IOP Conf. Series: Journal of Physics: Conf. Series **788** (2017) 012057 doi:10.1088/1742-6596/788/1/012057

One effective algorithm for finding a minimum cut of any transport network

S V Petrov[1], A N Snezhin[2] and V M Prostokishin[1,2]

[1] National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), 31 Kashirskoe shosse, 115409 Moscow, Russia

[2] Gazprom VNIIGAZ LLC, box 130, 115583 Moscow, Russia

E-mail: VMProstokishin@mephi.ru

Abstract. Finding the minimal cut on the directed graph is an important task that has not yet been solved in a general form. The solution of this problem allows one to find "the pipeline bottleneck" of the transport network and manage changing flows efficiently. In this article, authors discussed a new algorithm for solving this problem, built on the basis of consideration of the corresponding dual problem.

Introduction

Transport networks are one of the bases part of the fuel and energy complexes. An important problem is the localization of "the pipeline bottleneck" of the transport network also. A directed graph with varying flows and possible inflows and outflows of substance at the vertices and at the edges of the graph is the model of the network. This article describes an algorithm for the solution of the generalized problem (assuming constant inflows and outflows in arbitrary network nodes) finding the minimal cut and determine the maximum flow through the network.

1. Problem statement

Suppose to each edge ij of the graph assign the positive number q_{ij}. Any such number q_{ij} is called the capacity constraint on the edge. Consider two vertices s and t over from the graph. Let s be the source, t be the sink. Such graph, with some additional conditions, is called the network.

Assume that to each edge ij of the graph assign the non-negative number x_{ij}; then a set $\{x_{ij}\}$ is called the flow from s into t if the following conditions hold:

$$\sum_i x_{ij} - \sum_k x_{jk} = \begin{cases} -\nu, & \text{if } j = s, \\ 0, & \text{if } j \neq s, t, \\ \nu, & \text{if } j = t, \end{cases} \tag{I}$$

$$\nu \geqslant 0, \tag{II}$$

$$0 \leqslant x_{ij} \leqslant q_{ij}, \text{ for all } i, j. \tag{III}$$

By ν denote the maximum flow of the network. Then the maximum flow problem is of the form:

$$\nu \to \max\,. \tag{IV}$$

Dual problem: To find "the pipeline bottleneck" of the network such that the value of a flow in the network can be less than or equal to the value $V = \max \nu$.

The cut is the minimal cut if the capacity constraint of it is minimal over from all them.

Let $S = \{s_1, s_2, \ldots, s_r\}$ be a nonempty set of the numbers of the vertices, where there exist sources, $T = \{t_1, t_2, \ldots, t_u\}$ be a nonempty set of the numbers of the vertices, where there exist sinks.

Note that any vertex can have either a source or a sink only. Also, it can have neither a source nor a sink. By assumption, there exist one source and one sink in the network. It means that $S \neq \emptyset,\ T \neq \emptyset,\ S \cap T = \emptyset$. Then for amount of the sources and the sinks the following conditions hold: $2 \leqslant r + u \leqslant m,\ 1 \leqslant r \leqslant m,\ 1 \leqslant u \leqslant m$.

Finite inflows and outflows onto the edges (also into the vertices) of the network are added in this work as distinguished from the problems discussed in [1].

Let the transport network be represented with a corresponding graph. Then the problem has the following form:

$$x_{T(sj)} + \sum_i x_{T(ij)} - \sum_k x_{T(jk)} = f_j - \sum_i g_{T(ij)},\ j = s_1, s_2, \ldots, s_r, \tag{1}$$

$$\sum_i x_{T(ij)} - \sum_k x_{T(jk)} = f_j - \sum_i g_{T(ij)},\ j \neq s_1, s_2, \ldots, s_r, t_1, t_2, \ldots, t_u, \tag{2}$$

$$\sum_i x_{T(ij)} - \sum_k x_{T(jk)} - x_{T(jt)} = f_j - \sum_i g_{T(ij)},\ j = t_1, t_2, \ldots, t_u, \tag{3}$$

$$x_{T(ij)} \leqslant q_{T(ij)}, \tag{4}$$

$$- x_{T(ij)} \leqslant g_{T(ij)}, \tag{5}$$

$$x_{T(ij)} \geqslant 0,\ x_{T(sj)} \geqslant 0,\ x_{T(jt)} \geqslant 0, \tag{6}$$

$$\nu = \sum_{i=1}^{r} x_{n+i} = x_{n+1} + x_{n+2} + \cdots + x_{n+r} \to \max, \tag{7}$$

where m is the number of all vertices of the graph, n is the number of all edges, x_{ij} is a variable of the flow starting from the i-th vertex and ending in the j-th vertex, q_{ij} is the capacity constraint of the network's edge, g_{ij} is the inflow onto the edge, and f_i is the inflow into the i-th vertex.

Here, a transition is made from numbering of the variables and constants with two indeces towards numbering them with one index, preserving a bijection between them [2]. This bijection is denoted by T and it has the following form:

$$\begin{aligned}
&T : ij \leftrightarrow l; \quad i, j = 1, 2, \ldots, m;\ l = 1, 2, \ldots, n,\\
&T : \nu \leftrightarrow x_0, \quad T : x_{ij} \leftrightarrow x_l, \quad T : q_{ij} \leftrightarrow q_l, \quad T : c_{ij} \leftrightarrow c_l,\\
&T : ss_i \to n + i,\ i = 1, 2, \ldots, r, \quad T : tt_k \to n + r + k,\ k = 1, 2, \ldots, u.
\end{aligned}$$

where (i, j) is an index of the coordinate of a nonzero element in the adjacency-matrix representation, $l = 1, 2, \ldots, n,\ T(z_{ij}) = z_{T(ij)} = z_l$, and z is either a variable or a given value.

The matrix form of the system of equations and inequalities (1)-(7) is

$$F(x) = \sum_q c_q x_q \to \max, \tag{8}$$
$$A'x = d, \tag{9}$$
$$A''x \leqslant q, \tag{10}$$
$$A'''x \leqslant g, \tag{11}$$
$$x \geqslant 0, \tag{12}$$

where

$$d_j = f_j - \sum_i g_{T(ij)},\ j = 1, 2, \ldots, m. \tag{13}$$

We must substitute two inequalities for the equation (9)

$$A'x \leqslant d, \tag{14}$$
$$A'_{-}x \leqslant d_{-}, \tag{15}$$

where $A'_{-} = -A',\ d_{-} = -d_{-}$.

Thus, we obtain the standard form of the linear-programming problem. Let a solution our problem be feasible. Here, we take inverse bijection T^{-1} and can receive the values of the flows over all edges.

2. Problem of finding a minimal cut

Dual problem. The aim of this section is to solve the following problem: to have got edges of a minimal cut such that it should separate the sources S and the sinks T. Also, the following conditions hold: the constant inflows and outflows in vertices and over edges are balanced.

Now, we take the primal linear-programming problem having the standard form

$$\max \langle x, c \rangle,$$
$$Ax \leqslant b,$$
$$x \geqslant 0.$$

It follows in the standard way that the dual linear-programming problem is

$$\min \langle b^{tr}, y \rangle, \tag{16}$$
$$A^{tr}y \geqslant c^{tr}, \tag{17}$$
$$y \geqslant 0, \tag{18}$$

where $y = (y_1, y_2, \ldots, y_{2\cdot(m+n)})^{tr}$ are dual variables.

It can easily be checked (see [3]) that the standard form of the problem (16 – 18) is

$$\max \langle -b^{tr}, y \rangle, \tag{19}$$
$$-A^{tr}y \leqslant -c^{tr}, \tag{20}$$
$$y \geqslant 0. \tag{21}$$

Note that, the formula $\min \langle b^{tr}, y \rangle = -\max \langle -b^{tr}, y \rangle$ is correct.

We stress that the components of formulae $(19-21)$ are

$$b^{tr} = (d^{tr}, d_{-}^{tr}, q^{tr}, g^{tr}) = (d_1, \ldots, d_m, -d_1, \ldots, -d_m, q_1, \ldots, q_n, g_1, \ldots, g_n), \tag{22}$$

$$A^{tr} = ((A')^{tr}(A'_{-})^{tr}(A'')^{tr}(A''')^{tr}) = ((A')^{tr} \; -(A')^{tr}(A'')^{tr}(A''')^{tr}), \tag{23}$$

$$-c = (0, \cdots, 0, -1_{n+1}, -1_{n+2}, \cdots, -1_{n+r}, 0, \cdots, 0), \tag{24}$$

where the matrix A^{tr} has a size of $(n+r+u) \times (2 \cdot (m+n))$, the row vector b^{tr} has a dimension $1 \times (2 \cdot (m+n))$, the column vector c^{tr} has a dimension $(n+r+u) \times 1$, and the vector of dual variables $y = (y_1, y_2, \ldots, y_{2\cdot(m+n)})^{tr}$ has a dimension $(m+n) \times 1$.

The coordinates of the column vector b^{tr} is of the form (22). In this way, in our numbering, the set the edges of the network takes the coordinates y_i, $(i = 2\cdot m+1, \ldots 2\cdot m+n)$ of the vector y. The other coordinates correspond to the terms of a distribution of the flow at the vertices or the edges of the network.

Let a column vector $\hat{y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{2\cdot(m+n)})^{tr}$ be a solution of our problem $(16-18)$. Consider the column vector

$$\tilde{y} = (0, \ldots, 0, \tilde{y}_{2\cdot m+1}, \ldots, \tilde{y}_{2\cdot m+n}, 0, \ldots, 0)^{tr} = (0, \ldots, 0, \hat{y}_{2\cdot m+1}, \ldots, \hat{y}_{2\cdot m+n}, 0, \ldots, 0)^{tr}.$$

Lemma. *Any coordinates of the column vector $\tilde{y}$ are equal to either 0 or 1.*

And our main result is the following:

Theorem. *The edge with the number i corresponds the unit value of the coordinates $\tilde{y}_{2\cdot m+i}$, where $i = 1, 2, \ldots, n$, of vector $\tilde{y}$. All these edges are the minimal cut such that this cut separates the sources S and the sinks T.*

This statement is the criterion: if the arc is marked by number 1 then this arc belongs to the minimal cut; if the arc is marked by number 0 then it does not belong to the minimal cut. We find one of "the pipeline bottleneck" of the network for the maximum-flow.

Constant inflows and outflows in vertices and over edges take some part of the flow. Thus the sum of capacity constraints of edges for such a cut can be larger than the minimum of the dual problem. Let us remark that it follows from numerical experiments: the assertion of the theorem is correct for an unconnected graph.

The provided solution to the task of finding the pipeline bottlenecks can be used effectively in the optimization of the strategic planning in the reconstruction and development of network objects for the Gazprom gas-transport system, the Transneft oil pipeline system, etc. Also, this solution can be used for the control of flows upon the whole.

References

[1] Beklemishev D V 2008 *Additional chapters of linear algebra: Tutorial. Second edition* (SPb, Moscow, Krasnodar: Lan Press) pp 450-456

[2] Ribnikov K K 2010 *Introduction to discrete mathematics and the theory of extreme problems* (Moscow: Gelios ARV Press) pp 243-247

[3] Gorelik V A and Ushakov I A 1986 *Operations research. Textbook for college with a degree in Applied Mathematics* (Moscow: Mashinostroenie Press) pp 62-65

[4] Cormen T H, Leiserson C E, Rivest R L and Stein C 2008 *Introduction to Algorithms. Third edition* (Cambridge, Massachusetts / London, England: The MIT Press) pp 843-862, 879-885