On finding a maximum flow in a network with special structure and some applications[1]

Alexander V. Karzanov

1. In this paper we consider certain classes of integral flow networks. One class is formed by so-called *simple networks*. Special cases are: networks used for solving the problem on "set representatives" by P. Hall, networks constructed for finding minimum cut sets (sets of nodes separating two distinguished nodes of a graph), and networks for solving the problem on distinct common representatives [1].

Another class is formed by the *unit-capacity* networks (i.e. with all arc capacities equal to one). In this class we will also consider the subclass formed by *layered* networks (the definition will be given later).

In this paper we apply to these networks the algorithm of finding a maximum flow due to E.A. Dinitz [2] and establish the running time bounds in these cases, namely: $Cp\sqrt{n}$ for simple networks, and $Cpn^{2/3}$ for unit-capacity networks in a general case, which will further be refined for layered networks (depending on their "lengths"). Hereinafter n is the number of nodes, p is the number of arcs of a network, and C is a constant (independent of n, m).

As an application, we give an algorithm of reducing a matrix to a block-triangular form (with the maximum possible number of blocks) by use of independent permutations on rows and columns. It is known that using such a form, one can decrease the running time of solving systems of linear equations of large size which are encountered in many problems in practice, see [3]. We will show that this task is reduced to solving a certain "set representatives" problem and then the problem of extracting the bi-components in a graph (the time bound for the former is $Cp\sqrt{n}$, and for the latter is Cp, where p is the number of nonzero elements of a matrix). This is especially efficient in the case of sparse matrices, i.e. when $p = o(n^2)$. Such matrices are often reduced to the block-triangular form with many blocks.

2. Let s and t be distinguished nodes in an (un)directed graph $G = (X, U)$. Following terminology in [1], call a subset of nodes $Z \subset X$ an s,t*-separated set* if in the subgraph obtained from G by removing Z the node s cannot be connected to t by an (un)directed path (note that if s is not connected by a path to t already in the graph G, then $\emptyset$ is an s,t-separated set).

An undirected graph can be turned into a directed one by replacing each edge xy by two arcs $\overrightarrow{xy}$ and $\overrightarrow{yx}$. Then the separated sets are not changed. In what follows we will deal with directed graphs only.

It is shown in [1] that the problem of finding a minimum (size) s,t-separated set one can reduce to the problem of finding a maximum flow in a certain flow network

[1]Author's translation (preserving the original style and notation as much as possible) from: А.В. Карзанов, О нахождении максимального потока в сетях специального вида и некоторых приложениях, В кн.: *Математические вопросы управления производством*, Вып. 5, изд-во МГУ, Москва, 1973, с. 81–94. (A.V. Karzanov, O nakhozhdenii maksimal'nogo potoka v setyakh spetsial'nogo vida i nekotorykh prilozheniyakh, In: *Matematicheskie Voprosy Upravleniya Proizvodstvom* (L.A. Lyusternik, ed.), Moscow State Univ. Press, Moscow, 1973, Issue 5, pp. 81–94, in Russian.)

$\widetilde{G}_{s,t}$ [2]. The node set $\widetilde{X}$ of $\widetilde{G}_{s,t}$ consists of the elements: (a) s'', (b) t', (c) x', x'' for all $x \in X$, $x \neq s, t$. The arc set $\widetilde{U}$ of $\widetilde{G}_{s,t}$ consists of: (a) the arcs $\overrightarrow{x'x''}$, and (b) the arcs $\overrightarrow{x''y'}$ for $\overrightarrow{xy} \in U$. The arcs of the form $\overrightarrow{x'x''}$ are called *split-node-arcs*.

We assign the capacity c to be equal to 1 for each split-node-arc, and to be ∞ for the other arcs.

Figure 1 illustrates the construction for an instance of $G_{s,t}$.

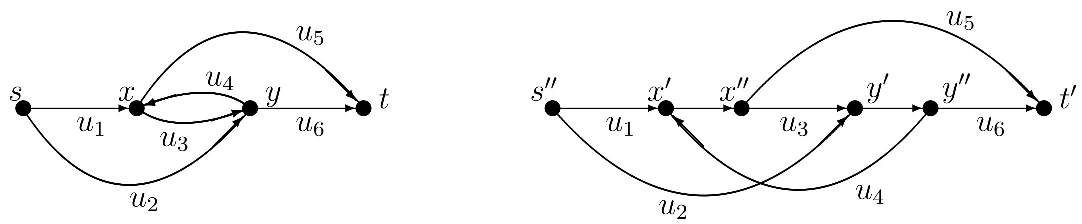

Рис. 1:

Let us say that a network $\Gamma_{s,t} = (Y, V)$ with integer capacities c is *simple* if for any node $y \in Y - \{s, t\}$, at least one of the following takes place: (a) y has at most one outgoing arc, and if $\overrightarrow{yz}$ is such an arc then $c(\overrightarrow{yz}) = 1$; or (b) y has at most one incoming arc, and if $\overrightarrow{xy}$ is such an arc then $c(\overrightarrow{xy}) = 1$. Such arcs $\overrightarrow{zy}$ and $\overrightarrow{xy}$ are called *critical.*

It is easy to see that the network $\widetilde{G}_{s,t}$ as above is simple and all split-node-arcs in it are critical.

The problem on “set representatives”, besides the classical formulation, admits the following three equivalent settings (cf. [1,4]):

A) In an $n \times n$ matrix M whose entries are 0 and 1, it is required to find a maximum number of entries 1 with no two entries contained in the same row or in the same column (the *matrix form*);

B) Given a bipartite (directed) graph $G = (V', V''; U)$ with $|V'| = |V''|$, find a maximum matching (the *graph form*);

C) Let $Q_{s,t} = (Z, W)$ be a flow network in which $Z - \{s, t\}$ is partitioned into two disjoint subsets Z_1 and Z_2 such that s is connected by outgoing arcs to all nodes in Z_1, t is connected by incoming arcs going from all nodes in Z_2, and the other arcs go from Z_1 to Z_2. All arc capacities are ones. It is required to find a maximum flow in $Q_{s,t}$ (the *flow network form*).

(The network $Q_{s,t}$ is simple since each node in Z_1 has a unique incoming arc and each node in Z_2 has a unique outgoing arc.)

Let us say that the problem is *perfectly solved* if the number of found entries (resp. the cardinality of a matching, the flow value) is equal to n. Otherwise we say that the problem is *imperfectly solved.*

[2]A graph $G_{s,t} = (X, U)$ is called a *flow network* if it there are given two distinguished nodes: a *source* s and a *sink* t, and a nonnegative function c on U (of arc *capacities*). A *flow* in $G_{s,t}$ is a function $f(u)$, $u \in U$, satisfying: (1) $\forall u \in U$: $0 \leq f(u) \leq c(u)$, and (2) $\forall x \in X - \{s, t\}$: $\sum_{\overrightarrow{xy} \in U} f(\overrightarrow{xy}) - \sum_{\overrightarrow{zx} \in U} f(\overrightarrow{zx}) = 0$. The number $\sum_{\overrightarrow{sy} \in U} f(\overrightarrow{sy}) - \sum_{\overrightarrow{zs} \in U} f(\overrightarrow{zs})$ is called the *value* of f. A *maximum flow* is a flow of the maximum possible value.

3. We start with briefly describing Dititz's algorithm for finding a maximum flow in an arbitrary flow network $G_{s,t} = (X, U)$ by using shortest augmenting paths subgraphs (or *manuals*, in terminology of [2]).

Let P be a flow in $G_{s,t}$. Construct the network $G_{s,t}(P)$ with the same nodes. If the flow in an arc $\overrightarrow{xy}$ of $G_{s,t}$ is equal to q, then we define the capacity of $\overrightarrow{xy}$ in $G_{s,t}(P)$ to be $c(\overrightarrow{xy}) - q$ (*operation I*). We throughout assume that if the capacity of an arc of a graph in question is equal to one, then this arc is automatically deleted from the graph. If $q > 0$, then we add to $G_{s,t}(P)$ the arc $\overrightarrow{yx}$ with the capacity $c(\overrightarrow{yx}) := q$ (*operation II*). Thereby the graph may become a multigraph. We call the created arc $\overrightarrow{yx}$ *reverse* to the "genuine" arc $\overrightarrow{xy}$. In the network $G_{s,t}(P)$ one extracts the subgraph S with the same nodes whose arcs are precisely those contained in shortest paths from s to t; this S is called the shortest augmenting paths subgraph in $G_{s,t}(P)$, or the *manual* (translating the Russian word справочная used in [2]). We seek for a flow ΔP in S such that subtracting ΔP from the capacities in S (and then deleting the zero capacity arcs) makes t disconnected from s; such a flow is called *blocking*. To find a blocking flow takes Cpn time in general, where $p := |U|$. Next the flow ΔP is added "algebraically" to P (i.e. to the flow q in a "genuine" arc one is added the new flow Δq in it, and from q one is subtracted the flow $\Delta q'$ in the reverse arc), after which the reverse arcs are deleted and we repeat the procedure (with the new current flow in $G_{s,t}$). One proves that the number of the number of iterations is at most n, whence the time bound of the whole algorithm is Cpn^2.

Next we show that a blocking flow in the shortest augmenting paths subgraph S of a simple network $\Gamma_{s,t} = (X, U)$ can be found in Cp time. Handling S according to [2] consists in constructing a path from s to t (taking $O(k)$ time, where k is the distance (i.e. minimum number of edges of a path) from s to t in S), followed by pushing the maximum flow along this path and making operation I (with deleting the saturated arcs). Also one deletes the arcs no longer contained in paths of length k from s to t (taking $O(\ell)$ time, where ℓ is the number of deleted arcs). Then one constructs a new path from s to t in the current manual, and so on. Observe that: (a) if the path (constructed at an iteration) contains a critical arc u, then u becomes saturated since it has the minimum possible (nonzero) capacity equal to 1, and (b) if $u = \overrightarrow{xy}$, $x \neq s$, is a non-critical arc contained in the path, then, by the definition of a simple network, the critical arc entering x must belong to this path as well, and after deleting the latter arc, the node x (and therefore $\overrightarrow{xy}$) becomes unreachable from s. Hence the total number of operations applied to an arbitrary arc in S is $O(1)$, implying the time bound Cp.

Lemma 1. *Let P be an integer flow in a simple network $G_{s,t}$. Then a path-flow decomposition*[3] *of P consists of paths from s to t, any two having no common intermediate nodes.*

Proof. It obviously follows from the definition of a simple network.

Lemma 2. *Let $\Gamma_{s,t}$ be a simple network, and P an integer flow in it. Then the network $\Gamma_{s,t}(P)$ is simple as well.*

[3] A flow P_1 is called a *path-flow* if the arcs where P_1 is nonzero form a path from s to t, and the value of P_1 is equal to the minimum capacity of these arcs. A *path-flow decomposition* of a flow is its representation as the sum of path-flows (which need not be unique in general), cf. [1].

Proof. For any $x \in X - \{s,t\}$, the following cases are possible:

1) P does not pass x (i.e. P is zero on the arcs incident to x). Then x and its incident arcs preserve in $\Gamma_{s,t}(P)$.

2) x satisfies condition (a) in the definition of simple networks, and P passes x. Then P contains the critical arc of the form $\overrightarrow{xy}$ and, therefore, some arc $\overrightarrow{zx}$ (by the integrality of P, $\overrightarrow{zx}$ is unique and the flow in it is equal to 1).

In $\Gamma_{s,t}(P)$ there appears the new arc $\overrightarrow{yx}$ with $c(\overrightarrow{yx}) = 1$ and the new arc $\overrightarrow{xz}$ with $c(\overrightarrow{xz}) = 1$. (See Fig. 2.) Therefore, x in $\Gamma_{s,t}(P)$ continues to satisfy condition (a) in the definition of simple networks.

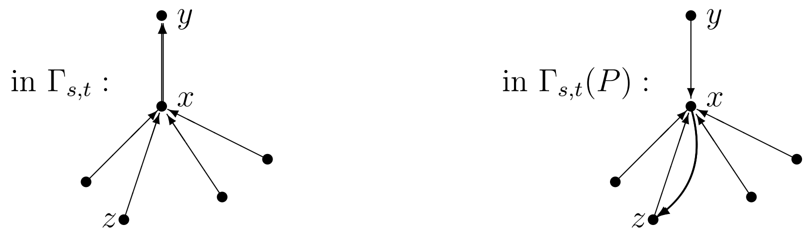

Рис. 2:

3) P passes x, and x in $\Gamma_{s,t}$ satisfies condition (b) in the definition of simple networks. Similar to the previous case, one shows that condition (b) continues to hold for x in $\Gamma_{s,t}(P)$.

The lemma is proven. ∎

Theorem 1. *For a simple network $\Gamma_{s,t}$, the number of shortest augmenting paths subgraphs constructed during the algorithm does not exceed $2\sqrt{n}$.*

Proof. Let k manuals $S_1, \ldots, S_k$ have already been constructed during the algorithm and a flow M_k has been found in $\Gamma_{s,t}$. Denote the maximum flow value in $\Gamma_{s,t}$ by m. Then the value of a maximum flow $\overline{M}_k$ in $\Gamma_{s,t}(M_k)$ is equal to $m - |M_k|$, where $|M|$ denotes the value of a flow M. We establish a relation between the length $\ell(S_{k+1})$ of the next manual S_{k+1} and the value $|\overline{M}_k|$ (where the *length* $\ell(S_i)$ is the distance from s to t in $\Gamma_{s,t}(M_{i-1})$, or in S_i).

Let us refer to the set O^r_{k+1} of nodes in $\Gamma_{s,t}(M_k)$ being at distance r from s as a *layer* (then t lies in the layer $O^{\ell(S_{k+1})}_{k+1}$).

Lemma 3. *$|\overline{M}_k|$ does not exceed the number of nodes in any layer O^r_{k+1}, $1 \le r < \ell(S_{k+1})$.*

Proof. By Lemma 2, the network $\Gamma_{s,t}(M_k)$ is simple. Therefore, by Lemma 1, a (unique) path-flow decomposition of the flow $\overline{M}_k$ in $\Gamma_{s,t}(M_k)$ consists of paths without common intermediate nodes. Since any path from s to t necessarily meets each layer O^r_{k+1}, $1 \le r < \ell(S_{k+1})$, one has $|O^r_{k+1}| \ge |\overline{M}_k|$, as required. ∎

The inequalities $|O^r_{k+1}| \ge |\overline{M}_k|$, $1 \le r < \ell(S_{k+1})$, and $\sum_{r=1}^{\ell(S_{k+1})-1} |O^r_{k+1}| < n$ imply

$$(\ell(S_{k+1}) - 1)|\overline{M}_k| < n. \tag{1}$$

Using this relation, we now easily obtain Theorem 1. Indeed, the lengths $\ell(S_i)$ form a monotone increasing sequence, hence the number of manuals of length is at most

$\sqrt{n}$ does not exceed $\sqrt{n}$. And for the remaining manual S_i, we obtain from (1) that $|\overline{M}_i| < \sqrt{n}$. Obviously, the values $|M_i|$ are monotone decreasing, hence the number of the remaining manuals is less than $\sqrt{n}$. Thus, the total number of manuals does not exceed $2\sqrt{n}$, as required in the theorem. ∎

Therefore, a maximum flow in a simple network can be found in $Cp\sqrt{n}$ time. Note that the time bound can be written more carefully as $Cp\min\{2\sqrt{n}, m\}$ (where m is the maximum flow value).

A minimum separated set in a graph G (i.e. a subset of nodes whose removal makes the graph disconnected) can be found in $Cp\overline{p}\sqrt{n}$ time, where $\overline{p}$ is the number of edges of the complementary graph $\overline{G}$ (this can be done by finding a minimum x,y-separated set for each pair x,y of nodes of G connected by an edge in $\overline{G}$, since if x and y are not adjacent in $\overline{G}$, then no x,y-separated set exists). In a general case, with $p = O(n^2)$ and $\overline{p} = O(n^2)$, one obtains the bound $Cn^{4.5}$.

Remark. In [6] the author proved that Dinitz' algorithm applied to the problem on "set representatives" has the time bound $Cp\sqrt{n}$ (the proof there, based on similar ideas, does not consider a general simple network). Also it is shown there that the running time in the worst case is $Cp\sqrt{n}$ as well. Hence our time bound for this algorithm applied to simple networks is exact.

4. Let $G_{s,t}$ be a network in which the capacities of all arcs are ones (a *unit-capacity* network).

Let the distance from s to t in the network is equal to $\ell > 2$. For convenience we also introduce the number $\ell' := \ell - 2$.

Denote by O^i the set of nodes at distance i from s, and let $s_i := |O^i|$, $i = 1, \dots, \ell'+1$. Define $s'_i := (s_i + s_{i+1})/2$ and $s' := \min\{s'_i \colon i = 1, \dots, \ell'\}$.

The maximum flow value m in $G_{s,t}$ satisfies

$$m < n/2, \tag{2}$$

since $m \le s_1$ and $m \le s_{\ell'+1}$; and satisfies

$$m \le {s'}^2 \tag{3}$$

since $m < s_i s_{i+1} \le {s'_i}^2$, $i = 1, \dots, \ell'$.

Also $\sum_{i=1}^{\ell'} s'_i = \frac{s_1}{2} + s_2 + \ldots + s_{\ell'} + \frac{s_{\ell'+1}}{2} < n$, whence

$$\ell' s' < n. \tag{4}$$

Combining (3) and (4), we obtain

$$m < n^2/{\ell'}^2. \tag{5}$$

Let in the process of finding a maximum flow in $G_{s,t}$ there have already been constructed $\lfloor n^{2/3} \rfloor$ shortest augmenting paths subgraphs. Then $\ell(S_{\lfloor n^{2/3} \rfloor + 1}) > n^{2/3}$, and applying to $G_{s,t}(M_{\lfloor n^{2/3} \rfloor})$ inequality (5), we obtain that $|\overline{M_{\lfloor n^{2/3} \rfloor}}| < n^2/n^{2(2/3)} = n^{2/3}$. This implies that the total number of manuals is less than $2n^{2/3}$, yielding time bound $Cpn^{2/3}$ for the unit-capacity networks.

Next, let us say that a network $G_{s,t}$ is *layered* if all simple paths from s to t have one and the same length ℓ. W.l.o.g., one may assume that such a network is acyclic and that s and t are unique zero indegree and unique zero outdegree nodes in it, respectively.

Lemma 3. *For a layered network $G_{s,t}$, any chain*[4] *of length r from s to t contains $\frac{r-\ell}{2}$ backward arcs.*

Proof. Consider an arbitrary chain from s. One easily shows by induction on its length that if the chain terminates in a layer O^i, then the number of forward arcs is greater by i than the number of backward arcs in it. When $i = \ell$, we obtain the required assertion. ∎

Let $G_{s,t}$ be a layered unit-capacity network. A path-flow decomposition of a maximum flow $\overline{M}_k$ in a network $G_{s,t}(M_k)$ consists of paths of length at least $\ell(S_{k+1})$. In each of these paths ξ, the reverse arcs (defined as in part 3) correspond to the backward arcs of the chain in $G_{s,t}$ related to ξ. Therefore, by Lemma 4, ξ contains at least $(\ell(S_{k+1}) - \ell)/2$ reverse arcs. Since the number of such paths is equal to $\overline{M}_k$ and the total number of reverse arcs in $G_{s,t}(M_k)$ is equal to $|M_k|\,\ell$, we obtain

$$|\overline{M}_k|\,(\ell(S_{k+1}) - \ell)/2 < |M_k|\,\ell,$$

or

$$(\ell(S_{k+1}) - \ell)\,|\overline{M}_k| < 2\ell m. \tag{6}$$

Let A be the set of manuals S_i for which $\ell(S_i) - \ell \le \sqrt{2\ell m}$. Since $\ell(S_1) = \ell$, we have $|A| \le \sqrt{2\ell m}$.

For the set B of remaining manuals S_k, one holds $|\overline{M}_k| < \sqrt{2\ell m}$ (by (6)), whence $|B| < \sqrt{2\ell m}$. Therefore,

$$|A| + |B| < 2\sqrt{2\ell m}.$$

Using (2) and (5), we obtain $|A| + |B| < C\sqrt{\ell n}$ and $|A| + |B| < C\frac{n}{\sqrt{\ell}}$ (where C is a constant). Also $|A| + |B| \le m < n^2/\ell'^2$.

Finally (combining the above bounds),

$$|A| + |B| < C \min\{\sqrt{\ell n},\ \frac{n}{\sqrt{\ell}},\ \frac{n^2}{\ell^2},\ n^{2/3}\}. \tag{7}$$

Let us examine (7) in two extremal cases of layered unit-capacity networks.

1) If $\ell = O(1)$, then $|A| + |B| < C\sqrt{n}$, and the time bound is $Cp\sqrt{n}$.

2) If $\ell = O(n)$, then $|A| + |B| = O(1)$, and the time bound is Cp.

5. A numerical matrix $A = (a_{ij})$ of size $n \times n$ is called *block-triangular* if there is a tuple of natural numbers $0 < k_1 < k_2 < \ldots < k_r = n$ with $r \ge 2$ such that for any ℓ, i, j with $i < k_\ell$ and $j > k_\ell$, one holds $a_{ij} = 0$ (see Fig. 3). The number r is called the *length* of the tuple (obviously, given a block-triangular matrix, it makes sense to consider the tuple of maximum length, which is unique). A matrix A is called *reducible*

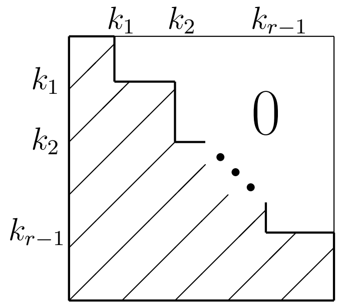

Рис. 3:

(to the block-triangular form) if it can be turned into a block-triangular one by use of some (independent) permutations of rows and columns.

Instead of A, one can consider the matrix $M = (m_{ij})_n^n$ where $m_{ij} = 1$ if $a_{ij} \neq 0$, and $m_{ij} = 0$ if $a_{ij} = 0$ (M is called the *sign-matrix* of A).

Denote the set of rows of M by Φ, and the set of columns by X.

For $\Phi' \subseteq \Phi$, let $\mathcal{D}(\Phi')$ denote the set of columns having a nonzero entry in some row from Φ'. If there exists $\Phi' \subset \Phi$ such that $|\Phi'| > |\mathcal{D}(\Phi')|$, then the matrix M (as well as A) is singular. In what follows we always consider matrices M not contained a row subset Φ' with this property, and refer to such an M as *consistent* (in the Russian original the term *non-degenerate* is used).

Let us say that a subset $C \subseteq \Phi$ is a *quasi-block* if $|C| = |\mathcal{D}(C)|$. The sets $\emptyset$ and Φ are regarded as *non-proper* quasi-blocks. Then the following takes place.

Lemma 5. If C_1 and C_2 are quasi-blocks, then $C' := C_1 \cap C_2$ and $C'' := C_1 \cup C_2$ are quasi-blocks as well, i.e. the set of quasi-blocks forms a lattice.[5]

Proof. It is easy to see that $\mathcal{D}(C') \subseteq \mathcal{D}(C_1) \cap \mathcal{D}(C_2)$. If $|C'| < |\mathcal{D}(C_1) \cap \mathcal{D}(C_2)|$, then $|C''| = |C_1| + |C_2| - |C'| > |\mathcal{D}(C_1)| + |\mathcal{D}(C_2)| - |\mathcal{D}(C_1) \cap \mathcal{D}(C_2)| = |\mathcal{D}(C_1 \cup C_2)| = |\mathcal{D}(C'')|$, i.e. the matrix M is non-consistent.

If $|C'| > |\mathcal{D}(C')|$, then M is non-consistent as well. Hence $|C'| = |\mathcal{D}(C')| = |\mathcal{D}(C_1) \cap \mathcal{D}(C_2)|$ and $|C''| = |\mathcal{D}(C'')|$, as required. ▮

We call a subset $B \subset \Phi$ a *block* if each quasi-block either entirely contains B or is disjoint from B, and B is inclusion-wise maximal under this property.

On the set of blocks one can introduce a comparison relation. We write $B_1 \prec B_2$ if for any quasi-block C, $B_2 \subseteq C$ implies $B_1 \subseteq C$. One easily proves that $\prec$ gives a partial order on the set of blocks.

We say that a column $x \in \mathcal{D}(B)$ is *proper* for a block B if $B' \prec B$ implies $x \notin B'$. The set of proper columns for B is denoted by $S(B)$.

Theorem 2. 1) *For any block B, $|B| = |S(B)|$.* 2) *If a block $\widetilde{B}$ is such that $\mathcal{D}(\widetilde{B}) \cap S(B) \neq \emptyset$, then $B \prec \widetilde{B}$.*

---

[4]A *chain* is an sequence $x_0, u_0, x_1, \dots, u_{q-1}, x_q$ where each u_i is an arc connecting nodes x_i and x_{i+1}. The number q is the *length* of the chain. An arc u_i is called *forward* if $u_i = \overrightarrow{x_i x_{i+1}}$ and *backward* if $u_i = \overleftarrow{x_i x_{i+1}}$.

[5]This lemma can be obtained as a corollary from a theorem on the intersection and union of critical sets in a bipartite graph, see [4].

Proof. Denote by $C(B)$ the minimal quasi-block containing B. Then $B' \subset C(B)$ and $B' \neq B$ hold if and only if $B' \prec B$ (where B' is a block). Obviously, if $B' \prec B$ and $B'' \subset C(B')$, then $B'' \prec B$. Let $\overline{C}(B) := \cup(C(B') \colon B' \prec B)$. Then $\overline{C}(B)$ is the quasi-block that is the union blocks smaller than B (by $\prec$). Therefore, $B = C(B) - \overline{C}(B)$. This implies $S(B) = \mathcal{D}(C(B)) - \mathcal{D}(\overline{C}(B))$ and $|S(B)| = |\mathcal{D}(C(B))| - |\mathcal{D}(\overline{C}(B))| = |C(B)| - |\overline{C}(B)| = |B|$.

Next we prove the second assertion. Suppose $B \not\prec \widetilde{B}$. Then $B \not\subset C(\widetilde{B})$. Take $C' := C(\widetilde{B}) \cup \overline{C}(B)$. Obviously, $\mathcal{D}(C' \cup C(B)) - \mathcal{D}(C') \subseteq S(B)$. This inclusion is strict since $\mathcal{D}(\widetilde{B}) \cap S(B) \neq \emptyset$. But this leads to a contradiction with $|S(B)| = |B| = |C' \cup C(B)| - |C'| = |\mathcal{D}(C' \cup C(B)| - |\mathcal{D}(C')|$.

The theorem is proven. ▮

Corollary. Let ξ be a linear extension of the partial order on the set $\mathcal{B}$ of blocks. Then one can permute rows and columns of of M (or A) so as to obtain a block-triangular matrix with the tuple length $|\mathcal{B}|$.

Proof. The desired permutations can be constructed by induction. Let $B_1, \dots, B_r$ be the sequence of blocks in ξ. We dispose the minimal block B_1 in ξ (containing $|B_1|$ rows and columns) into the upper-left corner of the constructed matrix. Let i blocks $B_1, \dots, B_i$ have been already disposed so that they occupy k_i upper rows and the columns of $\mathcal{D}(B_j)$, $j = 1, \dots, i$, becomes k_i left columns. Take the next block B_{i+1} and permute its rows into the positions from $k_i + 1$ to $k_i + |B_{i+1}|$. From Theorem 2 it follows that $S(B_i) \subseteq \{x_{k_i+1}, \dots, x_n\}$ and $|S(B_i)| = |B_i|$. Permute the columns in $S(B_i)$ into the positions from $k_i + 1$ to $k_i + |B_{i+1}|$.

Also it is easy to prove a converse assertion: let M be transformed by some permutations into a block-triangular matrix with a tuple $k_1, \dots, k_r$ of maximum length r. Then the corresponding sets B_i of rows, $i = 1, \dots, r$, constitute blocks, and their sequence (from the top to the bottom) gives a linear extension of the partial order on the blocks.

Finally, we describe an algorithm of finding the blocks of a matrix M (when it is irreducible, the algorithm outputs one "block" consisting of the entire Φ).

First we solve the problem on "set representatives" given by M. When M is consistent, the problem is perfectly solved (and vice versa). In other words, in the corresponding bipartite graph $G_M = (\Phi, X; U)$ we construct a matching p from Φ to X (regarding p as a bijective mapping).

Theorem 3. *For any block B, $p(B) = S(B)$.*

Proof. For any quasi-block C, it is obvious that $p(C) = \mathcal{D}(C)$. Then $B = C(B) - \overline{C}(B)$ and $S(B) = \mathcal{D}(C(B)) - \mathcal{D}(\overline{C}(B))$ imply $p(B) = S(B)$. ▮

Let us identify the nodes in each pair involved in p (and delete the loops appeared). The obtained graph with n nodes is denoted by $G'_M = (X', U')$. If a row φ and a column x are identified, the corresponding node of G'_M is denoted by (φx).

Theorem 4. *Let $L_1, \dots, L_k$ be the bi-components of G'_M, and $H(G'_M)$ the corresponding factor-graph.*[6] *Then:* 1) *if L_i is induced by nodes $(\varphi_{i_1} x_{p(i_1)}), \dots,$*

[6]A *bi-component* in a digraph is a strongly connected component. The *factor-graph* is the acyclic graph obtained by shrinking each bi-component and then identifying parallel arcs, see, e.g., [5].

$(\varphi_{i_r} x_{p(i_r)})$, *then* $\varphi_{i_1}, \dots, \varphi_{i_r}$ *form a block; and* 2) $H(G'_M)$ *determines the partial order on the blocks.*

Proof. Let $O = \{L_\alpha \colon \alpha \in \mathcal{A}\}$ be a set of bi-components with the property of "openness": if $L_\alpha \in O$ and $L_{\alpha'} \prec L_\alpha$, then $L_{\alpha'} \in O$. In other words, the set $X(O) = \{(\varphi_\beta x_{p(\beta)}) : \beta \in \mathcal{B}\}$ of nodes contained in the bi-components in O has no outgoing arc to $X' - X(O)$. Then, obviously, the set $\{\varphi_\beta \colon \beta \in \mathcal{B}\}$ is a quasi-block. It is easy to see that a converse takes place as well: if C is a quasi-block, then the set $X_C := \{(\varphi\, p(\varphi)) : \varphi \in C\}$ has no outgoing arcs to $X' - X_C$, and therefore it generates an "open" set of bi-components.

From this, in view of the fact that any bi-component can be represented as the difference of two "open" sets, the first assertion in the theorem follows.

The second assertion follows from the observation that the above correspondence maintains the structure of "open" sets (thinking of a quasi-block as forming an "open" subset in the set of blocks). ∎

The above theorems give rise to the following algorithm of transforming a matrix A (or M) into a block-triangular one.

1) Solve the problem on "set representatives" for M. If it is not perfectly solved, then M is not consistent, and the algorithm terminates.

2) In the graph $G_M = (\Phi, X; U)$, identify the pairs of nodes in the obtained matching, forming the graph G'_M.

3) Extract the bi-components in G'_M and construct the factor-graph $H(G'_M)$.

4) Find a linear extension $(L_{i_1}, \dots, L_{i_r})$ of the partial order on the bi-components (or on the nodes of $H(G'_M)$).

5) Make a permutation of rows and columns of M according to this linear order (where the rows and columns related to one and the same bi-component go in succession, in an arbitrary order).

If the number of non-zero elements in A is p, then the time bounds are: $Cp\sqrt{n}$ in step 1; Cn in step 2; Cp in step 3, see [7] (also there is a simpler algorithm due to Faradzhev [**?**] with the bound $C(p + n \log n)$); Cn in step 5. Thus, the running time is estimated by the time bound of the algorithm for solving the problem on "set representatives", i.e. by $Cp\sqrt{n}$. Note that for the correctness of our bounds one should that the matrix A be given by a list of its nonzero entries (or that the graph G_M be given). Otherwise, one needs to spend Cn^2 additional operations.

Remark. The above algorithm can be modified, preserving the same bound, for finding the structure of critical sets in a bipartite graph with a positive deficit (cf. [4]), or "blocks" of a degenerate matrix A.

References:

[1] L. Ford and D. Fulkerson, *Flows in Networks*, Mir, Moscow, 1966 (Russian transl.).

[2] E.A. Dinic, An algorithm for solution of a problem of maximum flow in a network with power estimation, *Doklady Akademii Nauk SSSR*, vol. 194, No. 4, 1970, in Russian.

[3] E.B. Ershov, On finding and using structural features of matrices in problems of planning, *Ekonomika i Matematicheskie Metody*, vol. 2, iss. 2, 1966, in Russian.

[4] O. Ore, *Graph Theory*, Nauka, Moscow, 1968 (Russian transl.).

[5] A.A.Zykov, *Theory of Finite Graphs*, vol. 1, Nauka, Novosibirsk, 1969, in Russian.

[6] A.V. Karzanov, Tochnaya otsenka algoritma nakhozhdeniya maksimal'nogo potoka, primenennogo k zadache "o predstavitelyakh", In: *Voprosy Kibernetiki. Trudy Seminara po Kombinatornoi Matematike (Moscow, 1971)*, Sovetskoe Radio, Moscow, 1973, pp.66-70, in Russian.

[7] A.V. Karzanov, Ekonomnyi algoritm nakhozhdeniya bikomponent grafa, In: *Trudy 3-i Zimnei Shkoly po Matematicheskomu Programmirovaniyu i Smezhnym Voprosam* (Drogobych, 1970), Issue 2, Moscow Inst. for Construction Engineering (MISI) Press, Moscow, 1970, pp.343-347, in Russian.

[8] I.A. Faradzhev, Efficient algorithms for solving certain problems on directed graphs, *Zhurnal Vichislitelnoi Matematiki i Matematichkoi Fiziki*, vol. 10, No. 4, 1970, in Russian.