PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW
MERA-PIAP
Al. Jerozolimskie 202 02-222 Warszawa Telefon 23-70-81

Zespół Urządzeń i Systemów Sterowania BE10

440

Główny wykonawca dr inż. Wiesław Stańczak

Wykonawcy

Konsultant

Nr zlecenia S 1341

Efektywne wykorzystanie metody zespołów Max-minimalnych do analizy struktury organizacyjnej przedsiębiorstwa na podstawie wewnętrznego przepływu informacji. Przygotowanie publikacji do druku.

Zleceniodawca

Pracę rozpoczęto dnia 93.01.15 zakończono dnia 93.07.29

Kierownik Zespołu
wz
dr inż. A. Syryczyński

Z-ca Dyrektora d/s Badawczo-Rozwojowych
dr inż. J. Jabłkowski

Praca zawiera:

stron 2
rysunków
fotografii
tabel
tablic
załączników 1

Rozdzielnik - ilość egz: 3

Egz. 1 BOINTE
Egz. 2. ZSS
Egz. 3 ZSS
Egz. 4
Egz. 5
Egz. 6

Nr rejestr. 6982

Analiza deskryptorowa

TEORIA GRAFÓW+PODZIAŁ GRAFU NA PODGRAFY+ ZESPOŁY Max-minimalne+ALGORYTMIZACJA+ ZŁOŻONOŚĆ OBLICZENIOWA

Analiza dokumentacyjna

W pracy przytoczono algorytm wyszukiwania zespołów Max-minimalnych. Udowodniono jego prawidłowość, a następnie wykazano, że jest on klasy $O(n^2)$. Z poprzednich publikacji autora pracy wynika, że omawiany algorytm nadaje się do analizy struktury organizacyjnej przedsiębiorstwa na podstawie wewnętrznego przepływu informacji.

Tytuły poprzednich sprawozdań

UKD

MERA-PIAP/TW 331/78 5000

2

Skomputeryzowanie przedsiębiorstwa, poza wprowadzeniem niezbędnej infrastruktury sprzętowej (komputery i urządzenia we / wy, sieć - np. Novell) i programowej (edytory, bazy danych, programy natury ewidencyjno - księgowej, itp.) wymusza zmiany organizacyjne przedsiębiorstwa prowadzące do uefektywnienia procesu wprowadzania, uaktualniania, przepływu i wykorzystywania informacji zgromadzonej w centralnej bazie danych, bądź w rozproszonych bazach danych. Zarówno w przypadku centralnej bazy danych jak i w przypadku baz rozproszonych ważnym aspektem jest zadbanie o szybki (tzw. krótki czas reakcji systemu), bezkolizyjny dostęp do niezbędnych informacji.

Czynnikiem decydującym jest tu zarówno ograniczenie ruchu w sieci w wyniku wyeliminowania przesyłek zbędnych oraz zredukowania do niezbędnego minimum sięgania przez różne komórki organizacyjne do tych samych zbiorów danych, jak i ograniczenie obciążalności poszczególnych fragmentów bazy danych.

Wytyczną do realizacji zarysowanego wyżej celu uzyskać można grupując elementy zbioru wszystkich komórek organizacyjnych przedsiębiorstwa we wzajemnie rozłączne zespoły, posługując się przy tym odpowiednio określonym wskaźnikiem wzajemnego podobieństwa informacyjnego. Na przykład w przypadku realizacji rozproszonego modelu bazy danych każdemu zespołowi (ewentualnie kilku z nich) przypisana byłaby wewnętrzna, lokalna baza danych (podbaza), umiejscowiona w węźle głównym konkretnej podsieci przeznaczonej do obsługi komputerów użytkowanych w danym zespole komórek organizacyjnych. W ten sposób znaczna część ruchu zamykałaby się w danej podsieci. W przypadku scentralizowanej bazy danych zestaw wszystkich informacji można podzielić na poszczególne fragmenty zorientowane na wykorzystywanie przez poszczególne zespoły komórek, rozkładając bardziej równomiernie obciążenie fragmentów bazy głownej, a w ten sposób redukując czas reakcji systemu pomimo nie zmniejszonego globalnego ruchu w sieci.

W końcu 1991 roku powstała w PIAP koncepcja zastosowania do rozwiązania ww. zagadnień aparatu metody zespołow Max - minimalnych. Koncepcja ta była przedmiotem 2 publikacji w Biuletynie PIAP nr 5 - 157 / 91. Stanowiła ona kontynuację oryginalnego pomysłu omówionego w artykule: Stańczak W.: "An Introduction to Max - minimal Sets". Control and Cybernetics, Vol. 15, No. 1, 83 - 99.

Jednakże opublikowana w Biuletynie PIAP nr 5 - 157 / 91 metoda rozwiązywania zagadnienia wyszukiwania zespołow Max - minimalnych w zasadzie stanowi jedynie dowód konstrukcyjny algorytmizowalności problemu. Jest ona zbyt mało efektywna dla zastosowań praktycznych.

W pracy (Załącznik 1) zaproponowano efektywny algorytm wyznaczania zespołów Max - minimalnych. Jego złożoność obliczeniowa jest proporcjonalna do n^2, tzn nadaje się do praktycznych zastosowań, nawet w przypadku rozwiązywania problemów o dużej wymiarowości.

Zamieszczony w załaczniku 1 manuskrypt artykułu zostanie zgłoszony do druku w periodyku naukowym Control and Cybernetics.

Załącznik 1

An $O(n^2)$ algorithm for finding Max - minimal sets

Wiesław Stańczak
Industrial Research Institute for Automation and Measurements
Warszawa, Poland

In [9, 10] the relationships between the generation of Max - minimal sets and the evaluation of maximum capacities through a network is shown. A polynomial - type algorithm for solving the latter problem is proposed, due to the idea introduced by Hu [3]. The application of this algorithm and some additional features of Max - minimal sets (derived here) lead to a modification of the method for generating Max - minimal sets described in [10]. It is shown that this new method needs $O(n^2)$ elementary operations, while the previous one is of the $O(n^4)$ type.

1. Introduction

The concept of Max - minimal sets as introduced in [8] and redefined further in [10] can be applied for solving numerous problems of a graph partitioning type [8, 9]. These problems consist, in general, in dividing the set of vertices into subsets, such that the aggregate mutual connections between vertices (calculated as maxima among appropriate elementary connections) in a subset are greater than those between the vertices in the subset and the vertices outside it. A polynomial - type algorithm for solving this problem is proposed in [10]. It needs $O(n^4)$ elementary operations, i.e. in average it consumes a number proportional to $n^3/\ln n$ such operations for determining a single Max - minimal set, since the cardinality of the class of all Max - minimal sets in a graph with n vertices does not exceed $[1.6 + \ln(n - 1)]n$ [8]. This fact limits the applicability of the method to the cases of a less dimension and thus reduces it's practical usefulness.

The paper is organized as follows. First, an algorithm for calculating the maximum capacities through a network (this problem was first stated by Pollack [6]) is derived. In distinction to that described in [10], it bases on an idea

5

proposed by Hu [3]. Second, it is shown that the new algorithm needs $O(n^2)$ elementary arithmetic and/or computer operations. Third, the definition of Max - minimal sets is recalled and some their features are reminded. Fourth, a new characteristics of Max - minimal sets is proposed. It requires a lesser computational effort to check whether a nonempty proper subset of a given vertex - set constitutes a Max - minimal set than those stated previously in [8, 9]. That leads to a new algorithm for finding Max - minimal sets. Its basic idea is, in fact, the same as for the algorithm considered in [9, 10], but the new features derived here give a possibility to delete and/or redefine several steps of the previous algorithm, which results in decreasing the complexity as it will be seen in Section 4. Fifth, the remaining details of the implementation for the new algorithm are described and there is proved that it is of type $O(n^2)$.

2. Maximum capacities through a graph

Let X, $|X| = n > 1$, be a finite set and w be a mapping symmetric with respect to its arguments with the domain $\{\{x, y\} : x, y \in X\}$ and with the range consisting of the set of real numbers and a dummy value denoted throughout the paper by d. The latter is assumed to possess the following properties: $\max\{d, a\} = \max\{d, d\} = \min\{d, d\} = d$ and $\min\{d, a\} = a$ for any real a. We consider a weighted graph (G, w) defined by the complete undirected graph $G = (X, E)$ without self - loops (we assume that the reader is familiar with the basic notions of the classic graph theory as described, e.g., in [1, 5], and thus we do not remind appropriate definitions) having X as its vertex - set and $E = \{\{x, y\} : x, y \in X, x \neq y\}$ as its edge set, and the above mentioned function w, which describes weights attached to edges.

The Max - capacity $V(C)$ of a cut - set C is understood by means of the formula $V(C) = \max\{w(x, y) : \{x, y\} \in C\}$ [9, 10]. Moreover, let $P = (U, E_P)$, $U \subset X$, $|U| > 1$, be a (simple) path in G. The capacity $W(P)$ of P is defined as $W(P) = \min\{w(x, y) : \{x, y\} \in E_P\}$ [3, 6, 9, 10]. We refer to any cut set $C^* = C(x; y)$ separating two distinct vertices x and y (i.e such cut - set which deletion breaks all paths joining these vertices) as to Max - minimal cut - set if its Max - capacity attains the smallest value among all cut - sets in G separating the indicated x and y [9, 10]. Its value is denoted here by t_{xy} and called the (x, y)th terminal Max - capacity, $x, y \in X$, $x \neq y$. For convenience we additionally assume $t_{xy} = d$ for $x = y \in X$, and the square matrix $[t_{xy}]$ of order n is known as the terminal Max - capacity matrix for (G, w) [9, 10]. The terminal Max - capacity matrix for any given (G, w) can be realized by some weighted tree (T_R, w_{TR}), $T_R = (X, E_{TR})$ (see Theorem 1 in [9]) and by some weighted path (P_R, w_{PR}), $P_R = (X, E_{PR})$ (see Theorem 2 in [9]), which means that each (x, y)th terminal Max - capacity has the same value in (G, w) as for the mentioned tree and path (so - called the tree and / or the path realizing the terminal Max - capacity matrix of (G, w), respectively). The importance of paths realizing terminal Max - capacity matrices in the theory of Max - minimal sets becomes obvious in the light of [9, 10] and is confirmed in Section 3.

The maximum capacity W_{xy} through a weighted graph (G, w) between its distinct vertices x and y is the greatest capacity among all paths in (G, w) joining these x and y [3, 6, 10]. If we additionally define $W_{xy} = d$ for each $x = y \in X$, then we obtain (see Theorem 8 in [10], which is, in fact a black carbon copy of Theorem 1 in Section 8.7.2 of [1]) that the simple relation

$$t_{xy} = W_{xy}, \text{ for any } x, y \in X, \qquad (1)$$

connects maximum capacities through (G, w) and terminal Max - capacities for (G, w). Thus, since the method for the construction of path realizing a given terminal Max - capacity matrix is known (see the PRTM procedure in [10], being of type $O(n^2)$) and is rather efficient, then according to (1), it remains to pay our attention to derive a procedure which finds W_{xy}'s and uses to do it a less numbers of operations than $O(n^3)$ required for the EMPC procedure described in [10]. We begin with some modification of the implementation proposed by Kevin and Whitney [4] for the algorithm of Prim [7] and Dijkstra [2], which generates shortest spanning trees.

Spanning Tree with Maximum Capacities (STMC)

1. For each $x \in X$ set $L(x) := 0$.
2. Set $E_T := \emptyset$.
3. Take any $x \in X$.
4. Set $Y := X - \{x\}$.
5. For each $y \in Y$ set $\alpha(y) := x$ and $c(y) := w(x, y)$.
6. If $Y = \emptyset$, then FINISH the STMC.
7. Evaluate $c := \max\{c(y) : y \in Y\}$.
8. Find any $z \in Y$ such that $c(z) = c$.
9. Set $x := \alpha(z)$.
10. Update $E_T := E_T \cup \{\{x, z\}\}$.
11. $t_{xz} := c$ and $t_{zx} := c$.
12. $L(z) := L(z) + 1$; $L(x) = L(x) + 1$.
13. $A_z(L(z)) := x$; $A_x(L(x)) := z$.
14. $Y := Y - \{z\}$.
15. Define $U := Y$.
16. Examine whether $U = \emptyset$. If so, then pass to Step 6.
17. Take any $y \in U$.
18. Check whether $c(y) > w(z, y)$. If so, then go to Step 20.
19. Set $c(y) := w(z, y)$ and $\alpha(y) := z$.
20. Update $U := U - \{y\}$ and return to Step 16.

The STMC realizes the concept sketched by Hu in [3]. Let us now consider this procedure with more insight. The finiteness of X ensures that in each passage through Steps 7 and 8 some $z \in Y$ is chosen ($Y \neq \emptyset$ and $Y \neq X$ due to Steps 4, 6 and 14). Therefore, Steps 14 and 6 yield that the STMC terminates when the main iteration (consisting, in fact, of Steps 6 - 20) was performed exactly $n - 1$ times.

Due to Steps 7, 8 and the second part of Step 5, the relation $c(z) \leq t_{xz}$ is obvious. Let us now suppose that

$$c(z) < t_{xz}, \tag{2}$$

which is equivalent to the existence of a path in (G, w), say P, $P = (U, E_P)$, $x, z \in U$, joining x and z, such that $W_{xz} = W(P) > c(z)$, according to (1). On the other hand, since STMC is applied to a complete graph, then in each its main iteration the cut - set $C_{Y, X-Y}$ corresponding to the partition of X into Y and $X - Y$ is handled. Moreover, $V(C_{Y, X-Y}) = c(z)$, according to Steps 7, 5 and 15 - 20, and due to the

8

definition of Max - capacity. The evident relation $C_{Y,X-Y} \cap E_P \neq \emptyset$ leads to $c(z) \geq W_{xz} = W(P)$, by the definition of $W(P)$. Combining the last inequality with (1), we get $c(z) \geq t_{xz}$, which contradicts (2) and proves the validity of Step 11, (see also Steps 7 and 8).

The above remarks, the inspection of Steps 1, 12 and 13, and the fact that the STMC is an evident modification of the shortest spanning tree procedure stated in [2, 4, 7] can be summarized as follows.

Lemma 1. The STMC generates the (spanning) tree $T = (X, E_T)$ in $O(n)$ iterations and the weights t_{xz} $(\langle x, z\rangle \in E_T)$ are the (x, z)th terminal Max - capacities in (G, w), indeed. Moreover, for each $x \in X$ we have $L(x) > 0$, and $A_x(i)$, $i = 1, 2, \ldots, L(x)$, is a list of vertices adjacent to (i.e. joined by an edge with) x in T.

From Steps 5 and 14 - 20 it follows that the values of $\alpha(y)$'s used in Step 9 are taken from $X - Y$. Therefore in each main iteration the only edge being adjoined to E_T, say $\langle x, z\rangle$, (Step 10) connects a member of $X - Y$ with an element of Y (Step 8). Since x is taken into account by the STMC earlier than z, then it is convenient to say that x is older than z and/or that z is younger then x. These remarks yield the following property of STMC, which will be used further in this section.

Corollary 1, Let $\langle x, z\rangle \in E_T$. $A_x(1) = z$ and $A_z(1) = x$ if and only if either x or z is the starting vertex (i.e. chosen in Step 3) of STMC, and the other is obtained in the first main iteration. Moreover, the relations $A_z(1) = x$ and $A_x(1) \neq z$ are equivalent to the fact that x is older than z.

Any two distinct vertices of a tree are joined in it by a single path and no more such paths exist in that tree. Let u, $v \in X$, $u \neq v$, and we consider a path $P = (U, E_P)$ joining them in $T = (X, E_T)$ generated by the STMC. From the discussion above Lemma 1 it follows that each edge $\langle x, z\rangle \in E_P$ corresponds to some cut - set $C_{Y,X-Y}$ in (G, w) such that $x \in Y$ and $z \in X - Y$. Without any loss of generality we can assume that $u \in Y$. The additional supposition $v \in Y$ yields the existence of a path, say $P' = (U', E')$, joining u and v in a subtree $T' = (Y, E'_T)$ of T, $E'_T \subset E_T$. Since $E' \subset E'_T$ and $\langle x, z\rangle \notin E'_T$, then P' and P are distinct, i.e. we get a contradiction. Therefore, we have $v \in X - Y$. In other words, the considered $C_{Y,X-Y}$'s separate u and v in (G, w). That implies the inequality

$$t_{uv} \le W^{*}(P), \qquad (3)$$

where the asterisk indicates that the capacity of P' is evaluated in T instead of in (G, w).

On the other hand, P is also a path joining u and v in (G, w), since the latter is a complete graph. Thus, $C(u; v) \cap E_P \ne \emptyset$ for any cut - set $C(u; v)$ separating u and v in (G, w), i.e. $W(P) \le V(C(u; v))$ by definitions of Max - capacity and path's capacity. Moreover, Steps 5 and 15 - 20 imply that $t_{xz} = w(x, z)$ for any $\langle x, z\rangle \in E_T$, which yields $W^{*}(P) = W(P)$. Thus, by the definition of terminal Max - capacity, we get $W^{*}(P) \le t_{uv}$. Combining the latter with (3), we have $t_{uv} = W^{*}(P)$. Therefore, we obtain

Theorem 1. The STMC produces a tree realizing the terminal Max - capacity matrix of (G, w).

Thus, instead of constructing the terminal Max - capacity matrix directly, i.e. on a basis of (G, w), we can do it by using $T = (X, E_T)$, previously obtained with the aid of STMC.

Let us now consider the following procedure applied to $T = (X, E_T)$, $L(x)$'s, $A_x(p)$'s, $p = 1, 2, \ldots, L(x)$, and t_{xy}'s, where $x, y \in X$, obtained as an output of STMC described above:

Maximum Capacities through a Graph (MCG)

1. For each $x \in X$ set $t_{xx} := d$ (a dummy value), set $h(x) := 0$ and set $l(x) := L(x)$.
2. $k := 0$; $i := 1$.
3. Take any $x \in X$.
4. Set $s(1) := x$.
5. Take $y := A_x(l(x))$ (choosing a vertex adjacent to x).
6. Update $l(x) := l(x) - 1$.

7. Check whether $h(y) \neq 0$. If so (i.e. y was reached previously), then pass to Step 8. Otherwise, go to Step 10.
8. If $l(y) = 0$, then return to Step 5.
9. Set $x := y$ and return to Step 5.
10. $i := i + 1$; $k := k + 1$.
11. Set $s(i) := y$ and $h(y) := i$.
12. $j := k$.
13. Set $u = \min\{t_{s(j)x}, t_{xy}\}$.
14. $t_{ys(j)} := u$ and $t_{s(j)y} := u$.
15. $j := j - 1$.
16. If $j = 0$, then pass to Step 17. Otherwise, return to Step 13.
17. If $i := |X|$, then FINISH the MCG.
18. Set $x := y$ and return to Step 5.

Steps 1 - 4 constitute an introductory phase. The (main) iteration consists of Steps 5 - 18. Let us assume that we are now in Step 5. If we reach it for the first time, then $l(x) = L(x) > 0$ and $A_x(p)$, $p = 1, 2, \dots, l(x)$, is a complete list of vertices adjacent to x in $T = (X, E_T)$, due to Lemma 1, where $l(x)$ indicates the current value of $L(x)$. Otherwise we can arrive at Step 5 from either Step 18 or from Steps 8 and 9.

Let us now suppose that $l(x) = 0$ during the passage to Step 5, which is equivalent to the situation in which all the vertices adjacent to x in T were previously visited from x. If we return to Step 5 from Step 18, then this event must be recently preceded by performing Steps 5 - 7, and 10 - 17, i.e. the condition $h(y_{old}) = 0$ (where $x = x_{new} = y_{old}$, by Step 18) must be satisfied in Step 7. It means that $x = x_{new}$ was not previously visited, i.e. $l(x) > 0$, due to Lemma 1, which contradicts our supposition.

If we return to Step 5 from Step 9, then $l(y_{old}) \neq 0$ due to Step 8, $x = x_{new} = y_{old}$ according to Step 9, and we get the same result as in the previous case.

Let us now assume that we return to Step 5 from Step 8. Our supposition yields $A_x(1) = y$ (see Steps 5 - 6). First we consider the second case handled in Corollary 1, in which we have $A_y(1) \neq x$, and $A_y(\rho) = x$, for some index ρ such that $1 < \rho \leq L(y)$. But it implies that previously the passage from y to x has occured (since, otherwise, $l(y) = \rho > 1$) either via Step 9 or via Step 18. In both the cases we get $l(y_{new}) = l(x_{old}) = \rho - 1 > 0$, which contradicts the assumption that we return to Step 5 from Step 8. It remains to discuss the case $A_x(1) = y$ and $A_y(1) = x$. The relation $l(y) = 0$ (since we return to Step 5 from Step 8) implies that other vertices have been previously visited $L(y)$ times from y, and no passage via x occurred, due to Lemma 1.

Let us now discuss some specific situations. First, we assume that $L(z) > 1$ and each $u = A_z(i)$, $i = 2, 3, \ldots, L(z)$, is a pendant vertex (i.e. of degree one) in T, $z \in X - \{x, y\}$, where x and y are as above. Corollary 1 implies that $A_u(1) = z$ for any mentioned u. If in some iteration of MCG we visit z, then it is evident that we pass to $A_z(L(z))$, return to z, pass to $A_z(L(z) - 1))$, return to z again, pass to $A_z(L(z) - 2))$, etc., until we arrive at z and get $l(z) = 1$. The same, obviously, holds for $z = y$.

Let us now suppose that there exists a path $P = (U, E_P)$ in T, such that $U \subset X - \{x, y\}$, some $u = u_0 \in U$ is pendant in T, and other u's have degree two in T. Obviously, P is a subpath of some path P' passing via x and y in T. By $k(u)$, $u \in U$, we denote the number of vertices in P' whose separate y and u. It is evident that u is younger if and only if $k(u)$ is greater. The inspection of MCG shows that if we start with some $z = u \in U$, then first we visit all younger u's, if exist, and thus we return to older ones.

By deleting the edge $\{x, y\}$ from $T = (X, E_T)$ we obtain two subtrees, namely $T_y = (X_y, E_y)$ and $T_x = (X_x, E_x)$, where $X_x \cap X_y = \emptyset$, $X_x \cup X_y = X$, $x \in X_x$ and $y \in X_y$. Thus, combining the two specific cases considered above, we conclude that each vertex of X_y has been visited at least once before the $L(y)$th return to y. Therefore, Step 10 ensures that $i = |X_y|$. The same construction can be applied to T_x, but before we return for the $L(x)$th time to x, the condition $i = |X| = |X_y| + |X_x|$ has occurred, i.e. the performance of MCG has finished, due to Step 17. Thus, in this case the condition $l(x) = 0$ does not hold in Step 5.

Summarizing, in each (main) iteration of MCG the direction of passage is well - defined. Moreover t_{xy}'s used in Step 13 correspond to edges of T, because $y = A_x(l(x))$ (see Step 5). They were previously obtained in STMC. The values of $t_{s(j)x}$'s are calculated in the preceding iterations (see Steps 1, 4 and 10 - 16). Therefore, the substitution in Step 13 is well - defined. Moreover, it is consistent with the definition of path's capacity. By Steps 4, 10 - 11 and 17 all the $s(i)$'s, $i = 1, 2, \ldots, n = |X|$ are evaluated. Moreover, Steps 7 - 9 and 11 ensure that the inequality $i \neq i'$ is equivalent to $s(i) \neq s(i')$. Let us take r and q, $1 \leq r, q \leq n$, such that $u = s(r)$ and $v = s(q)$, $u, v \in X$. If $u = v$, then t_{uv} is evaluated in Step 1. If $u \neq v$, then without any loss of generality we can suppose that $r < q$. Thus the value of $t_{uv} = t_{vu}$ is determined in Steps 12 - 16 for $i = q$ and $j = r$. Finally, taking into account the stopping rule contained in Steps 2, 10 and 17 and using Theorem 1, we get

Theorem 2. The MCG applied to the results of STMC generates the terminal Max - capacity matrix of (G, w) in a finite number of iterations.

Now, it remains to estimate the complexity of STMC and MCG. Steps 7 and 8 of STMC can be realized simultaneously and require $|Y| - 1$ comparisons in each main iteration. In Steps 15 - 20 we perform $|Y|$ comparisons and at most $2|Y|$ substitutions in each main iteration. Therefore, due to Lemma 1, Steps 7, 8 and 15 - 20 need $O(n^2)$ operations for the whole STMC. Since in Steps 1 and 5 we perform $3n - 2$ substitutions only once, and the number of operations in Steps 2 - 4, 6 and 9 - 14 does not depend on n and/or $|Y|$, then the STMC is of type $O(n^2)$.

Steps 4, 5 and 6 of MCG ensure that from any vertex of X, say x, the vertices adjacent to it in T are visited at most $L(x)$ times (due to the stopping rule realized by Steps 2, 10, and 17), where $L(x)$ is, in fact, the degree of x in T. Since the sum of degrees of vertices in the tree T equals $2(n - 1)$, then visits to vertices of T are performed $O(n)$ times. The control of MCG consists of the second substitution in Step 1, Steps 2 - 11 and 17 - 18. Any of Steps 2 - 11 and 17 - 18 requires a constant number of simple arithmetic and/or elementary computer operations, which is independent of n. Since the mentioned steps (at least some of them) are executed during each visit to a vertex and the second substitution in Step 1 is performed n times, then the control of MCG consumes $O(n)$ operations. By the first substitution in Step 1 and by Steps 12 - 16 the appropriate entries of the terminal Max - capacity matrix are evaluated. The former costs n substitutions. Steps 13 - 16 consume two comparisons, three substitutions and a single subtraction per each pair t_{uv}, t_{vu}, $u, v \in X$, $u \neq v$, and Step 12 needs only one substitution for a whole loop performed during each visit to a vertex. Since each mentioned pair t_{uv}, t_{vu} is considered only once (due to Steps 7 - 9), then the part of MCG directly connected with the evaluation of terminal Max - capacity matrix is of type $O(n^2)$, and the same holds for the whole MCG.

Summarizing, the solution to the problem of maximum capacities through a graph (and thus the evaluation of terminal Max - capacity matrix (see (1)) with the aid of STMC and further by applying MCG to the results given by the former needs $O(n^2)$ elementary operations. For comparison, EMPC proposed in [10] produces the same output using $O(n^3)$ such operations.

3. Max - minimal sets and paths realizing terminal Max - capacity matrices

We begin with a convenient generalization of the definition of Max - minimal sets (introduced in [8] for complete graphs with non - negative edge weights and extended for any real weights in [10]). Namely, let $F = (X, E_F)$ be a subgraph of $G = (X, E)$ described in Section 2, and $w_{(F)} = w_{|E_F}$, i.e. a restriction of w to the domain E_F. For nonempty and disjoint subsets A and B of X we define

$$m_{(F)}(A, B) = \max\{\min\{w(x, y) : x, y \in X, x \neq y\}, w_{(F)}(x, y) : x \in A, y \in B\} \quad (4)$$

A nonempty proper subset S of X is called a Max - minimal set in (F, w) if the inequality $m_{(F)}(R, S - R) > m_{(F)}(R, X - S)$ holds for each $R \subset S$ such that $\emptyset \neq R \neq S$.

It can be easily verified that $m(A, B) = m_{(G)}(A, B) = \max\{w(x, y) : x \in A, y \in B\}$, which yields that for $F = G$ the above definition of Max - minimal sets is consistent with that given in [8, 9, 10]. Moreover, in the further discussion we need the following features of Max - minimal sets (compare with Theorem 1 in [8], Proposition 2 in [9] and Theorem 4 in [9], respectively).

Proposition 1. S is a Max - minimal set in (G, w), if and only if either $S = \{x\}$, $x \in X$, or S is a union of pairwise disjoint Max - minimal sets in (G, w), say $S_1, S_2, \ldots, S_k$, and the condition

$$m(Z, X - Z) < \min\{m(S_i, X - S_i) : i = 1, 2, \ldots, k\} \quad (5)$$

is satisfied for $Z = S$ and does not hold for each $Z = \bigcup_{i \in J} S_i$, where $J \subset \{1, 2, , k\}$, $1 < |J| < k$.

Proposition 2. If S is a Max - minimal set in (G, w), then the elements of S are consecutive vertices (not separated by any element of $X - S$) in each path realizing the terminal Max - capacity matrix of (G, w).

Proposition 3. Let S be a subset of X such that $1 < |S| < |X|$. Then S is a Max - minimal set in (G, w) if and only if the inequality

$$t_{xz} > t_{xy} \quad (6)$$

holds for each pair of distinct $x, z \in S$ and any $y \in X - S$.

Let $(G, \overline{w})$ be defined analogously as (G, w) in Section 2, but with the only distinction that $w(x, y)$'s are replaced by t_{xy}'s, $x, y \in X$. Moreover, throughout this section we also assume that $(P, \tilde{w})$, $P = (X, E_P)$, indicates a path realizing the terminal Max - capacity matrix of (G, w). Evidently, P is a subgraph of G, i.e. $\tilde{w} = \overline{w}_{(P)}$ (analogously, below we also write $\tilde{m}$ instead of $\overline{m}_{(P)}$). Thus, Theorem 3 of [9] can be rewritten as follows.

Proposition 4. If S is a Max - minimal set in (G, w), then S

is also a Max - minimal set in any $(P, \tilde{w})$.

Now, we are in a position to prove the main result of this section and the whole paper.

Theorem 3. S is a Max - minimal set in a path realizing the terminal Max - capacity matrix of (G, w) if and only if it is also a Max - minimal set in (G, w).

Proof. Let S be a Max - minimal set in $(P, \tilde{w})$. We can restrict our considerations to the case $|S| > 1$ only, since for $|S| = 1$ the assertion is obvious. By Proposition 2, the definition of Max - minimal set and the description of $(G, \bar{w})$, we get

$$\tilde{m}(S, X - S) = \max\{t_{x_{k-1}x_k}, t_{x_{k+|S|-1}x_{k+|S|}}\}, \qquad (7)$$

if only $S \cap \{x_1, x_n\} = \emptyset$, where for convenience, one of two possible orientations for the path P is assumed, and the subscripts assigned to vertices indicate the positions of vertices in P with respect to this orientation. According to Proposition 1, the value of $\tilde{m}(S, X - S)$ should be compared with $\tilde{m}(R, X - R)$'s for nonempty proper subsets R of S such that R's are also Max - minimal sets in $(P, \tilde{w})$. Therefore, using Proposition 2 again, we obtain

$$\tilde{m}(R, X - R) = \max\{t_{x_{i-1}x_i}, t_{x_jx_{j+1}}\} \qquad (8)$$

for the R's as mentioned above, where $i, j = k, k + 1, \ldots, k + |S| - 1$. Combining (7) and (8) with (5) for $Z = S$, we obtain

$$\max\{t_{x_{k-1}x_k}, t_{x_{k+|S|-1}x_{k+|S|}}\} < t_{x_{u-1}x_u}, \qquad (9)$$

for any $u = k + 1, k + 2, \ldots, k + |S| - 2$, since, in particular $|R| = 1$. The inequality (9) immediately implies (6) for each pair of distinct $x, z \in S$ and any $y \in X - S$. Thus, S is Max - minimal set in (G, w), due to Proposition 3.

Since S is a Max - minimal set in $(P, \tilde{w})$, then $|S| < |X|$, by definition. Therefore, $0 \leq |S \cap \{x_1, x_n\}| \leq 1$. If $|S \cap \{x_1, x_n\}| = 1$, then either $t_{x_{k-1}x_k}$ or $t_{x_{k+|S|-1}x_{k+|S|}}$ should be replaced by $\min\{t_{xy} : x, y \in X, x \neq y\}$ in (7) and (9), and the rest of arguing remains in force.

Thus, for accomplishing the whole proof it suffices to apply Proposition 4. Q. E. D.

The use of Theorem 3 instead of Proposition 1 gives a possibility to decrease the computational effort required for checking whether any set is a Max - minimal set in (G, w). Namely, the approach described in Section 6 of [10] yields the execution of $2(n - r) - 1$ comparisons to compute the value of $m(Q, X - Q)$ (see Steps 18 - 20 of SP in [10]) for a single Q, $|Q| = r > 1$, which consists of consecutive vertices in $(P, \tilde{w})$ and, preliminarily, $O(n^2)$ comparisons to obtain $m(\{x\}, X - \{x\})$'s (Step 3 of SP in [10]), in addition. In contrast, (7) and (8) imply that to calculate $\tilde{m}(Q, X - Q)$ (and, in particular, $\tilde{m}(\{x\}, X - \{x\})$) it suffices to perform only one such operation. That proves the practical importance of Theorem 3. In the new version of the algorithm for finding Max - minimal sets, which is described in Section 4 in details, we use some consequence of Theorem 3 and Propositions 1 and 2 instead of Theorem 3 itself. Namely

Corollary 2. S is a Max - minimal set in (G, w) if and only if either $S = \{x\}$, $x \in X$, or S is constituted by consecutive vertices in $(P, \tilde{w})$ and it is a union of pairwise disjoint Max - minimal sets in (G, w), say $S_1, S_2, \ldots, S_k$, such that the condition

$$\tilde{m}(Z, X - Z) < \min\{\tilde{m}(S_i, X - S_i) : i = 1, 2, \ldots, k\} \qquad (10)$$

is satisfied for $Z = S$ and does not hold for each $Z = \bigcup_{i \in J} S_i$, where $J \subset \{1, 2, \ldots, k\}$, $1 < |J| < k$, and the latter form of Z also consists of consecutive vertices in $(P, \tilde{w})$.

Finally, we mention a next feature of Max - minimal sets in (G, w) (see Lemma 2 in [8]), which can be easily extended for the case of Max - minimal sets in $(P, \tilde{w})$ by means of Theorem 3.

Proposition 5. Two Max - minimal sets in $(P, \tilde{w})$ are either disjoint or one of them includes the other.

Proposition 5 is used in Section 4 for showing the validity of the new algorithm.

4. The new algorithm

The idea of the new algorithm is similar to the previous one described in [10]. Namely, basing on any path realizing the terminal Max - capacity matrix for a given (G, w) the successive r - tuples, $r = 2, 3, \ldots, n - 1$, of consecutive vertices are examined whether they constitute a Max - minimal set. If the result of examination is positive, then, due to Proposition 5, the considered r - tuple is merged into a single vertex, the resultant self - loops are deleted, and the above mentioned way of searching is applied to the r - tuples of this updated path, etc. The performance begins for $r = 2$, and to satisfy the conditions given in Corollary 2 no $(r + 1)$ - tuples are considered before finishing the complete inspection of r - tuples.

Let us denote

$$h_u = t_{x_u x_{u+1}}, \qquad (11)$$

where the labelling of vertices is consistent with their ordering in a path realizing any terminal Max - capacity matrix, say in $(P, \tilde{w})$ (this assumption holds throughout this section), and

$$h_{max} = \max\{t_{xy} : x, y \in X, x \neq y\},$$

$$(12)$$

$$h_{min} = \min\{t_{xy} : x, y \in X, x \neq y\}.$$

Now, we consider the following procedure:

. Searching Max - minimal Sets (SMS)

1. Each $\{x\}$, $x \in X$, is a Max - minimal set in (G, w) (and in $(P, \tilde{w})$).
2. $h_0 := h_{min}$; $h_n := h_{min}$.
3. Set CARD(u) := 1, PRED(u) := $u - 1$, SUCC(u) := $u + 1$, NEXT(u) := $u + 1$ and $\alpha(u) := \max\{h_{u-1}, h_u\}$, for $u = 1, 2, \ldots, n = |X|$.
4. $r := 2$.
5. $u := 1$.
6. c := SUCC(u).
7. If $c = n + 1$, then go to Step 15.
8. $k := c$ + CARD(c); $j := 0$.

9. $h := h_c$; $\alpha := \alpha(u)$.
10. Set $b := \max(h_{PRED(u)}, h)$ and $\alpha(u) := \min(\alpha, \alpha(c))$.
11. Check whether $b < \alpha(u)$. If so (i.e. the considered set is a Max - minimal set in (G, w)), then pass to Step 17. Otherwise $\alpha(u) := \alpha$.
12. SUCC(u) := k.
13. If $j > 0$, then go to Step 21.
14. Set u := NEXT(u) and return to Step 6.
15. $r := r + 1$.
16. If $r < n$, then return to Step 5. Otherwise, STOP.
17. The considered $(k - u)$ - tuple $(x_u, x_{u+1}, \ldots, x_{k-1})$ is a Max - minimal set in (G, w).
18. Set $h_u := h$ and CARD(u) := $k - u$.
19. Update $j := j + 1$ and set $\alpha := h_{PRED(u)}$.
20. BEG(j) := u; $i := r$.
21. Set u := PRED(u).
22. If $u = 0$, then pass to Step 25.
23. $i := i - 1$.
24. If $i \neq 0$, then return to Step 10.
25. v := BEG(j).
26. PRED(k) := v; NEXT(v) := k.
27. If $u = 0$, the return to Step 5.
28. $i := r$.
29. NEXT(u) := v.
30. $i := i - 1$.
31. If $i \neq 0$, then pass to Step 34.
32. Set NEXT(u) := k.
33. If $k = n + 1$, then go to Step 15. Otherwise, pass to Step 14.
34. u := PRED(v).
35. NEXT(v) := u.
36. Set $v := u$ and return to Step 30.

Moreover, we assume that the SMS is applied to the $(P, \tilde{w})$. If for no r - tuple consisting of consecutive vertices in $(P, \tilde{w})$ the condition in Step 11 hold then, obviously, only preliminary Steps 1 - 4 and further Steps 5 - 16 in a loop are performed. It is evident that in such a case first all the pairs (x_i, x_{i+1}), for $i = 1, 2, \ldots, n - 1$, further all the triples (x_j, x_{j+1}, x_{j+2}), $j = 1, 2, \ldots, n - 2$, and, in general, all the r - tuples $(x_u, x_{u+1}, \ldots, x_{u+r-1})$, $u = 1$,

$2, \ldots, n - r + 1$, are inspected, $r = 2, 3, \ldots, n - 1$.

Let us now suppose that the currently handled r - tuple has the form $Z = \langle x_u, x_{u+1}, \ldots, x_{u+r-1} \rangle$, and that no visit to Step 17 has been previously performed. The value of $\alpha(u)$ corresponds to the right - hand side of (10), and b equals $\tilde{m}(Z, X - Z)$, due to (7). In other words, the condition checked in Step 11 is the same as in (10). Moreover, due to our supposition, no set like $\langle x_u, x_{u+1}, \ldots, x_{u+l-1} \rangle$, $l = 2, 3, \ldots, r - 1$, is a Max - minimal set. Thus, due to Corollary 2, the satisfaction or violation of the condition in Step 11 is equivalent to the fact that Z is or is not a Max - minimal set. The latter situation was considered above. Let us now pay our attention to the case in which Z is a Max - minimal set, indeed. According to Proposition 5 no union of any nonempty proper subset of Z and a subset of $X - Z$ constitute a Max - minimal set. Therefore, it seems that Z can be merged into a single vertex (consider Steps 17 - 20 and 25 - 36 for a constant $j \geq 1$) without omitting any Max - minimal set in the further performance of SMC. It implies that the next examined r - tuple, if exists, should begin with x_{u+r}, which is realized by Step 32 (see also the parts of Steps 2, 8 and 18 concerning CARD(u) and Step 6).

Let us now additionally suppose that $u > r$, we started to inspect the $(r + 1)$ - tuples, the currently handled $(r + 1)$ - tuple has the form $\langle x_{u-r}, x_{u-r+1}, \ldots, x_u \rangle$, and that no visit to Step 17 has been previously performed for $(r + 1)$ - tuples. This $\langle x_{u-r}, x_{u-r+1}, \ldots, x_u \rangle$ contains x_u, where x_u stands for the previously considered Z, i.e. in fact, this (pseudo) $(r + 1)$ - tuple corresponds to the $2r$ - tuple $\langle x_{u-r}, x_{u-r+1}, \ldots, x_u, x_{u+1}, \ldots, x_{u+r-1} \rangle$. It means that, e.g., all sets of type $Z_l = \langle x_{u-r+l}, x_{u-r+l+1}, \ldots, x_u, x_{u+1}, \ldots, x_{u+r-1} \rangle$, $l = 1, 2, \ldots, r - 1$, are not examined at all. To avoid this situation Steps 19 - 36 are introduced. For $j = 1$ they handle first Z_{r-1}, second $Z_{r-2}, \ldots$, and finally Z_1, since $Z_{r-1} \subset Z_{r-2} \subset \ldots, \subset Z_1$ (compare with Corollary 2 and Proposition 5). If any new Max - minimal set is found, say Z_l, then u is replaced by $u + r - l$, etc., and the modification realized by Steps 17 - 27 and 10 - 13 starts again and again until either the first vertex in $(P, \tilde{w})$ is backtracked (i.e. $u = 0$, see Step 22) or a sequence $\pi(s, r - 1) = (x_s, x_{s+1}, \ldots, x_{s+r-2})$ of $r - 1$ consecutive vertices in $(P, \tilde{w})$ is found such that they do not belong to a Max - minimal set and are the close predecessors of the last Max - minimal set, say $\bar{Z}$, constructed in this modification. Obviously, $\bar{Z} = \{x_{s+r-1}, x_{s+r}, \ldots, x_{u+r-1}\}$.

From Proposition 5 it follows that by applying the above mentioned modification, listing the Max - minimal sets obtained in it and, finally, by merging $\bar{Z}$ into a single vertex, we do not lose any Max - minimal set in $(P, \tilde{w})$. On the other hand, Corollary 2 requires further handling of sets whose contain $\bar{Z}$ and any part of $\pi(s, r - 1)$ in the similar order as described before. Namely, if we denote $Z_{l,q} = \{x_{s+l}, x_{s+l+1}, \ldots, x_{s+r-2}, x_{s+r-1}, x_{s+r}, \ldots, x_{u+r-1}, x_{u+r}, \ldots, x_{u+r+q}\}$, $q = 0, 1, \ldots, n - u - r$, and $l = 0, 1, \ldots, r - 2$, then first $Z_{r-2,0}, Z_{r-3,0}, \ldots, Z_{0,0}$ should be examined, further $Z_{r-2,1}, \ldots, Z_{0,1}$, etc. Such arrangement of inspection is realizing by reversing the NEXT(.)'s labels attached to the elements of $\pi(s, r - 1)$ (Steps 25 - 36) and by assigning to them the same successor SUCC(.) in Step 12. These remarks and the use of Theorem 3 lead to:

Theorem 4. If the sequence $(x_u : u = 1, 2, \ldots, n)$ gives the ordering of vertices in a path realizing the terminal Max - capacity matrix of (G, w), and h_u, h_{min}, h_{max} are as in (11) and (12), respectively, then the SMS finds all Max - minimal sets in (G, w).

Therefore, the new algorithm for finding Max - minimal sets can be defined as follows

1. Execute the STMC
2. Perform the MCG
3. Realize the PRTM1
4. Execute the SMS,

where PRTM1 denotes the following modification of the PRMT described in [10]:

21

Procedure PRTM1

1. Set $A := X$.
2. $p := 1$; $q = 0$; $f := 0$.
3. If $|A| = 1$, then go to Step 12.
4. Take any $x \in A$.
5. Set $t_A := \min\{t_{xu} : u \in A - \{x\}\}$.
6. If $f = 0$, then set $h_{min} := t_A$ and $h_{max} := t_A$.
7. $f := 1$.
8. Construct $H := \{y : t_{xy} = t_A, y \in A\}$.
9. $p := p + 1$.
10. $A_p := H$; $h_p := t_A$.
11. $A := A - H$ and return to Step 3.
12. $q := q + 1$.
13. Set $x_q := x$, where $\{x\} = A$.
14. If $q = n$, then FINISH the PRTM1.
15. Set $A := A_p$ and set $h_q := h_p$.
16. Modify $h_{max} := \max\{h_{max}, h_q\}$; modify $h_{min} := \min\{h_{min}, h_q\}$.
17. Update $p := p - 1$ and return to Step 3.

Obviously, the PRTM1 also produces a path realizing the terminal Max - capacity matrix and, additionally, evaluates h_u's, h_{min} and h_{max} as defined in (11) and (12), respectively (see Theorem 9 in [10]). Moreover, similarly as in Section 6 of [10], it can be proved that the PRTM1 is of type $O(n^2)$.

It remains to estimate the complexity of SMS, and further, of the whole new algorithm. The remarks preceding Theorem 4 yield that in the extreme case the SMS handles all the r - tuples, $r = 2, 3, \ldots, n - 1$, of consecutive elements in a path realizing the terminal Max - capacity matrix exactly once in Steps 4 - 16 and Steps 17 - 27 and 10 - 13 (the so - called modification consisting of Steps 17 - 27 and 10 - 13 avoids the disregard of some r - tuples only), i.e. $O(n^2)$ subsets of X. Each of Steps 4 - 27 requires a number of elementary operations independent of n and of r. Thus, their total numerical complexity is proportional to n^2. The iteration loop constituted by Steps 28 - 36 is performed for some specific r - tuples (see the so - called modification considered above Theorem 4) only and, moreover, exactly once

22

for any of these r - tuples, i.e. it is executed at most n^2 times. Each of Steps 28 - 36 is realized by a constant (independent of n and of r) number of elementary operations. Thus they need $O(n^2)$ such operations in total. The complexity of Steps 1 - 4 is, obviously, of type $O(n)$. Therefore, the whole SMS requires $O(n^2)$ elementary operations. Since the complexities of STMC, MCG and PRMT1 are at most of type $O(n^2)$, then the whole new algorithm for finding Max - minimal sets is of type $O(n^2)$.

References

1. Christofides N. Graph theory. An algorithmic approach. New York, Academic Press, 1975.
2. Dijkstra E. A note on two problems in connection with graphs. Numerische Mathematik, 1, (1959), 269 - 271.
3. Hu T. C. The maximum capacity route problem. Operations Research, 9, (1961) 6, 898 - 900.
4. Kevin V. , Whitney M. Algorithm 422 - Minimal spanning tree. Comm. of Association for Computing Machinery, 15, (1972) 4, 273 - 274.
5. Mayeda W. Graph theory. New York. John Wiley. 1972.
6. Pollack M. The maximum capacity through a network. Operations Research, 8, (1960) 5, 733 - 736.
7. Prim R. C. Shortest connection networks and some generalizations. Bell System Technical J., 36 (1957) 6, 1389 - 1401.
8. Stańczak W. An introduction to Max - minimal sets. Control and Cybernetics, 15 (1986) 1, 83 - 99.
9. Stańczak W. Max - minimal sets and cut - sets in edge weighted graphs. Biuletyn PIAP, 5 - 157 (1991), 5 - 36.
10. Stańczak W. A polynomial - type algorithm for finding Max - minimal sets. Biuletyn PIAP, 5 - 157 (1991), 37 - 68.

23