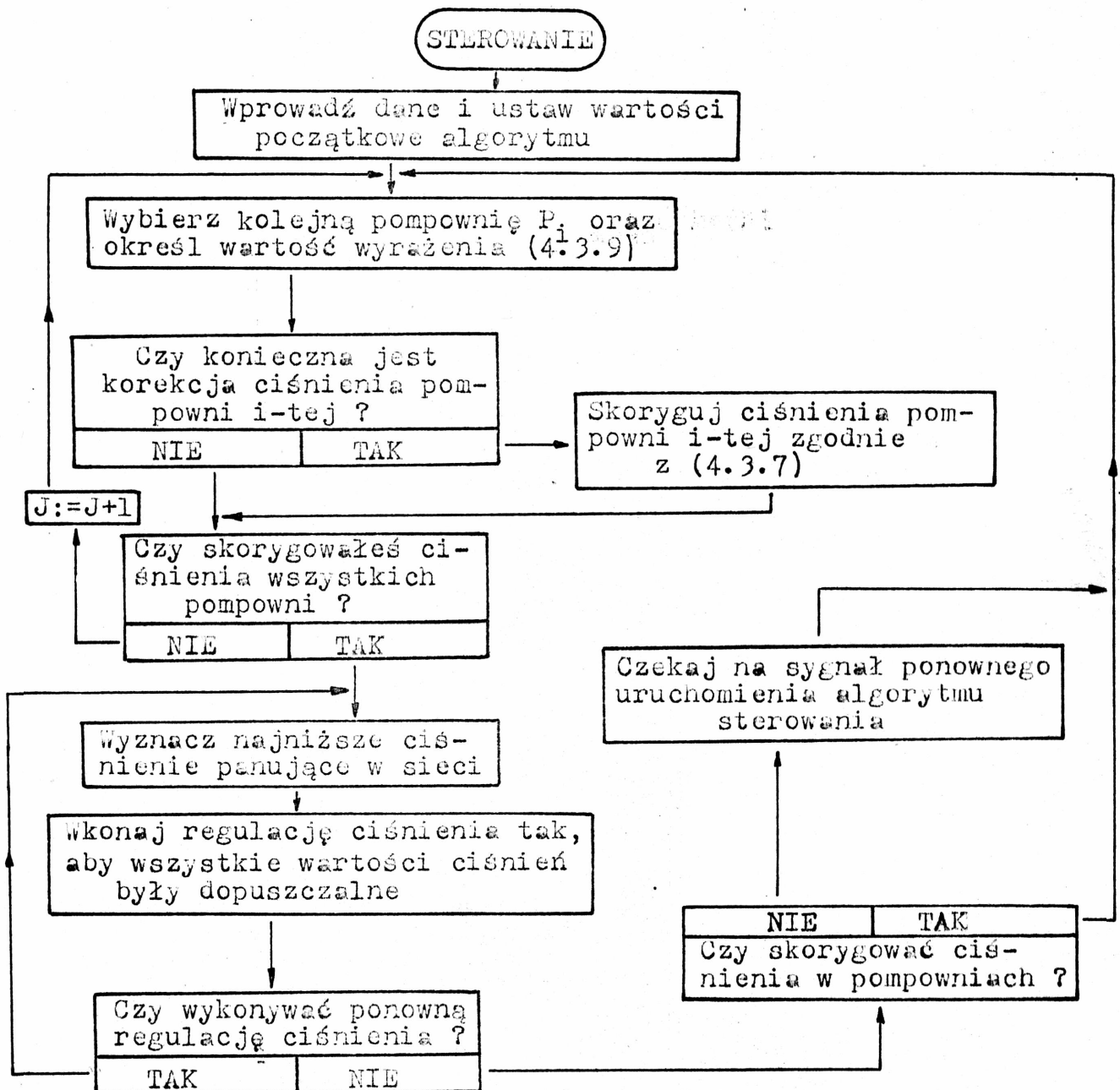
Analiza literatury i wyników badań zamieszczonych w pracy [49] wskazują, że szczególnie potrzebne są algorytmy i układy sterowania, których realizacja praktyczna nie wymagałaby dużych nakładów inwestycyjnych.

Zauważmy, że Twierdzenie 3.1 wyrażające podstawową własność rozwiązania optymalnego zagadnienia (1.3.4)-(1.3.7), zapewnia spełnienie tych wymagań. Sprowadza ono bowiem zagadnienie optymalnego sterowania siecią do problemu pomiaru i kontroli oraz regulacji ciśnień utrzymywanych we wszystkich pompowniach, w taki sposób aby spełniały one warunek (3.2.36).

Korzyści z tego faktu płynące polegają głównie na możliwości zrezygnowania z budowy modelu matematycznego sterowanej sieci i jego okresowej identyfikacji. Ponadto zależność (3.2.36) pozwala na wypracowanie decyzji sterujących bez konieczności rozwiązywania złożonych zadań optymalizacji statycznej. Oczywiście wszyst-

kie te fakty, jak również ograniczenie, niezbędnej do właściwego sterowania, ilości punktów pomiarowych do wartości minimalnej, znacznie obniżają koszty systemu sterowania i upraszczają sieć przesyłu informacji.

Proponowany algorytm, minimalno-energetycznego sterowania przepływami w sieci dystrybucji wody, przyjmie zatem postać algorytmu II (rozdz.4.3), przy czym kroki 2÷7 zastąpione zostają odczytem aktualnych wartości ciśnień $p_1, p_2, \dots, p_i$ oraz modyfikacji ulega kryterium stopu (rys.6.1).



Rys.6.1. Schemat blokowy algorytmu sterowania zespołami pompowymi zasilającymi sieć dystrybucji wody

W przedstawionym algorytmie (rys.6.1) na jeden pełny cykl regulacji, składają się dwa etapy:

- korekcji wzajemnych oddziaływań zespołów pompowych w poszczególnych pompowniach w celu zapewnienia optymalnych warunków współpracy układów pompowych,
- regulacji ciśnień zasilania w pompowniach w celu dostosowania rzeczywistych wydajności zespołów pompowych do aktualnego zapotrzebowania odbiorców.

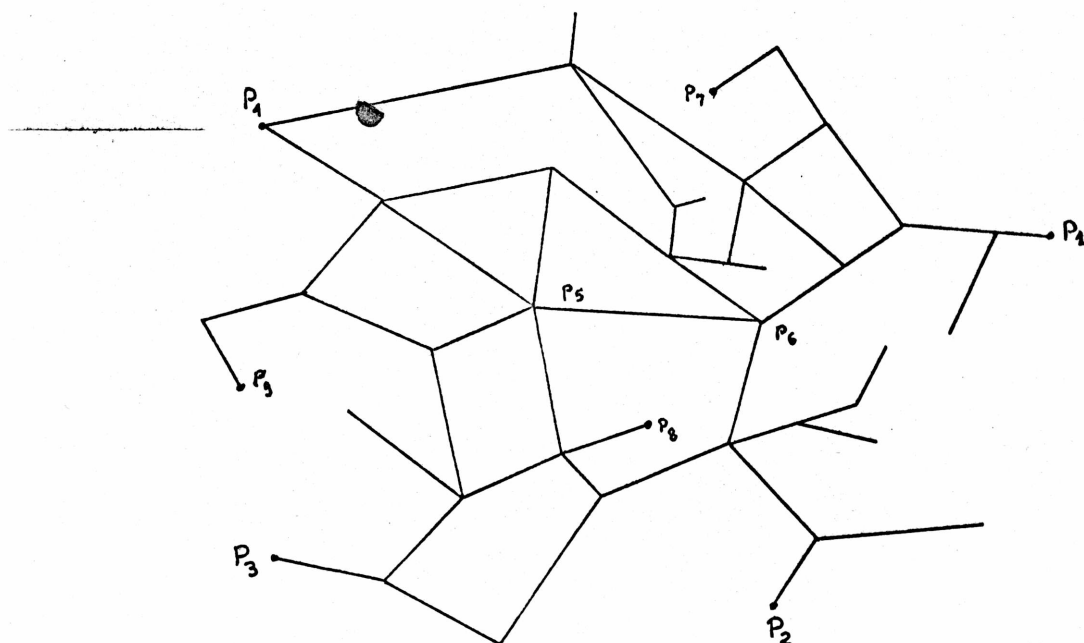
W czasie trwania etapu pierwszego, korekcji ulegają wartości ciśnień zasilania tak, aby spełniały one warunek (3.2.36). Etap drugi polega natomiast na takim doborze minimalnych ciśnień zasilania, dla których wartości ciśnień panujących w sieci znajdują się w wymaganym przedziale p_{min}, p_{max} . W przypadku, gdy warunek ten nie może zostać spełniony, ciśnienia zasilania dobierane są tak, aby żadne z ciśnień panujących w sieci nie przekroczyło górnej granicy przedziału dopuszczalnego.

Jak nietrudno dostrzec, specyfika metody sprawia, że częstość wykonywania procedury korekcji może być (w stosunku do procedury regulacji) znacznie mniejsza. Tak więc, na proces sterowania składa się okresowe wykonywanie pełnego algorytmu i kilku cykli skróconych (tylko procedura regulacji). Oczywiście, ilość cykli skróconych przypadających na jeden cykl pełny, jak również częstotliwość wykonywania cykli pełnych, zależą w sposób istotny od dynamiki zmian zachodzących w sieci i powinny być określane na podstawie wyników badań przeprowadzonych bezpośrednio na obiekcie.

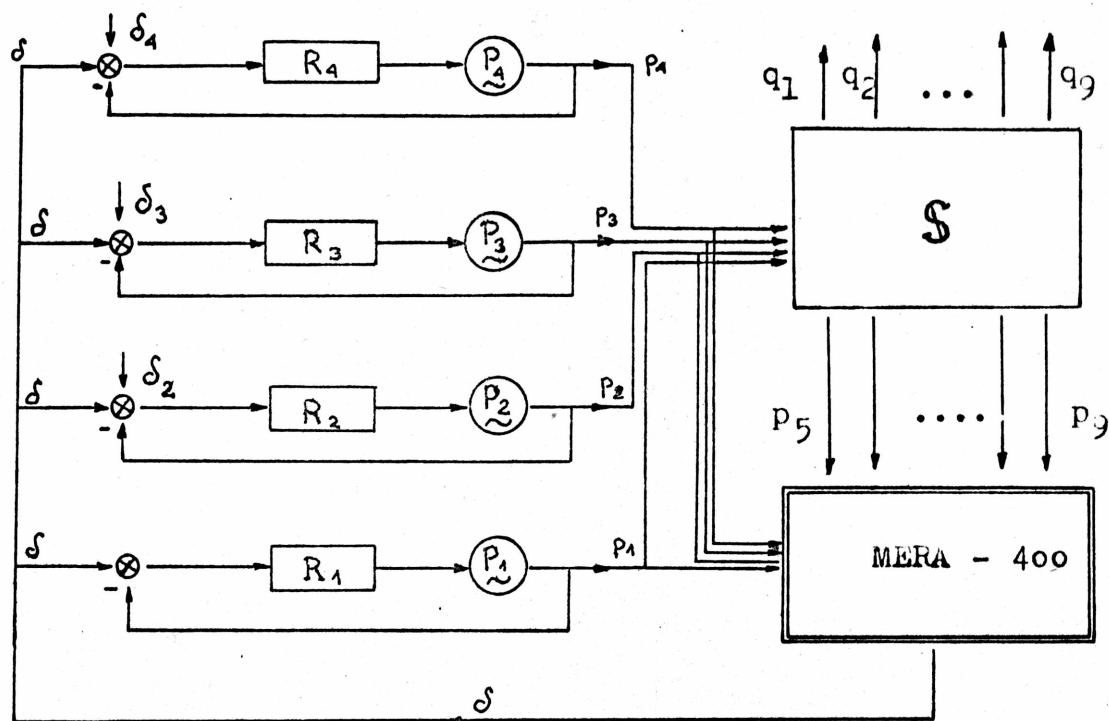
Mając na uwadze wszechstronne przedstawienie idei leżącej u podstaw proponowanego układu regulacji, rozpatrzmy sieć dystrybucji wody w aglomeracji miejsko-przemysłowej, przedstawioną na rys.6.2.

Pompownie P_2, P_3, P_4 , połączone z pompownią centralną P_1 siecią teletransmisji, wyposażone są w regulatory zapewniające utrzymanie właściwej (zgodnej z (3.2.36)) różnicy ciśnień σ_j w stosunku do pompowni centralnej. W dyspozytorni centralnej, wyposażonej w minikomputer MERA-400, po otrzymaniu za pośrednictwem kanału automatyki przemysłowej INTEL DIGIT-PI, informacji o wartościach ciśnień $p_5, p_6, \dots, p_{10}$ panujących w miarodajnych punktach sieci, wypracowywana jest poprawka σ . Następnie przekazywana ona jest za pośrednictwem sprzętu automatyki kompleksowej

INTELDIGIT-PI do pozostałych pompowni (rys.6.3), gdzie podawana jest na wejścia regulatorów ustalających punkty pracy poszczególnych zespołów pompowych.



Rys.6.2. Struktura połączeń rozpatrywanej sieci dystrybucji wody



Rys.6.3. Schemat funkcjonalny systemu automatycznego sterowania siecią dystrybucji wody

Przedstawiony algorytm sterowania został oprogramowany i przetestowany na minikomputerze MERA-400 w języku FORTRAN IV-E. Otrzymane wyniki pozwalają przypuszczać, że może on być użyteczny na każdym etapie wprowadzania automatyzacji sterowania [41], zarówno tam gdzie pracą sieci kieruje bezpośrednio dyspozytor, jak również w przypadku gdy zadanie to powierzone jest układowi regulacji automatycznej czy maszynie cyfrowej.

W tym ostatnim przypadku, ze względu na dużą szybkość wykonywania operacji, jak również krótki czas trwania pełnego cyklu sterowania opisanego algorytmu, decyzje sterujące podejmowane są okresowo, umożliwiając współbieżne wykonywanie dodatkowych analiz i obliczeń nie związanych bezpośrednio z procesem sterowania.

7. UWAGI I WNIOSKI KOŃCOWE

W pracy określono zasady rozwiązywania zadań statycznej optymalizacji nieliniowej w sieciach przepływowych postaci (1.2.17), (1.2.21)-(1.2.25). Wykazano, że do rozwiązywania zagadnień optymalizacji nieliniowej typu (1.3.4)-(1.3.7) wykorzystać można ideę leżącą u podstaw znanej z literatury metody prądów składowych i wyrównawczych, co znacznie poprawia efektywność stosowanych algorytmów oraz umożliwia opracowanie algorytmów nowych, o wyraźnie skróconym czasie trwania obliczeń i zredukowanej zajętości pamięci operacyjnej maszyn cyfrowych.

Problem (1.3.4)-(1.3.7) to n -wymiarowe, statyczne zagadnienie optymalizacji nieliniowej o dużej liczbie ograniczeń, na które składają się trzy grupy równań. Dwie pierwsze to " n " ograniczeń liniowych na zmienne poprzeczne x -(1.3.4) oraz zmienne podłużne y (1.3.5), trzecia natomiast to równania nieliniowe-(1.3.6) określające powiązania pomiędzy odpowiednimi składowymi wektorów x i y .

Dla powyższego problemu, w rozdziale trzecim, określono warunki konieczne i wystarczające optymalności rozwiązania, umożliwiające transformację zadania wyjściowego (1.3.4)-(1.3.7) do układu nieliniowych równań algebraicznych (3.3.15)-(3.3.17). Następnie wykazano możliwość redukcji otrzymanego układu, tak iż w efekcie końcowym otrzymano do rozwiązania układ $(n-w+z)$ równań nieliniowych postaci (3.3.36).

Uzyskana w ten sposób znaczna redukcja wymiarowości rozpatrywanego zagadnienia, w połączeniu z odpowiednim doбором metody rozwiązywania układu (3.3.36), pozwoliła na opracowanie szybkiego i efektywnego algorytmu (I) optymalizacji, opartego na znanej w literaturze metodzie Crossa-Łobaczewa. Bardzo mała zajętość pamięci operacyjnej maszyny cyfrowej, jaką charakteryzuje się algorytm I, wynika z wykorzystania w nim tej niezwykle prostej metody iteracyjnej. Jednakże konsekwencją takiego wyboru była słaba zbieżność opracowanego algorytmu (I), stąd dalsze uprawnienia związane były z przyspieszeniem tej zbieżności.

W tym celu, wykorzystując liniową zbieżność metody Crossa-Łobaczewa, opracowano procedurę przyspieszającą (3), zbliżoną do procesu δ^2 -Aitkena oraz określono warunki jej stosowalności. Wykazano również celowość wykorzystania opracowanej procedury dla przyspieszenia zbieżności metody Crossa-Łobaczewa.

Wyniki przeprowadzonych badań testowych zamieszczone w rozdziale czwartym, wykazują, że główną zaletą algorytmu I jest duża efektywność obliczeń połączona ze stosunkowo małą zajętością pamięci operacyjnej oraz niewielka czułość na dobór punktu startowego. Natomiast negatywna własność metody Crossa-Łobaczewa, jaką jest zależność czasu trwania obliczeń od wyboru macierzy B_D , została osłabiona poprzez zastosowanie procedury przyspieszającej.

W celu lepszej oceny własności algorytmu I, dokonano jego porównania z algorytmami opartymi na metodach: Rosenbrocka, Fletchera-Powella-Davidona i zmodyfikowanej metodzie Newtona. Przeprowadzone serie badań porównawczych, których wyniki przedstawiono w rozdziale piątym, wykazały wyższość opracowanego algorytmu w stosunku do dwóch pierwszych (Rosenbrock, F-P-D). Charakteryzuje się on znacznie krótszymi czasami obliczeń, jakie są niezbędne dla uzyskania rozwiązania z wymaganą dokładnością. Dla wszystkich zadań testowych obserwowano 3-5 krotne skrócenie czasu obliczeń przy średnio 1,5-krotnie mniejszej zajętości pamięci.

W przypadku algorytmu opartego na zmodyfikowanej metodzie Newtona, osiągnięte korzyści są mniejsze (skrócenie czasu obliczeń rzędu 1,5-2), jednakże również wskazują na konkurencyjność opracowanego algorytmu I.

Szczególną uwagę poświęcono w pracy problemowi zastosowań praktycznych opracowanej metody rozwiązywania zagadnień (1.3.4)-(1.3.7). Stąd m.in. wykorzystanie w algorytmie I, znanej i opro-

gramowanej metody Crossa-Lobaczewa, powszechnie używanej przy projektowaniu i modernizacji sieci przepływowych. Wybór taki stworzył warunki umożliwiające wykorzystanie opracowanej metody w biurach projektowych, po wprowadzeniu niewielkich zmian w istniejących programach, które jednak nie zmieniają zasadniczej struktury ich obecnej budowy.

Zastosowania praktyczne opracowanej metody nie sprowadzają się jednak, tylko do prac projektowych. Przedstawiony w rozdziale czwartym algorytm II umożliwia optymalizację (w sensie minimalizacji strat energetycznych) procesu sterowania sieciami przepływowymi w czasie rzeczywistym. Koncepcja budowy takiego systemu sterowania zaprezentowana została na III Krajowej Konferencji - Zastosowanie komputerów w Przemysle, a wyniki tej prezentacji oraz dyskusja nad proponowaną koncepcją wydają się potwierdzać jej znaczenie praktyczne.

Określone w pracy zasady optymalizacji nieliniowych sieci przepływowych ograniczone zostały do zadań postaci (1.3.4)-(1.3.7), w których zależność (1.3.6) jest funkcją potęgową. Otrzymane rezultaty sugerują celowość rozszerzenia badań teoretycznych na zagadnienia optymalizacji w nieliniowych sieciach transportowych oraz na przypadek, gdy zależność (1.3.6) jest wielomianem, a w aspekcie zastosowań praktycznych - na komputerowe wspomaganie optymalizacji sterowania i zarządzania pracą omawianych systemów.

Na zakończenie autor pragnie wyrazić serdeczne podziękowanie Panu Doc. dr inż. Tadeuszowi Stanickiemu za twórczą inspirację oraz troskliwą i wszechstronną opiekę w trakcie realizacji rozprawy. Równie gorąco pragnie podziękować Dr inż. Jerzemu Kotowskiemu i Dr inż. Markowi Olesiakowi oraz Koleżance i Kolegom z Zespołu Badawczego, za cenne uwagi, które pomogły nadać ostateczny kształt pracy.

LITERATURA

- [1] Auriaux G.: Elaboration automatique de consignes pour la conduite d'un reseau de distribution d'eau: T.S.M.-L'eau, No.1, 1980.
- [2] Benders J.F.: Partitioning Procedures for solving Mixed-Variables Programming Problems, Numerische Mathematik 4, str.238-252, 1962.
- [3] Berge C.; Ghoulia-Houri A.: Programming, Games and Transportation Networks, London 1965.
- [4] Błony Z.: Analogia elektryczna w zastosowaniu do hydraulicznego obliczania sieci gazociągowym, Gaz, Woda, Technika Sanitarna, Nr 1, 1966.
- [5] Brooks R., Smith C., Stone A.: The Dissection of Rectangles into Squares, Duke Math. Journal, Vol. 7, p.312-340, 1940.
- [6] Busacker R., Saaty T.: Finite Graphs and Networks, Mc Graw Hill, London 1973.
- [7] Cao C., Sulla convergenza del metodo di Cross; Pubblicazioni Dell'Istituto di Idraulica e Construzioni Idrauliche, Nr 13, Universita' Degli Studi di Cagliari 1963.
- [8] Cea J.: Optymalizacja, teoria i algorytmy, PWN, Warszawa 1976.
- [9] Claude C., Lomax Jr.; Network flow distribution using the McJlroy Analyser or IBM-650 Digital Computer, Washington State Institute of Technology Bulletin, No.247.
- [10] Cross H.: Analysis of Flow in Networks of Conduits or Conductors, Bulletin University of Jllinois Engng. Exp. Station No.286, 1936.
- [11] Deo N.: Teoria grafów i jej zastosowania w technice i informatyce, PWN, Warszawa 1980.
- [12] Dobson W., Williams R.: Simulation of Distribution Networks by General Purpose Analogue Computer, Journal of the Institute of Water Engineering.
- [13] Elias P., Feinstein A., Shannon C.: Note on the Maximal Flow Through a Network, IRE Trans. Inform. Theory, Vol.IT-2, p.117-119, Dec. 1956.
- [14] Fiacco A., Mc Cormick G.: Nonlinear Programming, Sequential Unconstrained Minimization Techniques, Wiley, New York 1968.
- [15] Findeisen W., Szymanowski J., Wierzbiński A.: Teoria i metody obliczeniowe optymalizacji, PWN, Warszawa 1977.

- [16] Fletcher R., Powell M.: A Rapidly Convergent Descent Method for Minimization, The Computer Journal, Vol.6, p.163, 1963.
- [17] Fletcher R., Reeves C.: Function Minimization by Conjugate Gradients, The Computer Journal, Vol.7, p.149, 1964.
- [18] Ford L., Fulkerson D.: Przepływy w sieciach, PWN, Warszawa 1969.
- [19] Gawrych-Żukowski A., Kotowski J., Olesiak M.: Algorytm wyznaczania minimalno-energetycznego sterowania złożoną siecią rozprowadzania wody, raport ICT nr 106/80, Wrocław 1980.
- [20] Geoffrion A.: Generalized Benders Decomposition, Journal of Optimization Theory and Application, Vol.10, p.237-260, 1972.
- [21] Haarhoff P., Buys J.: A New Method for the Optimization of a Nonlinear Function Subject to Nonlinear Constraints, Computing Journal, Vol.13, 1970.
- [22] Harris B.: Graph Theory and Its Applications, Academic Press, New York, London 1970.
- [23] Hestenes M.: Multiplex and Gradient Methods, JOTA, Vol.4, 1969.
- [24] Jewdokimow A.: Optimalnyje zadaczi na inżyniernych sietiach, U.S.Charkow 1976.
- [25] Johnson D., Kenstra J., Rinnooy Kan A.: The Complexity of the Network Design Problem, Networks, Vol.8, 1978.
- [26] Karlin S.: Mathematical Methods and Theory in Games, Programming and Economics, Addison-Wesley, Massachusetts 1959.
- [27] Kirchhoff G.: Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung Galvanischer Strome geführt wird, Ann. der Phys. und Chemie, Vol.72, p.497-508, 1897.
- [28] Kirchhoff G.: Vorlesungen über mathematische Physik, Bd.3, Electricitat und Magnetismus, Leipzig 1891.
- [29] Klimaszewski T., Pejas J.: Komputerowo wspomagane metody projektowania sieci wodociągowych, ICT Politechniki Wrocławskiej, 1978.
- [30] Kochenburger R.: Modelowanie układów dynamicznych przy użyciu maszyn matematycznych, Warszawa 1975.
- [31] Korzan B.: Elementy teorii grafów i sieci. Metody i zastosowania, WNT, Warszawa 1978.
- [32] Kusterski W. i inni: Obliczanie rozpyłów i strat ciśnienia w sieciach wodociągowych wielostronnie zasilanych, Wrocł. Biuro Projektów Budownictwa Przemysłowego, Wrocław 1969.

- [33] Kotowski J., Olesiak M.: The Optimization of the Energy Wastes in the Complex Water-Supply Systems, Proc 6-th IFAC/IFIP Int. Conf. of Digital Comp. Appl. to Process Control, Dusseldorf, 10-17 Oct. 1980.
- [34] Kotowski J., Olesiak M.: Gradientowe metody optymalizacji parametrów sieci wodociągowej, Zeszyty Nauk. Polit. Śląskiej nr 60, Gliwice 1981.
- [35] Krzyszczuk H.: Optymalizacja rozdziału czynnika na odcinki sieci hydraulicznej, pierścieniowej. System programów SIEĆ, Gosp. Wodno-Ściekowe, Nr 3, Warszawa 1969.
- [36] Łobaczew W.: Prijemny raszczeta wodoprowodnych sietiej, RFSR, Moskva 1950.
- [37] March L., Stedman P.: The Geometry of the Environment, PIBA Publications, London 1971.
- [38] Maxwell J.: Treatise on Electricity and Magnetism, Unabridged republication of the 3-rd edition, published by the Clarendon Press in 1891, Dover Publ, New York 1954.
- [39] Mesarovic M., Takahara Y.: Obscaja teorija sistem, matielmaticheskoje osnovy, Mir, Moskva 1978.
- [40] Nikodem J.: Model matematyczny rozprowadzania wody w sieciach przepływowych w warunkach, gdy podaż przewyższa popyt, Materiały II Międzynarodowego Symp. Nauk. - Podstawy Projek. Systemów Zaopatrzenia w Wodę Aglomeracji Miejsko-Przemysł., Karpacz 1980.
- [41] Nikodem J.: Algorytm minimalno-energetycznej optymalizacji nieliniowych sieci przepływowych, Materiały III Konf. Metod i Środków Proj. Automatycznego, Warszawa 1981.
- [42] Nikodem J.: Algorytm minimalno-energetycznego sterowania siecią dystrybucji wody w czasie rzeczywistym z wykorzystaniem minikomputera MERA-400, mat. III Konf. - Zastosowania Komputerów w Przemysle, Szczecin 1981.
- [43] Peterson B.: A Cut-Flow Procedure for Transportation Network Optimization, Networks, Vol.10, No.1, p.33-43, 1980.
- [44] Piekarski M.: Zastosowanie maszyn analogowych do modelowania układów regulacji Wydajności tłoczni gazu, GWTS, Nr 1, 1966.
- [45] Pleśnik J.: The Complex of Designing a Network with Minimum Diameter, Networks, Vol.11, p.77-85, 1981.
- [46] Praca zbiorowa, Program PP-7, obliczania sieci wodociągowych dla maszyn serii ODRA, Prace Wrocławsk. Biura Projektów Budow. Przemysłowego, Wrocław 1969.

- [47] Praca zbiorowa, Opracowanie modelu układu rozprowadzania wody w aglomeracji miejsko-przemysłowej do celów komputerowo wspomaganego sterowania systemem, raport ICT nr 52/80, 1980.
- [48] Praca zbiorowa: Opracowanie podstawowych założeń: metod i algorytmów dla potrzeb operatywnego sterowania złożonym systemem wodociągowym, praca wykonana w ramach PR-7(01.09.01), 1981.
- [49] Praca zbiorowa: Opracowanie metod automatycznego sterowania siecią wodociągową z uwzględnieniem ich technicznej realizacji, praca wykonana w ramach PR-7(01.09.06), Wrocław 1981.
- [50] Praeger W.: Problems of Traffic and Transportation, Proc. of Symposium on Operations Research in Business and Industry, Kansas City Midwest Research Institute, Kansas City 1954.
- [51] Ralston A.: Wstęp do analizy numerycznej, PWN, Warszawa 1971.
- [52] Rasmussen H.: Simplified Optimization of Water Supply Systems, Proc. ASCE, Journal of the Environmental Engineering Division, No 2, pages 313-327, 1976.
- [53] Schuldt S.: A Method of Multipliers for Mathematical Programming Problems with Equality and Inequality Constraints, JOTA, Vol.17, 1975.
- [54] Stanicki T. i inni: System OPT-1 do wspomagania nauczania w zakresie metod optymalizacji deterministycznej, Cz.II, Dokumentacja Programowa, raport ICT, nr 264, Politechnika Wrocławska 1978.
- [55] Treumper K.: Optimal Flows in Nonlinear Gain Networks, Networks, Vol.8, p.17-36, 1978.
- [56] Voyles C., Wilke H.: Selection of Circuit Arrangements for Distribution Network Analysis by the Hardy Cross Method-AWWA Journal, Marzzo 1979.
- [57] Watanada T.: Least - cost Design of Water Distribution Systems, Proc. ASCE, Journal of the Hydraulics DIVISION, No. 10, p.1997-1513, 1979.
- [58] Williams R.: A New Water Network Distribution Analyser, Water and Water Engng., Vol.68, No.815, January 1964.
- [59] Youngs J.: Kirchhoff's Law and Map Colouring Problems in Mathematical Aspects of Electrical Network Analysis, American Math. Society Providence, RI, 1971.
- [60] Lorens C.: Flowgraphs for the Modeling and Analysis of Linear Systems, Mc.Graw-Hill, New York 1964.

- [61] Mason S.: Feedback Theory, Further Properties of Signal Flow Graphs, Proc. IRE, No.7, p.920-926, 1956.
- [62] Watson D.: Network Methods in Models of Production: Networks, Vol.10, No.1, 1980.
- [63] Strassen V.: Gaussian Elimination is Not Optimal, Numerische Math. Band 3, Heft A, Springer-Verlag, Berlin 1969.

Mgr inż. Jan Nikodem

Raport złożono w Redakcji
Instytutu 21.06.1982 r.

Niniejszy raport otrzymują:

1. OINT - Biblioteka Międzyinstytutowa I-6	1 egz.
2. Biblioteka Główna Politechniki Wrocł.	1 egz.
3. Z-ca Dyrektora I-6 ds. Kształcenia Kadry Naukowej	1 egz.
4. Promotor	1 egz.
5. Recenzenci	2 egz.
6. Autor	4 egz.

Razem : 10 egz.

OPTIMIZATION ALGORITHMS OF ENERGY WASTES IN NONLINEAR FLOW NETWORKS

In this dissertation a new method of the optimization of energy wastes in the nonlinear complex networks flow was presented.

For this aim the multidimensional nonlinear statical problem with constraints was described. The optimization criterion was the total energy wastes in network and the considered problem belongs to the main problem named as optimization in nonlinear Kirchhoff's network /two laws stated by Kirchhoff were constraints/.

The necessary and sufficient conditions for existence of optimal solution were formulated based on Khun-Tucker theory. To reduced obtained conditions firstly, a well-know equalizing and component current method were generalized. Secondly, some properties of the Kirchhoff's network which gave the possibility to reduce number of equations were used.

Moreover there were given most efficient algorithms to find a solution of finally obtained system of equations. This algorithm was constructed basis on modified Henry Cross method aided by special acceleration procedure which can be compared to very well-know in literature δ^2 -Aitken's procedure.

A generalization of equalizing and component current method gave second simple algorithm for determined a solution of considered problem. This algorithm was very similar to first but it may be applied in on line automatical control process.

Finally, to demonstrated the manner in which described algorithms can be applied in the practical case, considered two examples.

9. DODATEK

9.1. Postać źródłowa programu ALG1 - opracowanego w oparciu o algorytm I

SOURCE UPDATE
LISTING

09. 06. 1982

STRONA 1

```

1      1      PROGRAM ALG1
2      2      REAL X(100),DE(4,50),R(100),H(100)
3      3      INTEGER Z,SK,A(30,20),B(30,20),P(100,3),G(30,20)
4      4      INTEGER SP(30),S(30),NK(30),KP(6),KK(6),LP(3),LK(3)
5      5      INTEGER D(200)
6      6      READ(1,0)N,W,Z,LI,L1,L2,EPS,KA
7      7      READ(1,0)((P(J,I),I=1,3),J=,N)
8      8      READ(1,0)(Q(I),I=1,W)
9      9      READ(1,0)(H(I),I=,N)
10     10     K=1
11     11     DO 100 J=1,W
12     12     DO 8 IP=1,2
13     13     SK=2*IP-3
14     14     DO 8 I=1,N
15     15     IF(P(I,IP).NE.J) GO TO 8
16     16     A(J,K)=I*SK
17     17     K=K+1
18     18     8 CONTINUE
19     19     S(J)=K-1
20     20     100 K=1
21     21     DO 183 I=1,Z
22     22     183 NK(I)=S(I)
23     23     IU=W-1
24     24     DO 601 I=1,IU
25     25     DO 602 K=,W
26     26     IF(K.EQ.1) GO TO 602
27     27     IP=S(K)
28     28     DO 602 J=1,IP
29     29     IF(IABS(P(K,J)).EQ.IABS(P(I,))) GO TO 604
30     30     602 CONTINUE
31     31     GO TO 601
32     32     604 IX=-P(K,J)
33     33     P(K,J)=0
34     34     IP=S(I)
35     35     IK=S(K)
36     36     DO 605 L=2,IP
37     37     DO 606 LU=1,IK
38     38     IF(IABS(P(I,L)).NE.IABS(P(K,LU))) GO TO 606
39     39     P(K,LU)=0
40     40     GO TO 605
41     41     606 CONTINUE
42     42     P(K,LU)=X/P(I,1)
43     43     P(K,LU)=P(K,LU)*P(I,L)
44     44     S(K)=S(K)+1
45     45     605 CONTINUE
46     46     DO 607 IMM=1,2
47     47     IP=S(K)
48     48     DO 607 LU=1,IP
49     49     IF(P(K,LU)NE.0) GO TO 607
50     50     IK=S(K)-1
51     51     DO 608 IZ=U,IK
52     52     IN=IZ+1
53     53     608 P(K,IZ)=P(K,IN)

```

SOURCE UPDATE
LISTING

09. 06. 1982

STRONA 2

```

54 54 S(K)=(K)-1
55 55 607 CONTINUE
56 56 GO TO 602
57 57 601 CONTINUE
58 58 IP=W-1
59 59 DO 65 I=1,IP
60 60 IF(P(I,1).LT.0) GO TO 609
61 61 LU=S(I)
62 62 DO 611 K=1,LU
63 63 611 P(I,K)=-P(I,K)
64 64 609 CONTINUE
65 65 NUO=IABS(P(I,1))
66 66 615 R(NUO,1)=111
67 67 L=1
68 68 K=1
69 69 DO 612 J=1,N
70 70 IF(R(J,1).EQ.111) GO TO 612
71 71 R(L,K)=J
72 72 K=K+1
73 73 IP=W-1
74 74 DO 613 M=1,IP
75 75 IK=S(M)
76 76 DO 613 MM=2,IK
77 77 IF(IABS(P(M,MM)).EQ.J) GO TO 64
78 78 613 CONTINUE
79 79 SR(L)=K-1
80 80 L=L+1
81 81 K=1
82 82 612 CONTINUE
83 83 GO TO 510
84 84 614 R(L,K)=-P(M,MM)*P(M,1)/J
85 85 K=K+1
86 86 GO TO 63
87 87 510 CONTINUE
88 88 DO 555 I=1,8
89 89 555 WRITE(2,0) (P(I,J),J=1,6)
90 90 DO 666 I=1,8
91 91 666 WRITE(2,0) (R(I,J),J=1,7)
92 92 KZ=3
93 93 KI=2
94 94 D(1)=P(1,1)
95 95 ZE=D(1)
96 96 334 CONTINUE
97 97 DO 333 IP=1,2
98 98 IPP=3-IP
99 99 SK=3-IP*2
100 100 IK=W-1
101 101 DO 333 K=1,IK
102 102 IR=P(K,1)
103 103 IF(IR.EQ.0) GO TO 333
104 104 IF(P(IR,IP).NE.ZE) GO TO 333
105 105 IF(KZ.EQ.1) KI=KI+1
106 106 KZ=3

```

SOURCE UPDATE
LISTING

09. 06. 1982

STRONA 3

107	107		KY=3
108	108		IZ=KI+1
109	109		D(IZ)=P(IR,IPP)
110	110		ZE=P(IR,IPP)
111	111		D(KI)=IR*GN
112	112		P(K,1)=0
113	113		KI=KI+2
114	114		GO TO 334
115	115	333	CONTINUE
116	116		IF(KZ.EQ.3) GO TO 339
117	117		KY=KY+2
118	118		KI=KI-1
119	119		IZ=KI-KY
120	120		IF(IZ.LE.0) GO TO 340
121	121	339	D(KI)=999
122	122		IK=KI+1
123	123		D(IK)=D(IZ)
124	124		ZE=D(IZ)
125	125		KI=KI+1
126	126		KZ=1
127	127		GO TO 334
128	128	340	CONTINUE
129	129	341	KI=I-2
130	130		IF(D(KI).EQ.999) GO TO 341
131	131		L=IABS(D(KI))
132	132		IK=D(KI)
133	133		IZ=ZI+1
134	134		IZ=D(IZ)
135	135		IZ=IZ*SIGN(IK)
136	136		X(L)=Q(IZ)
137	137		IK=KI-1
138	138		IK=D(IK)
139	139		Q(IK)=Q(IK)+Q(IZ)
140	140		IF(KI.LE.2) GO TO 342
141	141		GO TO 341
142	142	342	CONTINUE
143	143		NW=N-W+1
144	144		CALL TIMEAT(KP)
145	145		CALL TIME(LP)
146	146		K=1
147	147		DO 202 JA=1,LI
148	148		S=0
149	149		DO 503 J=,NW
150	150		AA=0
151	151		C=0
152	152		INK=SP(J)
153	153		DO 386 KW=1,INK
154	154		PA=ABS(B(J,KW))
155	155		DD=1
156	156		DO 99 NI=1,KA
157	157	99	DD=D*X(PA)
158	158		DD=DD*R(PA)
159	159		IF(X(PA).EQ.0.)X(PA)=1.0E-10

SOURCE UPDATE
LISTING

09. 06. 1982

STRONA 4

160	160		PP=FLOAT(PA/B(J,KW))
161	161		DI=DD*X(PA)
162	162		DA=(SIGN(DI,X(PA))-H(PA))*PP
163	163		INK=SP(J)
164	164		C=C+ABS(DD)
165	165		C=+ABS(DD)
166	166		AA=AA+DA
167	167		IF(C.EQ.0.) GO TO 503
168	168		DH=AA/C
169	169		DH=AA/C
170	170		NT=KA+1
171	171		RE=FLOAT(NT)
172	172		DH=-DH/RE
173	173		DE(K,J)=DH
174	174		INK=SP(J)
175	175		DO 19KW=1,INK
176	176		PA=IABS(B(J,KW))
177	177		PP=FLOAT(PA/(J,KW))
178	178	19	X(PA)=X(PA)+DH*PP
179	179		S=S+ABS(DH)
180	180	503	CONTINUE
181	181	222	CONTINUE
182	182		IF(S.LE.PS) GO TO 203
183	183		IF(JA.EQ.L1) GO TO 44
184	184		IF(K.NE.4) K=K+1
185	185		IF(K.EQ.4) K=1
186	186	202	CONTINUE
187	187		GO TO 203
188	188	444	CONTINUE
189	189		E=0
190	190		D=0
191	191		C=0
192	192		AA=DE(4,J)/DE(3,J)
193	193		C=DE(2,J)/DE(1,J)
194	194		C=C+DE(1,J)**2
195	195		D=D+DE(2,J)**2
196	196		E=E+DE(3,J)**2
197	197		IF(AA.GE.0.2) GO TO 44
198	198		C=SQR(C)
199	199		D=SQR(D)
200	200		E=SQR(E)
201	201		C=C/D
202	202		C=C-D
203	203		C=ABS(C-D)
204	204		IF(C.EQ.0.) GO TO 44
205	205		IF(C.EQ.5.) GO TO 44
206	206		C=DE(3,J)-DE(4,J)
207	207		IF(C.EQ.0.) GO TO 44
208	208		DH=DE(4,J)**2/C
209	209		INK=SP(J)
210	210		DO 44 KW=1,INK
211	211		PA=B(J,KW)
212	212		IF(PA.LE.0) PA=-PA

SOURCE UPDATE
LISTING

09. 06. 1982

STRONA 5

```

213 213      PP=LOAT(PA/B(J,KW))
214 214      X(PA)=X(PA)+DH*PP
215 215      44  CONTINUE
216 216      L1=L1+L2
217 217      K=1
218 218      GO TO 202
219 219      203  CONTINUE
220 220      CALL TIRDAT(KK)
221 221      CALL TIME(LK)
222 222      J=K(2)-LP(2)
223 223      I=KK(5)-KP(5)
224 224      L=KK(6)-KP(6)
225 225      WRITE(2,56)
226 226      WRITE(2,57)
227 227      WRITE(2,58)
228 228      WRITE(2,59)N,W,Z,L1,L2,EPS,KA
229 229      WRITE(260) (P(I,J),I=1,N),J=1,2), (R(J),J=1,N)
230 230      WRITE(2,61) (Q(I),I=1,W), (H(J),J=1,N)
231 231      WRITE(2,63) (X(I),I=1,N), (DE(I,50),I=1,N),F,I
232 232      56  FORMAT(6X,70(1H*))//28X,"ALGORYTHY OPTYMALIZACJI"/,9X,
233 233      -"NIELINIOWYCH SIECI PRZEPLYWOWYCH DROGA MINIMALIZACJI"
234 234      -"/29X,"STRAT ENERGETYCZNYCH"/,5X,0(1H*))
235 235      57  FORMAT(25X,"WYNIKI OBLICZEN PROGRAMU ALG1"/33X,"PRZYKŁAD 1"/,22X,
236 236      -38(1H*))
237 237      58  FORMAT(26X,"DANE WEJŚCIOWE"/25X,14(1H-))
238 238      59  FORMAT(11X,"N="I1,3X,"W="I1,3X,"Z="I1,3X,"L1="I1,3X,"L2="I1,3X,
239 239      -"L2="I1,3X,"EPS="F7.5,3X,"KA="I1/)
240 240      60  FORMAT(17X,8(6X)I1)/10X,"P(I,J)="5X,I1,7(6X,I1)/,17X,8(F7.2)/)
241 241      61  FORMAT(11X,"Q=5F4.0)/10X,"H="8(F5.1)/)
242 242      62  FORMAT(26X,"WYNIKI OBLICZEN"/25X,16(1H-)/11X,"ROZWIĄZANIE OPTYMAL
243 243      -NE"/10X,22(1H-))
244 244      63  FORMAT(12X,"X="8(F8.4)/,11X,"Y="8(F8.4)/,11X,"WARTOŚĆ FUNKCJI
245 245      -CELU"/,10X,22(1H-)/12X,"F(X,Y)="F8.4/10X,"CZAS OBLICZEN"/10X
246 246      -CELU"/,10X,22(1H-)/34X,"F(X,Y)="F8.4/10X,"CZAS OBLICZEN"/10X,
247 247      -14(1H-)/34X,"T="I3," SEKUND")
248 248  EXI
** ILOSC REKORDOW W ZBIORZE WYJŚCIOWYM = 248

```

9.2. Postać źródłowa programu ALG2 - opracowanego
w oparciu o algorytm II

SOURCE UPDATE
LISTING

18. 06. 1982

STRONA 1

1	1	LIST
2	2	PROGRAM(ALG2)
3	3	INPUT 1=RO
4	4	OUTPUT 2=LP0
5	5	TRACE 2
6	6	END
7	7	
8	8	MASTER OPTIMALIZACJA
9	9	REAL P(100,4),H(100),QW(100),X(100),HN(100),HG(100)
10	10	REAL A(100,20),HH(100),QMA(10),GOL(30)
11	11	INTEGER PA,W,XX,Y,S(100),NN(10),NK(10)
12	12	READ(1,506) IH
13	13	READ(1,200) N,W,Z,LI,KA,EPS
14	14	READ(1,528) (QW(I),H(I),HN(I),HG(I),I=1,W)
15	15	READ(1,298) ((P(I,J),J=1,3),I=1,N)
16	16	506 FORMAT(I0)
17	17	200 FORMAT(4I0)
18	18	528 FORMAT(200F0.0)
19	19	298 FORMAT(1000F0.0)
20	20	K,NIK=1
21	21	AGA=0.5
22	22	DO 508 I=1,W
23	23	HN(I)=HN(I)+H(I)
24	24	HH(I)=H(I)
25	25	H(I)=H(I)/3
26	26	508 CONTINUE
27	27	DO 989 I=1,N
28	28	989 P(I,3)=SORT(P(I,3))
29	29	DO 100 J=1,W
30	30	DO 10 IP=1,2
31	31	IPP=3-IP
32	32	DO 8 I=1,N
33	33	IF(P(I,IP).NE.J) GO TO 8
34	34	A(J,K)=P(I,IPP)
35	35	K=K+1
36	36	A(J,K)=P(I,3)
37	37	K=K+1
38	38	8 CONTINUE
39	39	10 CONTINUE
40	40	S(J)=(K-1)/2
41	41	K=1
42	42	100 CONTINUE
43	43	DO 505 IC=1,IN
44	44	CALL ITIME(KP)
45	45	IF(NIK.EQ.1) GO TO 111
46	46	READ(1,507) (QW(I),I=LZ+1,W)
47	47	507 FORMAT(15F0.0)
48	48	111 CONTINUE
49	49	NIK=2
50	50	GOL(IC)=0.
51	51	DO 1002 I=LZ+1,W
52	52	GOL(IC)=GOL(IC)+QW(I)
53	53	1002 CONTINUE

SOURCE UPDATE
LISTING

18. 06. 1982

STRUNA 2

54	54		ANK=AKA+GOL(IC)
55	55		FS=0.1
56	56		DO 202 JA=1.LI
57	57		SS=0
58	58		DO 503 J=1,W
59	59		AA,BB=0
60	60		IF(J.LE.LZ) GO TO 503
61	61		DO 386 KW=1,S(J)
62	62		K=KW*2-1
63	63		PA=A(J,K)
64	64		DA=H(J)-H(PA)
65	65		IF(DA.GE.0) GO TO 17
66	66		DA=SQRT(-DA)
67	67		GO TO 19
68	68	17	DA=-SQRT(DA)
69	69	19	CONTINUE
70	70		IF(DA.EQ.0) GO TO 386
71	71		DN=A(J,K+1)*AGA/ABS(DA)
72	72		DA=DA*A(J,K+1)
73	73		BB=BB+DN
74	74		AA=AA+DA
75	75	386	CONTINUE
76	76		AA=AA-QW(J)
77	77		IF(BB.EQ.0) GO TO 503
78	78		DH=AA/BB
79	79		H(J)=H(J)+DH
80	80		SS=SS+ABS(DH)
81	81	503	CONTINUE
82	82		IF(SS.LE.EPS) GO TO 203
83	83	202	CONTINUE
84	84	203	CONTINUE
85	85		HA=H(1)
86	86		DO 320 I=2,W
87	87		IF(H(I).LE.HA) GO TO 321
88	88	320	CONTINUE
89	89		GO TO 38
90	90	321	HA=H(I)
91	91		GO TO 320
92	92	38	I=0
93	93		DO 206 IA=1,N
94	94		Y=P(IA,1)
95	95		XX=P(IA,2)
96	96		UK=H(Y)-H(XX)
97	97		E=HH(Y)-HH(XX)
98	98		U=ABS(UK)
99	99		IF(U.EQ.0) GO TO 206
100	100		UK=UK/U
101	101		E=-E
102	102		U=SQRT(U)
103	103		X(IA)=U*P(IA,3)*UK
104	104		FS=FS+ABS(X(IA)*X(IA)*U)
105	105		FS=FS+X(IA)*E
106	106	206	CONTINUE

SOURCE UPDATE
LISTING

18. 06. 1982

STRONA 3

```

107 107      AS=HD(1)-H(1)
108 108      DO 823 I=1,W
109 109      H(I)=H(I)-HA
110 110      IF ((HD(I)-H(I)).LE.AS) GO TO 823
111 111      AS=HD(I)-H(I)
112 112      823 CONTINUE
113 113      DO 824 I=1,W
114 114      824 H(I)=H(I)+AS-HH(I)
115 115      BW=0.
116 116      KI=1
117 117      DO 180 I=1,L7
118 118      DO 182 KW=KI,S(I)+KI-1
119 119      182 BW=BW+H(I)*X(KW)
120 120      180 KI=S(I)+KI
121 121      DO 181 I=1,W
122 122      181 H(I)=H(I)-AS+HH(I)+HA
123 123      CALL ITIME(KK)
124 124      KK=KK-KP
125 125      WRITE(2,56)
126 126      WRITE(2,57)
127 127      WRITE(2,58)
128 128      WRITE(2,59)N,W,Z,I,I,KA,EPS
129 129      WRITE(2,60) (P(I,J),J=1,4),I=1,3)
130 130      WRITE(2,12) (QW(I),I=1,4), (H(I),I=1,4)
131 131      WRITE(2,63) (X(I),I=1,4), (HH(I),I=1,4),FS,KK
132 132      56  FORMAT(6X,70(1H*)//24X,"OPTYMALIZACJA WARUNKOW PRACY"/,9X,
133 133      -"NIELINIOWYCH SIECI PRZEPLYWOWYCH DROGA MINIMALIZACII"
134 134      -"/29X,"STRATA ENERGIIYCZNYCH"/,5X,0(1H*))
135 135      57  FORMAT(25X,"WYNIKI OBLICZEN PROGRAMU #ALG2"/33X,"PRZYKLAD 2"/,22X,
136 136      -38(1H*))
137 137      58  FORMAT(26,"DANE WEJSCIOWE"/5X,14(1H-)/)
138 138      59  FORMAT(14X,"N=",I1,4X,"W=",I1,4X,"Z=",I1,4X,"LI=",I1,4X,
139 139      -"KA=",I1,4X,"EPS=",F6.4/)
140 140      60  FORMAT(36X,4(F4.0,3X)/26X,"P(I,J)=",4(F4.0,3X)/36X,4(F4.0,3X)/)
141 141      12  FORMAT(14X,"QW=",4X(F4.0)//14X,"H=",4(F4.0)/)
142 142      62  FORMAT(26X,"WYNIKI OBLICZEN"/25X,16(1H-)/11X,"ROZWIATAZANIE OPTYMAL
143 143      -NE "/10X,22(1H.)/)
144 144      63  FORMAT(12X,"X="8(F8.4)/,11X,"Y="8(F8.4)/,11X,"WARTOSC FUNKCJI
145 145      -CELU"/,10X,22(1H.)/12X,"F(X,Y)="F84/10X,"CZAS OBLICZEN"/10X
146 146      -14(1H.)/11X,"T=I3,SEKUND")
147 147      END

```

9.3. Wyniki obliczeń programu ALG1 dla przykładu 1

ALGORYTMY OPTIMALIZACJI
NIELINIOWYCH SIECI PRZEPYLOWYCH DROGA MINIMALIZACJI
STRAT ENERGETYCZNYCH

WYNIKI OBLICZEN PROGRAMU ALG1
PRZYKŁAD 1

DANE WEJŚCIOWE

N=8 W=5 Z=2 LI=100 L1=4 L2=6 EPS=0.00001 KA=1

	1	1	2	2	2	3	3	4
P(I,J)=	3	4	3	5	4	5	4	5
	0.01	0.02	0.04	0.05	0.04	0.03	0.02	0.01

R= 0. 0. 10. 5. 15.

H= -1.0 1.0 -1.5 -1.0 0.5 0.5 2.0 -1.5

WYNIKI OBLICZEN

ROZWIĄZANIE OPTIMALNE

X= 10.3002 8.3760 3.1398 3.8928 4.2911 -6.2123 9.6524 1.4548
Y= -0.0609 -2.4032 -1.1057 -0.2423 -1.2365 -1.6578 -3.8634 -1.5212

WARTOŚĆ FUNKCJI CELE

F(X,Y)= 80.2795

CZAS OBLICZEN

T= 1 SEKUND

9.4. Wyniki obliczeń programu ALG2 dla przykładu 2

ALGORYTHY OPTYMALIZACJI
NIELINIOWYCH SIECI PRZEPYLOWYCH DROGA MINIMALIZACJI
STRAT ENERGETYCZNYCH

WYNIKI OBLICZEN PROGRAMU ALG2
PRZYKLAD 2

DANE WEJSCIOWE

N=4 W=4 Z=2 LI=100 KA=1 EPS=0.0001

P(I,J)=
1. 2. 2. 1.
3. 3. 4. 4.
34. 640. 17. 4.

QW= 0. 0. 5. 12.

H = 2. 3. 5. 2.

WYNIKI OBLICZEN

ROZWIĄZANIE OPTYMALNE

.....

X= 4.0615 0.9385 3.9213 8.0787

Y= -561.3272 -561.6606 -261.3973 -261.3973

WARTOSC FUNKCJI CELU

.....

F(X,Y)= 5941.0229

CZAS OBLICZEN

.....

T= 1 SEKUND