XVIII Seminarium

ZASTOSOWANIE KOMPUTERÓW W NAUCE I TECHNICE' 2008
Oddział Gdański PTETiS
Referat nr 10

ALGORITHM AND PROGRAM FOR FINDING MINIMAL AND QUASI-MINIMAL CUTS IN GRAPHS

Andrey GRISHKEVICH[1], Łukasz PIĄTEK[2]

1. Politechnika Częstochowska, al. Armii Krajowej 17, 42-200 Częstochowa
   Tel, fax:0343250831 e-mail: grishkev_ramb@rambler.ru
2. Politechnika Częstochowska, al. Armii Krajowej 17, 42-200 Częstochowa
   Tel, fax:0343250831 e-mail: l_piatek@el.pcz.czest.pl

Abstract: It was found, that the set of minimum cuts, separating two chosen vertices in graph, have the structure of distributive lattice. It was developed an effective procedure for finding the set of 1, 2 and 3 elements cuts in graph based on the consideration of distributive lattice of the set of minimum cuts. The procedure consists of first, the algorithm for finding indecomposable minimal cuts of distributive lattice. Second, algorithm for synthesis, using resulting subset from stage one, of the entire set of minimum cuts. The third, is the algorithm for describing the set of quasi-minimum (close to the minimum, next to minimum) cuts in the form of sum of distributive lattices of minimal cuts found for the modified function of weight. The computer program, implementing these algorithms, is presented with examples.

Keywords: distributive lattice, indecomposable cut, one-, two-, three- elements cut

1. FINDING MINIMAL CUTS IN GRAPH

1.1. The distributive lattice of the minimal cuts of graph

Let $G=(V,U)$ – be a directed graph, where $V=\{v\}$ – set of vertices of the graph, $U=\{u=(i,j) : i\in V, j\in V\}$ – set of directed edges of the graph.

In graph G we can choose two vertices – the source s and the sink t ($s,t\in V$, $s\neq t$). Let A,B $(A\cap B=\varnothing)$ be some subsets of set of vertices. We can designate $(A,B)=\{(i,j) : (i,j)\in U, i\in A, j\in B\}$ set of directed edges leading from $i\in A$ to $j\in B$. In addition we can assume, that, first, between any two vertices $i,j\in V$ there is no more than one directed edge $(i,j)\in U$ and one directed edge $(j,i)\in U$, and secondly, there are no loops (i.e. edges of a kind $(i,i)\notin U$).

The cut [1], dividing vertices s, t of graph G, is a set of edges $r=(R,\overline{R})\subseteq U$, where $R\cap\overline{R}=\varnothing$, $R\cup\overline{R}=V$, $s\in R$, $t\in\overline{R}$. Set of all such cuts we can designate as $\mathbf{R}$.

To each edge $u\in U$ of the graph G we assign a nonnegative number $c(u)\geq 0$, which is named as weight (capacity) of the edge. Capacity (weight) of the cut we define as

$$c(r)=c(R,\overline{R})=\sum_{u\in(R,\overline{R})}c(u). \tag{1}$$

In set of cuts $\mathbf{R}$ of graph G a subset of the minimal cuts (cuts of the minimal weight) is allocated

$$\mathrm{M}_{\min,c}=\left\{m : m=\operatorname*{argmin}_{r\in\mathbf{R}}c(r)\right\}. \tag{2}$$

On set $\mathrm{M}_{\min,c}$ binary operations $\vee,\wedge$ are defined. For any

$$m_i=(M_i,\overline{M_i})\in\mathrm{M}_{\min,c},\ i=1,2, \tag{3}$$

we put

$$m_1\vee m_2=(M_1\cup M_2,\overline{M_1\cup M_2}), \tag{4}$$

$$m_1\wedge m_2=(M_1\cap M_2,\overline{M_1\cap M_2}). \tag{5}$$

The set of the minimal cuts $\mathrm{M}_{\min,c}$ with defined on it operations $\vee,\wedge$ is a distributive lattice $<\mathrm{M}_{\min,c};\vee,\wedge>$ [2,3].

The minimal cut $p\in\mathrm{M}_{\min,c}$ of a distributive lattice is undecomposable ($\vee$-undecomposable) [2,3], if for any $m_1,m_2\in\mathrm{M}_{\min,c}$ relation $p=m_1\vee m_2$ follows $p=m_1$ or $p=m_2$. We shall designate P_c the set of undecomposable cuts of the lattice $<\mathrm{M}_{\min,c};\vee,\wedge>$. It is obvious, that P_c is a partially ordered set as a subset of partially ordered set $\mathrm{M}_{\min,c}$.

The set of minimal cuts of graph in the distributive lattice can be analytically described [2,3]

Recenzent. Dr inż. Walerian Gruszczyński – Wydział Elektrotechniki, Telekomunikacji i Informatyki Politechnika Gdańska

$$\mathrm{M}_{\min,c} = \bigcup_{A \in \mathbf{A}(P_c)} (\underset{a \in A}{\vee} a), \quad (6)$$

where $\mathbf{A}(P_c)$ – set of antichains A of partially ordered set P_c.

The specified representation, shown above, forms a basis for a new decomposition approach to finding minimal cuts in graph. It consist, first, from searching only undecomposable minimal cuts in graph and, secondly, from synthesizing in distributive lattice of minimal cuts all set of minimal cuts on partially ordered subset of undecomposable cuts. The offered approach allows to reduce search in graph (number of graph's connection checks) due to allocation of only subsets of undecomposable minimal cuts.

1.2. Algorithm for searching the minimal cuts of weight *k* in graph

Let $S \subseteq U$ be the subset of edges in graph. We shall construct function of weight $c_S(u): U \to R^+$

$$c_S(u) = \begin{cases} \infty, & if \quad u \in S, \\ 1, & if \quad u \in U \setminus S. \end{cases} \quad (7)$$

Set of the minimal cuts concerning function $c_S(u)$ we designate as $\mathrm{M}_{\min,c_S}$. For the whole pre-assigned k we define the set M_k

$$\mathrm{M}_k = \begin{cases} \mathrm{M}_{\min,c_S}, & if \;\; c_S(m) = k \quad \forall m \in \mathrm{M}_{\min,c_S}, \\ \varnothing, & if \;\; c_S(m) \neq k \quad \forall m \in \mathrm{M}_{\min,c_S}. \end{cases} \quad (8)$$

Thus, set M_k contains cuts of graph G between vertices s and t of weight k (k-element cuts), which are minimal relating to function weights $c_S(u)$. If such cuts does not exist, then $\mathrm{M}_k = \varnothing$.

Let's consider the algorithm

$$\text{KCUT}\ (G;\ s, t;\ S;\ k;\ \mathrm{M}_k) \quad (9)$$

for finding set of the k (k=1, 2, 3) element cuts M_k, dividing vertices s and t ($s,t \in V$, $s \neq t$) of a directed graph G and minimal relating to function of weight $c_S(u)$.

Auxiliary sets $R_1^k, R_2^k, \ldots, R_k^k \subseteq U$ we define as follows: for all $i \neq j$ fairly $R_i^k \cap R_j^k = \varnothing$; for all $m \in \mathrm{M}_k$ we have $R_i^k \cap m \neq \varnothing$, i, j = 1, 2, ... , k.

The detailed description of work of algorithm KCUT by search of single-element minimal cuts is resulted in [2], two-element – in [4], three-element – in [5].

2. ENUMERATION OF QUASI-MINIMAL CUTS IN GRAPH

2.1. Statement of a problem

Under single-($\mathbf{M}_1$), two-($\mathbf{M}_2$) and three-element ($\mathbf{M}_3$) cuts in graph G we shall understand accordingly cuts of weight one, two and three in case of capacity value $c(u) = 1$ for all $u \in U$. Such name is justified by that single-element (two-element, three-element) cuts consist of one (two, three) elements (edges in graph). Thus, for example, two-element (three-element) cuts at presence of single-element (and/or two-element) cuts are not minimal. But are close to minimal (following for minimal, quazi-minimal cuts). The problem consists in finding all elements of the specified sets (enumeration minimal and quazi-minimal cuts).

Enumeration quazi-minimal relating to function of weight $c(u) = 1$ for all $u \in U$ two- and three-element cuts can be summarised (trace) to the sequence of problems of enumeration minimal two and three-element cuts relating to function of weight $c_S(u)$ for some set of sets $\{\mathbf{S}\} = \mathbf{S}^*$.

2.2 Finding of two-element (three-element) cuts at existence of single-element (and absence two-element) cuts

Two-element (three-element) cuts in this case are not minimal. The set of edges in graph G, forming single-element cuts, is R_1^1. It is obvious, that any edge of that kind cannot enter into a two-element (three-element) cut. Giving to the specified cuts' edges sufficiently large weights (which forbid occurrence of corresponding edges in the minimal cuts set), it is possible to achieve, that two-element (three-element) cuts will be minimal. The finding of two-element (three-element) cuts can be carried out on the basis of

$$\text{KCUT}\ (G;\ s, t;\ R_1^1;\ 2;\ \mathbf{M}_2) \quad (10)$$

$$(\text{KCUT}\ (G;\ s, t;\ R_1^1;\ 3;\ \mathrm{M}_3^1)). \quad (11)$$

2.3. Finding of three-element cuts at existence two-element and absence of single-element cuts

Three-element cuts in this case are not minimal. The three-element cut can: 1) not contain edges of two-element cuts, 2) contain one edge of a two-element cut, 3) contain one edge of two various two-element cuts, namely, 3a) elements of two-element cuts lay in different sets R_1^2, R_2^2; 3b) elements of two-element cuts lay either in set R_1^2 or in set R_2^2.

Case 1. The set of edges in graph G, forming two-element cuts, is $R_1^2 \cup R_2^2$. It is obvious, that any such edge cannot enter into a demanded three-element cut. And the finding of three-element cuts can be carried out on the basis of

$$\text{KCUT}\ (G;\ s, t;\ R_1^2 \cup R_2^2;\ 3;\ \mathrm{M}_3^2). \quad (12)$$

Case 2. For any two-element cut $m \in \mathbf{M}_2$ fairly $m \cap R_1^2 = \varnothing$, $m \cap R_2^2 = \varnothing$, and $R_1^2 \cap R_2^2 = \varnothing$. Accordingly, set of three-element cuts, which contain an edge of a two-element cut from set R_2^2 (R_1^2), can be found by means of

$$\text{KCUT}\ (G;\ s, t;\ R_1^2;\ 3;\ \mathrm{M}_3^3) \quad (13)$$

$$(\text{KCUT}\ (G;\ s, t;\ R_2^2;\ 3;\ \mathrm{M}_3^4)). \quad (14)$$

Really, giving infinite weights for edges in R_1^2 (R_2^2) forbids all two-element cuts. For any $m \in \mathbf{M}_2$ it is fair to $m \cap R_1^2 \neq \varnothing$ ($m \cap R_2^2 \neq \varnothing$), however use of edges R_2^2 (R_1^2) at designing three-element cuts.

Case 3a. We consistently use all edges $u \in R_1^2$. It is necessary for finding three-element cuts, which contain the edge $u \in R_1^2$ and any from edges R_2^2: 1) to forbid occurrence of edges $R_1^2 \setminus u$ (they cannot enter into a three-element cut simultaneously with the edge u); 2) to forbid the two-element cuts, one of which edge is the edge u, thus having resolved occurrence of an edge u, that is reached by allocation of set of edges

$$R_2^{2u} = \{y : y \in R_2^2, (u, y) \in \mathbf{M}_2\}. \quad (15)$$

Thus, prohibition of set $\mathbf{S}_u = (R_1^2 \setminus u) \cup R_2^{2u}$ allows to find all demanded two-element cuts with an edge u. I.e.

$$\mathbf{S}^* = \{\mathbf{S}_u = (R_1^2 \setminus u) \cup R_2^{2u} : u \in R_1^2\}, \quad (16)$$

$$\mathrm{M}_3^5 = \bigcup_{\mathbf{S}_u \in \mathbf{S}^*} \mathrm{M}_{\min, c_{S_u}}, \quad (17)$$

where $\mathrm{M}_{\min, c_{S_u}}$ can be found on the basis of

$$\text{KCUT } (G;\, s,\, t;\; \mathbf{S}_u;\, 3;\; \mathrm{M}_{\min, c_{S_u}}). \quad (18)$$

Case 3b. Set of three-element cuts, which contain two elements from set R_2^2 (R_1^2), can be found by means of

$$\text{KCUT } (G;\, s,\, t;\; R_1^2;\, 3;\; \mathrm{M}_3^6 = \mathrm{M}_3^3), \quad (19)$$

$$(\text{KCUT } (G;\, s,\, t;\; R_2^2;\, 3;\; \mathrm{M}_3^7 = \mathrm{M}_3^4), \quad (20)$$

that corresponds to a case 2.

2.4. Finding of three-element cuts at existence one-and two-element cuts

In this case for any set $\mathbf{S}$, used at definition of three-element cuts, at existence two-element cuts, it is required to add set R_1^1.

2.5. Finding one, two and three-element cuts

Algorithm for finding one-, two- and three-element cuts ($\mathbf{M}$) it is possible to present finally as follows.

Step 1. To allocate set of single-element cuts $\mathbf{M}_1$ of graph

$$\text{KCUT } (G;\, s,\, t;\; \varnothing;\, 1;\; \mathbf{M}_1). \quad (21)$$

Step 2. To allocate set of two-element cuts of graph

$$\text{KCUT } (G;\, s,\, t;\; R_1^1;\, 2;\; \mathbf{M}_2). \quad (22)$$

Step 3. To allocate set of cuts M_3^3, M_3^4

$$\text{KCUT } (G;\, s,\, t;\; R_1^2 \cup R_2^2;\, 3;\; \mathrm{M}_3^3), \quad (23)$$

$$\text{KCUT } (G;\, s,\, t;\; R_1^1 \cup R_2^2;\, 3;\; \mathrm{M}_3^4), \quad (24)$$

$$\mathbf{M}_3 = \mathbf{M}_3 \cup \mathrm{M}_3^3 \cup \mathrm{M}_3^4. \quad (25)$$

Step 4. For all $u \in R_1^2$ to execute

{

$$R_2^{2u} = \{y : y \in R_2^2, (u, y) \in \mathbf{M}_2\}, \quad (26)$$

$$\text{KCUT } (G;\, s,\, t;\; R_1^1 \cup (R_1^2 \setminus u) \cup R_2^{2u};\, 3;\; \mathrm{M}_{\min, c_{S_u}}), \quad (27)$$

$$\mathbf{M}_3 = \mathbf{M}_3 \cup \mathrm{M}_{\min, c_{S_u}}. \quad (28)$$

}

Step 5. $\mathbf{M} = \mathbf{M}_1 \cup \mathbf{M}_2 \cup \mathbf{M}_3$.

3. THE COMPUTER PROGRAM

3.1. The computer program

The computer program [6] implements the algorithm for finding minimal and quasi-minimal (one-, two- and three-elements) cuts finding. The program finds cuts in graphs with edges directed and undirected and through edges or/and vertices [5].

The graph his described in a input text file. The results of program work are written in an output text file.

The program's interface permits to: limit cuts set to such cuts, which divide the graph through elements defined by the user; ability to automatic numbering of edges and vertices in graph; verification of input data integrity; graphical user interface.

The program was written in C++ language, uses libraries of STL (Standard Template Library) and contains 1700 lines of code.

3.2. Example 1

Finding minimal and quazi-minimal cuts of the scheme of electric connections of substation Fig.1 at an estimation of its reliability [7,8].

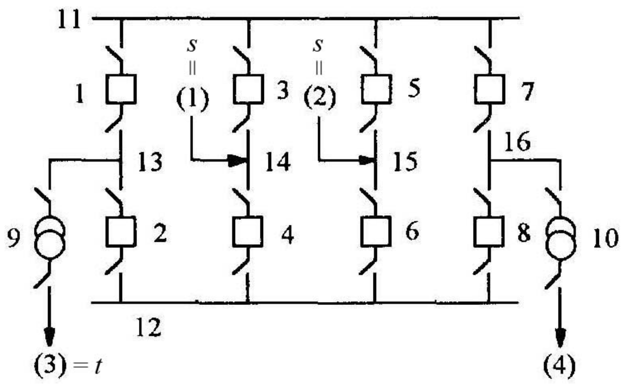

Fig. 1. Double main bus station configuration [7,8]

Below the sets, illustrating work of algorithm of finding one, two and three-element cuts, are given: $\mathbf{M}_1$={9, 13}; $\mathbf{M}_2$={(1, 2), (1, 12), (11, 2), (11, 12), (14, 15)}; R_1^2={14, 11, 1}, R_2^2={15, 12, 2}; M_3^3={(5, 12, 3), (15, 4, 3), (15, 12, 3)}, M_3^4={(6, 5, 14), (6, 11, 14), (6, 11, 4)}; $R_2^{2,14}$={11, 1, 15}, $M_{\min,c_{S_{14}}}$={(6, 5, 14), (12, 5, 14), (12, 5, 3)}; $R_2^{2,11}$={14, 1, 2, 12}, $M_{\min,c_{S_{11}}}$={(6, 4, 11), (15, 4, 3), (15, 4, 11)}; $R_2^{2,1}$={14, 11, 2, 12}, $M_{\min,c_{S_1}}$={(15, 4, 3)}; $\mathbf{M}_3$={(5, 12, 3), (15, 4, 3), (15, 12, 3), (6, 5, 14), (6, 11, 14), (6, 11, 4), (12, 5, 14), (15, 4, 11)}.

3.2. Example 2

Finding two and three-element cuts in finding the top estimation (estimation Esary-Proschan and/or Litwak-Uschakow) connectivity [9] bipolar networks Fig.2.

By means of the computer program following cuts $\mathbf{M}_2$={(1, 7), (3, 9)} ; $\mathbf{M}_3$= {(2, 4, 7), (2, 5, 8), (3, 6, 8)} are received. Allocation edges not crossed cuts {(1, 7), (2, 5, 8), (3, 9)} and {(2, 4, 7), (3, 6, 8)} does not represent special difficulties.

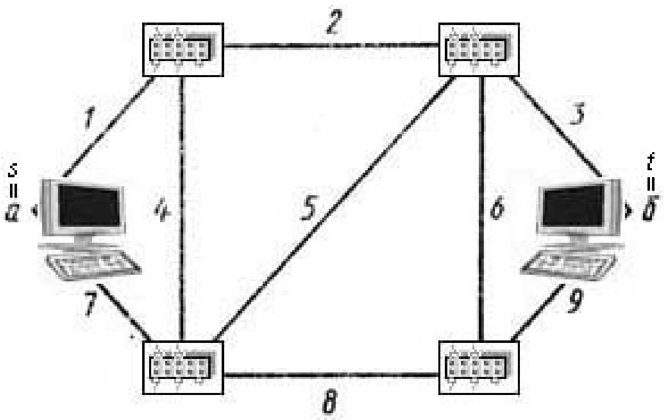

Fig. 2. Model of a communication network [9,10]

4. CONCLUSIONS

The original algorithm for finding one-, two- and three-element cuts in graph, using property of a distributive lattice of the minimal cuts, is developed. The computer program, realizing presented algorithm, can be used for estimation of reliability of complex systems of network structure.

5. REFERENCES

1. Ford L., Fulkerson D.: Flows in Networks Princeton University Press N. J., 1962, p. 194.
2. Grishkevich A.A.: Combinatorics methods of finding extreme structures of mathematic model of electric power transmission and systems Chelyabinsk Izd. SUSU 2004 (RU), p. 258, ISBN 5-696-02780-6.
3. Grishkevich A.A.: Distributive lattice of minimum cut sets of a directed graph, Informatyka teoretyczna i stosowana Vol. 4, Nr 7, Instyitut matematyki i informatyki Politechnika Czestochowska 2004, p. 7-22, ISSN 1643-2355.
4. Grishkevich A.A.: Algorithm for finding 2 elements minimal cut in directed graph, Informatyka Teoretyczna i Stosowana Vol. 6, Nr 10, Instyitut matematyki i informatyki Politechnika Czestochowska 2006 (PL), p. 85-100, ISSN 1643-2355.
5. Grishkevich A.A., Piątek Ł.: Algorithm for finding minimal 3 elements cuts in graph, Polish Journal of Environmental Studies Vol. 16, No 4a, 2007, p. 218-222, ISSN 1230-1485.
6. Grishkevich A.A., Piątek Ł.: Program of Enumeration of set 1, 2 and 3 element cuts – Accession number OFAP 10263, number of state registration (accession number VNTIC) 50200800673, 2008 Moskow (RU).
7. Billinton R., Lian G.: A new technique for active minimal cut set selection used in substation reliability evaluation, Electric Power Systems Research, Vol. 35, No. 5, 1995, p. 797-805, ISSN 0026-2714.
8. Filipiak S.: Methods of reliability estimations of high/medium voltage electrical substations, Numerical Methods and Computer Systems in Automatic Control and Electrical Engineering, Częstochowa University of Technology, 2005 (PL), p. 97-102, ISBN 83-7193-288-X.
9. Uschakov I.A.: The Estimation, forecasting and maintenance of reliability of networks of the COMPUTER Moscow Mechanical engineering, 1989 (RU), p. 71, ISBN 5-217-00704-4.
10. Ramirez-Marquez J.E., Coit D.W. : A Monte-Carlo simulation approach for approximating multi-state two-terminal reliability, Reliability Engineering and System Safety, Vol. 87, 2005, p. 253–264

ALGORYTM I PROGRAM ZNAJDOWANIA PRZEKROJÓW MINIMALNYCH I QUASI-MINIMALNYCH W GRAFACH

Słowa kluczowe: krata dystrybutywna, przekrój nierozkładalny, jeden, dwu, trzy elementowy przekrój

Ustalono, że zbiór minimalnych przekrojów, rozdzielających dwa zadane wierzchołki grafu, z wprowadzonymi na nim operacjami ma strukturę kraty dystrybutywnej. Opracowano skuteczną algorytmiczną procedurę znajdowania zbioru jeden, dwu i trzy elementowych przekrojów grafu bazującą na rozpatrzeniu dystrybutywnych krat zbioru minimalnych przekrojów grafu. Procedura składa się z, po pierwsze, algorytmu szukania nierozkładalnych minimalnych przekrojów kraty dystrybutywnej, po drugie, algorytmu syntezy po tym podzbiorze w kracie dystrybutywnej całego poszukiwanego zbioru minimalnych przekrojów i, po trzecie, algorytmu opisu zbioru quasi-minimalnych (bliskich do minimalnych, następnych po minimalnych) przekrojów w formie sumy krat dystrybutywnych minimalnych przekrojów, znalezionych dla zmodyfikowanej funkcji wagi. Przedstawiono zrealizowany program komputerowy i przedstawiono przykłady pracy programu