Acta Universitatis Carolinae. Mathematica et Physica

C. Roos; Miron Tegze

An extension of the MS pivoting rule to capacitated network flow problems

Acta Universitatis Carolinae. Mathematica et Physica, Vol. 29 (1988), No. 1, 73--88

Persistent URL: http://dml.cz/dmlcz/142593

1988 ACTA UNIVERSITATIS CAROLINAE—MATHEMATICA ET PHYSICA VOL. 29. NO. 1

An Extension of the MS Pivoting Rule to Capacitated Network Flow Problems

C. ROOS,
Delft*), The Netherlands**)
MIRON TEGZE
Faculty of Mathematics and Physics, Charles University, Prague***)

Received 5 September 1987

In [8], the first author proposed a new pivoting rule (called maximal slope rule, or shortly MS rule) for the dual simplex method which exploits the tree structure of the basis in the case of network flow problems. In the present paper, we extend his approach to the 'capacitated' network flow problem, i.e. to networks with (lower and) upper capacity bounds on the arcs.

It turns out that the coefficients appearing in the MS rule become more complicated in the capacitated case and they cannot be calculated with the same simplicity as in the uncapacitated case. We discuss the possibility to approximate these coefficients maintaining thus both the relative small number of iterations and fast calculation in each pivot step. Computational experiments are reported with several variants of the resulting pivoting rule.

1. Introduction

It is known that when solving network optimization problems with simplex based algorithms the network structure of the problem can be profitable exploited. The basis **B** can be represented as a spanning tree of the network and the equations $\mathbf{Bx} = \mathbf{b}$ and $\pi\mathbf{B} = \mathbf{c}$ can be solved combinatorially. The pivot step consists in a combinatorial updating of the basis tree.

Although, in general, compared with dual codes the primal codes are much more efficient for solving network flow problems (cf. e.g. [2, 3]), the pivoting rule for the dual simplex method which is proposed in [8] may well lead to dual codes which are competitive with primal codes. Computational results reported in [8] support this expectation. The pivoting rule is the so called *maximal slope rule*, or shortly *MS rule*. This rule significantly reduces the required number of iterations. Moreover the rule exploits the tree structure of the basis substantially: it tends to choose small subtrees to update, which helps to further increase the speed in each

*) The research for this paper was done during a three month stay of the second author in Delft which was arranged on the basis of the Cultural Agreement between home countries of the authors.

**) Delft University of Technology, Faculty of Mathematics and Informatics, P. O. Box 356, 2600 AJ Delft, The Netherlands.

***) Malostranské nám. 25, 118 00 Praha 1, Czechoslovakia.

pivot step. The paper [8] deals primarily with the uncapacitated network case. The aim of this paper is to extend the MS approach to the capacitated case.

This paper is organized as follows: Firstly we recall some facts and results concerning simplex network algorithms and the results of [8]. Then we derive the necessary modifications for the capacitated case and discuss the differences and the expected behaviour. Finally, some experimental results arc reported for several variants of the MS rule.

2. Preliminaries

Let $\mathscr{G} = (\mathscr{N}, \mathscr{A})$ be a connected oriented graph with n nodes and m arcs. In this paper we will use the convention that an arc pointing from node i to node j is denoted shortly as ij. Let $\mathbf{G}^* = [g^*_{k,ij}]_{k\in\mathscr{N}, ij\in\mathscr{A}}$ denote the node-arc incidence matrix of $\mathscr{G}$. So, for each node k and for each arc ij one has:

$$(2.1) \qquad g^*_{k,ij} = \begin{cases} 1 & \text{if} \quad k = j\,, \\ -1 & \text{if} \quad k = i\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Let $\mathbf{G}$ be the matrix obtained from $\mathbf{G}^*$ by omitting one row, let it be the r-th row. This is to avoid degeneracy of the forthcoming linear programming formulations, see e.g. [3, 4].

Now consider the network optimization problem

$$(2.2) \qquad \begin{array}{c} \min \mathbf{cf} \\ \hline \mathbf{Gf} = \mathbf{b} \\ \mathbf{f} \geqq \mathbf{0} \end{array}$$

where $\mathbf{b} = [b_i]_{i\in\mathscr{N}}$ is the vector of demands in the nodes different from the root node r, $\mathbf{c} = [c_{ij}]_{ij\in\mathscr{A}}$ is the cost vector, and $\mathbf{f} = [f_{ij}]_{ij\in\mathscr{A}}$ represents the flow.

It is well known that there exists a 1-1 correspondence between the spanning trees of $\mathscr{G}$ and the possible bases of this linear programming system (c.f. [3, 4, 7]). Given a spanning tree $\mathscr{T}$, we denote by $\mathbf{T}$ the submatrix of $\mathbf{G}$ formed by the columns which correspond to the arcs in $\mathscr{T}$ and we call these arcs the *in-tree arcs*. Proclaim the node r (i.e. the node whose row was omitted in $\mathbf{G}$) to be the *root* of $\mathscr{T}$. Then in any column of $\mathbf{T}$ corresponding to an arc which is incident to r there is only one nonzero entry; the row of this entry determines the corresponding *son* of the root r *hanging* on the arc. If s is one of such sons then $g_{s,ij} = 1$ if $ij = rs$ and $g_{s,ij} = -1$ $ij = sr$. If we omit in $\mathbf{T}$ all the rows corresponding to the root and its sons, we can similarly develop the second level of sons of r, etc., until the tree $\mathscr{T}$ has been reconstructed completely.

For our considerations, it is important that for any subtree of $\mathscr{T}$ which has a root

s and which is hanging on the arc ij, the correspondence between s and ij is given by:

(2.3) $$s = \begin{cases} j & \text{if} \quad g_{s,ij} = 1 . \\ i & \text{if} \quad g_{s,in} = -1 . \end{cases}$$

In this way the given spanning three $\mathcal{T}$ induces a 1-1 correspondence between the in-tree arcs and the nodes. In the following we shall frequently refer to this correspondence. In fact it is analogous to the well known 1-1 correspondences between the basic variables and the constraints in the usual linear programming context, because in the network programming case the basic variables are the flow values on the in-tree arcs, and the constraints are the balance equations in the nodes.

If we denote the in-tree part of the flow vector $\mathbf{f}$ by $\tilde{}\mathbf{f}$ then $\tilde{}\mathbf{f}$ can be calculated from

(2.4) $$\mathbf{T}\tilde{}\mathbf{f} = \mathbf{b} .$$

which gives

(2.5) $$\tilde{}\mathbf{f} = \mathbf{T}^{-1}\mathbf{b}$$

As stressed in [3, 4, 6], the computation of $\tilde{}\mathbf{f}$ can be accomplished combinatorially (using in fact equation (2.4), not calculating $\mathbf{T}^{-1}$, and using logical instead of arithmetical operations) which is the reason for the effectiveness of network algorithms when compared with classical linear programming ones.

Now suppose that the solution of the given network optimization problem proceeds via the dual simplex method. Then in each step a dual feasibele solution $\mathbf{f}$ is given together with a spanning tree $\mathcal{T}$. This dual solution, which will be denoted as π, is determined by $\mathcal{T}$ according to the equation:

(2.6) $$\pi\mathbf{T} = \tilde{}\mathbf{c} .$$

Dual feasibility means that $\pi\mathbf{G} \leqq \mathbf{c}$ (i.e. the *differential* $\mathbf{v} = \pi\mathbf{G}$ for the out-of-tree arcs is dual feasible: $\pi_j - \pi_i \leqq c_{ij}$ for each arc ij). If moreover $\tilde{}\mathbf{f} \geqq 0$ then the flow $\mathbf{f}$ is also primal feasible and hence optimal. If $\mathbf{f}$ is not primal feasible however, then an iteration consists of removing from $\mathcal{T}$ some in-tree arc (the *leaving arc*) violating the constraint $\tilde{}\mathbf{f} \geqq 0$ and replacing it by another arc (the *entering arc*), thus yielding one pivot step. As described in a vast literature, this pivoting is accomplished also combinatorially by using a cut which separates the subtrees arising from $\mathcal{T}$ when removing the leaving arc.

If there are more than one non-feasible arcs, the pivoting rule has to choose one of them as the leaving arc. Hence the dual simplex method is described as soon as the pivoting rule is stated.

The MS rule has a clear geometrical interpretation which is described in detail in [8]. Roughly speaking it consists in choosing the feasible direction, in the dual cone determined by the basis, which is both extremal and has the minimal angle with $\mathbf{b}$. Stated algebraically, the rule is

$$\max\left\{-\frac{x_j}{|\mathbf{r}_j|}\ ;\quad x_j = \mathbf{r}_j\mathbf{b} < 0\right\}, \tag{2.7}$$

where $\mathbf{r}_j$ denotes the j-th row of the inverse of the basis.

By using the above established 1-1 correspondence between the in-tree arcs and nodes via the basis tree, the MS rule can be stated as

$$\max\left\{-\frac{f_{ij}}{|r_{ij}|}\ ;\quad f_{ij} < 0\right\}. \tag{2.8}$$

As shown in [8], the value of $|r_{ij}|^2$ is nothing else as the size of the subtree hanging on the arc ij. Thus the 'MS ratio' is easy to obtain: it suffices to keep in memory for each node the size of the subtree of $\mathcal{T}$ rooted at that node and to update these values in each pivot step. In fact, this requires no additional computational burden, because in most effective network codes these numbers are already available (c.f. [2, 3]). The MS rule, when compared with the standard approach:

$$\max\{-f_{ij}; f_{ij} \geqq 0\} \tag{2.9}$$

has a positive side effect. It tends to choose small subtrees, thus accelerating the speed in each pivot step, since the amount of data, which has to be updated in each pivot step, is proportional to the size of the subtree. The computational results of [8] witness that this improvement is substantial (2–5 times faster).

3. The main result

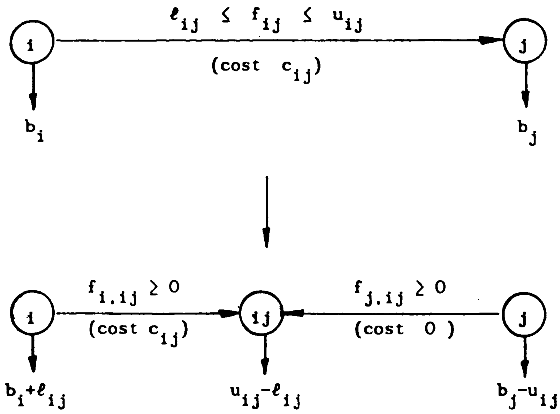
Throughout this section we deal with the following situation: As before, let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ be a connected oriented graph with n nodes and m arcs, and let

$$\begin{array}{c}\min \mathbf{cf}\\ \hline \mathbf{Gf} = \mathbf{b}\\ \mathbf{l} \leqq \mathbf{f} \leqq \mathbf{u}\end{array} \tag{3.1}$$

be a capacitated network flow optimization problem on $\mathcal{G}$, where $\mathbf{G} = [g_{k,ij}]$ denotes the node-arc incidence matrix of the graph $\mathcal{G}$ (with the r-th row omitted to avoid degeneracy, see Preliminaries), $\mathbf{b} = [b_k]$ denotes the vector of demands in the nodes, $\mathbf{l} = [l_{ij}]$ and $\mathbf{u} = [u_{ij}]$ are respectively the lower and upper capacity bounds on the arcs, and $\mathrm{c} = [c_{ij}]$ denotes the cost vector for a flow $\mathbf{f} = [f_{ij}]$ on $\mathcal{G}$.

3.1. Remark

Any capacitated network programming problem can be easily reduced to the uncapacitated case by replacing each arc by a couple of opposite oriented serial arcs with a dummy node between them in the following way:



(The flow $f_{i,ij}$ corresponds to $f_{ij} - l_{ij}$, and $f_{j,ij}$ to $u_{ij} - f_{ij}$). This reduction however leads to a considerable increase of the size of the problem: for each double bounded arc we have to add one arc and one node. □

Suppose that the computation proceeds via the dual simplex network method (as recalled in the Preliminaries). The current dual feasible solution **f** is given together with the basis tree $\mathscr{T}$. Denote by $\tilde{\mathbf{G}}$ the submatrix of **G** consisting of the columns corresponding to the in-tree arcs and by $\tilde{\mathbf{f}}$ the corresponding in-tree flow. Note that each out-of-tree arc has a flow value which is equal either to the lower or to the upper capacity bound. So, let $\bar{\mathbf{G}}$ denote the submatrix of **G** consisting of the columns of **G** corresponding to the upper bounded out-of-tree arcs and $\underline{\mathbf{G}}$ the submatrix of **G** consisting of the columns of **G** corresponding to the lower bounded ones; if the symbols $\bar{\mathbf{f}}$, $\underline{\mathbf{f}}$, $\bar{\mathbf{c}}$, $\underline{\mathbf{c}}$, $\tilde{\mathbf{c}}$, $\bar{\mathbf{l}}$, $\underline{\mathbf{l}}$, and $\tilde{\mathbf{l}}$ are used according to this partition of the matrix **G**, then for the current flow we have

$$
\begin{aligned}
\tilde{\mathbf{G}}\,\tilde{\mathbf{f}} &= \mathbf{b} - \bar{\mathbf{G}}\,\bar{\mathbf{u}} - \underline{\mathbf{G}}\,\underline{\mathbf{l}}\,, \\
\bar{\mathbf{f}} &= \bar{\mathbf{u}}\,, \\
\underline{\mathbf{f}} &= \underline{\mathbf{l}}\,.
\end{aligned} \tag{3.2}
$$

The dual feasibility of **f** is assumed. Hence if $\tilde{\mathbf{l}} \leqq \tilde{\mathbf{f}} \leqq \tilde{\mathbf{c}}$ then the flow is also primal feasible and hence optimal: if not, $\mathscr{J}$ will denote the set of arcs of $\mathscr{T}$ not satisfying the primal feasibility constraints.

In the remainder of this section we deal with the latter situation and we concentrate the attention on the choice of the leaving arc.

3.2. Theorem

The MS-pivoting rule selects as leaving arc an arc $ij \in \mathscr{J}$ which maximizes the ratio

$$
\frac{\tilde{l}_{ij} - \tilde{f}_{ij}}{\sqrt{(P_{ij} + R_{ij})}} \quad \text{if} \quad \tilde{l}_{ij} > \tilde{f}_{ij}\,, \tag{3.3}
$$

(3.4) $$\frac{{}^{\sim}f_{ij} - {}^{\sim}u_{ij}}{\sqrt{(P_{ij} + 1 + R_{ij})}} \quad \text{if} \quad {}^{\sim}f_{ij} > {}^{\sim}u_{ij}\,,$$

where R_{ij} denotes the size of the subtree of $\mathscr{T}$ which is hanging on the arc ij and P_{ij} the number of the upper bounded out-of-tree arcs entering or leaving this subtree.

Proof:

To start with we bring the given problem in the canonical form. By substitution of $\mathbf{y} := \mathbf{f} - \mathbf{l}$ and by introducing a slack vector $\mathbf{z}$ the problem can be rewritten as follows:

$$\begin{array}{l}\min \mathbf{cy} \\ \hline \mathbf{Gy} \qquad = \mathbf{b} - \mathbf{Gl} \\ \;\mathbf{y} + \mathbf{z} = \mathbf{u} - \mathbf{l} \\ \;\mathbf{y} \geqq 0\,, \quad \mathbf{z} \geqq 0\,. \end{array}$$

Using the above defined partitioning of the matrix $\mathbf{G}$ and extending this partitioning to all relevant vectors the constraint part of this problem can be rewritten as follows:

$$\begin{array}{lllll}
{}^{\sim}\mathbf{G}\,{}^{\sim}\mathbf{y} & + {}^{-}\mathbf{G}\,{}^{-}\mathbf{y} + {}_{-}\mathbf{G}\,{}_{-}\mathbf{y} & & = \mathbf{b} - \mathbf{Gl}\,, & \\
{}^{\sim}\mathbf{y} + {}^{\sim}\mathbf{z} & & & = {}^{\sim}\mathbf{u} - {}^{\sim}\mathbf{l} & ({}^{\sim}\mathbf{y}, {}^{\sim}\mathbf{z} \geqq 0)\,, \\
 & \quad {}^{-}\mathbf{y} & + {}^{-}\mathbf{z} & = {}^{-}\mathbf{u} - {}^{-}\mathbf{l} & ({}^{-}\mathbf{z} = 0)\,, \\
 & \qquad\quad {}_{-}\mathbf{y} & \quad + {}_{-}\mathbf{z} & = {}_{-}\mathbf{u} - {}_{-}\mathbf{l} & ({}_{-}\mathbf{y} = 0)\,.
\end{array}$$

The basic variables are ${}^{\sim}\mathbf{y}$, ${}^{\sim}\mathbf{z}$, ${}^{-}\mathbf{y}$ and ${}_{-}\mathbf{z}$. The conditions ${}^{\sim}\mathbf{y} \geqq 0$, ${}^{\sim}\mathbf{z} \geqq 0$ express primal feasibility and they are in general not all satisfied. Hence the basic submatrix $\mathbf{B}$ of $\mathbf{G}$ has the following block-structured form (the variables are added to stress the partition of the system):

$$\begin{array}{cc} & \begin{array}{cccc} {}^{\sim}\mathbf{y} & {}^{\sim}\mathbf{z} & {}^{-}\mathbf{y} & {}_{-}\mathbf{z} \\ \downarrow & \downarrow & \downarrow & \downarrow \end{array} \\ \mathbf{B} = & \begin{bmatrix} {}^{\sim}\mathbf{G} & \mathbf{0} & {}^{-}\mathbf{G} & \mathbf{0} \\ \mathbf{E} & \mathbf{E} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{E} \end{bmatrix}, \end{array}$$

where each $\mathbf{E}$ denotes an identity, and each $\mathbf{0}$ a zero matrix of the appropriate size. Now it easily follows that the inverse basis matrix is given by

$$\mathbf{B}^{-1} = \begin{bmatrix} {}^{\sim}\mathbf{G}^{-1} & \mathbf{0} & -{}^{\sim}\mathbf{G}^{-1}\,{}^{-}\mathbf{G} & \mathbf{0} \\ -{}^{\sim}\mathbf{G}^{-1} & \mathbf{E} & {}^{\sim}\mathbf{G}^{-1}\,{}^{-}\mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{E} \end{bmatrix},$$

New we are ready to determine the appropriate MS ratios. The MS ratio is given by $-\mathbf{r}\boldsymbol{\beta}/|\mathbf{r}|$, where $\mathbf{r}$ is some row of the inverse of the basis matrix $\mathbf{B}$ and $\boldsymbol{\beta}$ denotes the right hand side vector.

In the present case we need hence to distinguish between two cases dependent on whether the primal infeasibility of the leaving arc ij is reflected by ${}^{\sim}y_{ij} < 0$ (i.e. ${}^{\sim}f_{ij} < {}^{\sim}l_{ij}$) or by ${}^{\sim}z_{ij} < 0$ (i.e. ${}^{\sim}f_{ij} > {}^{\sim}u_{ij}$). If k denotes the root of the subtree hanging on the arc ij then the corresponding ribvector $\mathbf{r}_k$ is in these cases respectively given by:

$$\mathbf{r}_k = (\ \ \mathbf{q}_k, \mathbf{0}, \ \ -\mathbf{q}_k\,{}^{-}\mathbf{G}, \mathbf{0}) \quad \text{if} \quad {}^{\sim}y_{ij} < 0\,,$$
$$\mathbf{r}_k = (-\mathbf{q}_k, \mathbf{e}_{ij}, \ \ \mathbf{q}_k\,{}^{-}\mathbf{G}, \mathbf{0}) \quad \text{if} \quad {}^{\sim}z_{ij} < 0\,,$$

where $\mathbf{e}_{ij}$ is a unit sector of the appropriate size (with all but one entries equal to zero, and a one in the position corresponding to arc ij) and $\mathbf{q}_k$ denotes the k-th row of ${}^{\sim}\mathbf{G}^{-1}$. The node k and the arc ij are related by

$$\mathbf{q}_k\,{}^{\sim}\mathbf{G} = \mathbf{e}_{ij}\,.$$

Hence the ratios for the MS rule are

$$-\frac{\mathbf{q}_k(\mathbf{b} - \mathbf{G}\mathbf{l} - {}^{-}\mathbf{G}({}^{-}\mathbf{u} - {}^{-}\mathbf{l}))}{\sqrt{(|\mathbf{q}_k\,{}^{-}\mathbf{G}|^2 + |\mathbf{q}_k|^2)}} \qquad \text{if} \quad {}^{\sim}f_{ij} < {}^{\sim}l_{ij}\,.$$

$$-\frac{\mathbf{e}_{ij}({}^{\sim}\mathbf{u} - {}^{\sim}\mathbf{l}) - \mathbf{q}_k(\mathbf{b} - \mathbf{G}\mathbf{l} - {}^{-}\mathbf{G}({}^{-}\mathbf{u} - {}^{-}\mathbf{l}))}{\sqrt{(|\mathbf{q}_k\,{}^{-}\mathbf{G}|^2 + 1 + |\mathbf{q}_k|^2)}} \quad \text{if} \quad {}^{\sim}f_{ij} > {}^{\sim}u_{ij}\,.$$

Now let us firstly consider the enumerators in the above quotients. Note that the in-tree flow ${}^{\sim}\mathbf{f}$ satisfies

$${}^{\sim}\mathbf{G}\,{}^{\sim}\mathbf{f} + {}^{-}\mathbf{G}\,{}^{-}\mathbf{f} + {}_{-}\mathbf{G}\,{}_{-}\mathbf{f} = \mathbf{b}\,.$$

Hence, since ${}^{-}\mathbf{f} = {}^{-}\mathbf{u}$ and ${}^{-}\mathbf{f} = {}_{-}\mathbf{l}$, after multiplication with $\mathbf{q}_k$ we obtain

$$f_{ij} = \mathbf{q}_k(\mathbf{b} - {}^{-}\mathbf{G}\,{}^{-}\mathbf{u} - {}_{-}\mathbf{G}\,{}_{-}\mathbf{l})\,.$$

By direct computation we now obtain for the enumerator of the first ratio:

$$\mathbf{q}_k(\mathbf{b} - \mathbf{G}\mathbf{l} - {}^{-}\mathbf{G}({}^{-}\mathbf{u} - {}^{-}\mathbf{l})) =$$
$$= \mathbf{q}_k(\mathbf{b} - {}^{\sim}\mathbf{G}\,{}^{\sim}\mathbf{l} - {}^{-}\mathbf{G}\,{}^{-}\mathbf{l} - {}_{-}\mathbf{G}\,{}_{-}\mathbf{l} - {}^{-}\mathbf{G}\,{}^{-}\mathbf{u} + {}^{-}\mathbf{G}\,{}^{-}\mathbf{l}) =$$
$$= -{}^{\sim}l_{ij} + \mathbf{q}_k(\mathbf{b} - {}^{-}\mathbf{G}\,{}^{-}\mathbf{u} - {}_{-}\mathbf{G}\,{}_{-}\mathbf{l}) = {}^{\sim}f_{ij} - {}^{\sim}l_{ij}\,.$$

This fixes the first enumerator. The second enumerator now simply equals

$${}^{\sim}u_{ij} - {}^{\sim}l_{ij} - {}^{\sim}f_{ij} + {}^{\sim}l_{ij} = {}^{\sim}u_{ij} - {}^{\sim}f_{ij}\,.$$

It remains to treat the denominators. As shown in [8], $\mathbf{q}_k$ is the characteristic function of the incidence of nodes to the subtree hanging on the arc ij if $k = j$ and minus this function if $k = i$. Hence it is clear that $|q_k|^2 = R_{ij}$ as desired in the Theorem. Due to this structure of the vector $\mathbf{q}_k$, in the expression $\mathbf{q}_k\,{}^{-}\mathbf{G}$ (which is a vector whose entries correspond to the upper bounded out-ot-tree flows) the st-entry is zero whenever the initial node s and the terminal node t are both outside the subtree or both inside the subtree hanging on the arc ij; the st-entry of $\mathbf{q}_k\,{}^{-}\mathbf{G}$ is 1 (resp. -1)

if the arc st is entering or leaving this subtree in the opposite (resp. the same) direction as the in-tree arc (i, j).

Hence the norm $|\mathbf{q}_k{}^{-}\mathbf{G}|^2$ is simply the number of out-of-tree upper active arcs which enter or leave the subtree hanging on the arc ij. This number was denoted as P_{ij} in the Theorem.

Summarizing, we have found the following MS ratios:

$$\frac{{}^{\sim}l_{ij} - {}^{\sim}f_{ij}}{\sqrt{(P_{ij} + R_{ij})}} \quad \text{if} \quad {}^{\sim}f_{ij} < {}^{\sim}l_{ij},$$

$$\frac{{}^{\sim}f_{ij} - {}^{\sim}u_{ij}}{\sqrt{(P_{ij} + 1 + R_{ij})}} \quad \text{if} \quad {}^{\sim}f_{ij} > {}^{\sim}u_{ij},$$

which completes the proof of the Theorem. □

3.3. Remark

In the uncapacitated case (i.e. $\mathbf{f} \geqq 0$ only) the ratios are $-f_{ij}/\sqrt{R_{ij}}$, which is indeed a special case of the first ratio, obtained by setting $\mathbf{l} = 0$ in (3.3).

3.4. Remark

As mentioned before, the term R_{ij} in the denominators $\sqrt{(P_{ij} + 1 + R_{ij})}$ and $\sqrt{(P_{ij} + R_{ij})}$ of the MS-ratios for the capacitated case can be accomplished very quickly, since it is simply the size of a subtree which can be held in memory and updated easily. The authors do not see how to develop an efficient calculation scheme which yields the terms P_{ij} with the same ease. Therefore they suggest to approximate these term in the following way.

Let m denote the total number of upper bounded out-of-tree arcs. Then the expected value for the number of such arcs connecting a node set S (of cardinality s, say) with its complement $V\backslash S$ can be calculated straightforwardly. This value tuns out to be

$$\frac{s(n - s)}{n(n - 1)} 2m .$$

So the expected value for $R_{ij} + P_{ij}$ is given by

$$E(R_{ij} + P_{ij}) = R_{ij} + \frac{R_{ij}(n - R_{ij})}{n(n - 1)} 2m . \tag{3.5}$$

Since m can be easily updated in each pivoting step the expected value for the denominators in the expressions for both MS ratios can be obtained without hardly any additional computational effort.

4. Experimental results

For our experiments we used two sets of test problems, all generated by the network generator NETGEN [3]. The first set consists of 31 problems and the second set of 23 problems. Roughly speaking, the problems in the two sets can be classified as described in the following two tables.

Table 4.1. Problems in the first set

Problem-number	Number of nodes	Number of arcs	Type of the problem
1— 5	200	1300—2900	Transportation
6—10	300	3150—6300	Transportation
11—15	400	1500—4500	Assignment
16—27	400	1306—2836	Transshipment
28—31	1000	2900—4800	Transshipment

Table 4.2. Problems in the second set

Problem-number	Number of nodes	Number of arcs	Type of the problem
1— 4	100	250—2000	Transportation
5— 6	200	1200—2200	Transportation
7— 9	300	1000	Transportation
10	400	5000	Transportation
11—23	200—400	500—3000	Transhippment

A detailed description of the nature of these problems is given in the Appendix.

The problems were solved on an IBM PC/AT by using appropriate variants of a dual network code written by the first author [8]. Our aim has been to compare pivoting rules of the form

$$\max\left(-\frac{v_{ij}}{\sqrt{T_{ij}}} : v_{ij} := \text{flow violation on arc } ij\right).$$

We used the following choices of T_{ij}:

1. $T_{ij} = 1$ (classical case)
2. $T_{ij} = R_{ij}$ (uncapacitated case)
3. $T_{ij} = E(R_{ij} + P_{ij})$ (expected capacitated MS ratio)

4. $T_{ij} = R_{ij} + P_{ij}$ (capacitated case)
5. $T_{ij} = R_{ij} + Q_{ij}$
6. $T_{ij} = Q_{ij}$,

where Q_{ij} denotes the total number of out-of-tree arcs leaving the subtree hanging on arc ij. So we disregarded the term '1' in (3.4), because it seems reasonable to assume that the contribution of this term may well be neglected. Since we focussed our attention to the influence of the pivoting rule on the required number of pivotsteps, we used in all cases the same *initial* solution, namely the trivial one. The results of the computations are collected in Table 4.3 and Table 4.4 below.

Table 4.3. Comparison of several pivoting rules on the first set of test problems

Problem-number	Optimal value	Number of Iterations					
		1	2	3	4	5	6
1	821555	180	192	192	170	175	170
2	781822	279	331	327	291	305	290
3	577211	642	792	863	556	612	608
4	633759	1410	1395	1396	1016	1143	1129
5	1440933	1612	1198	1212	945	1060	992
6	1309555	2218	2307	2103	1534	1644	1472
7	2306773	1183	1164	1167	975	1112	1071
8	22796562	1206	1162	1158	1038	1082	1075
9	2367529	1566	1259	1261	1124	1133	1113
10	2453402	5866	5777	5778	3132	3895	3268
11	7951281	1826	1752	1873	1587	1611	1550
12	9610604	1559	1208	1302	1207	1187	1166
13	6298666	2070	1531	1590	1481	1516	1498
14	6412335	3139	2161	2473	2130	2111	2114
15	392467	347	365	364	336	349	360
16	120464	355	273	278	264	270	272
17	878593	715	696	795	660	657	623
18	94122583	763	689	752	668	666	677
19	9615337	733	683	824	657	650	656
20	1269581	278	232	234	232	227	231
21	2911385	522	458	508	458	483	491
22	3095330	300	279	276	290	286	283
23	7447606	566	563	594	553	532	524
Total		29335	26467	27320	21304	22706	21633

Let us derive some conclusions from these results. Firstly we see that the rules 4.5 and 6 behave approximately the same on all test problems. This did we not expect, because rule 4 is the MS rule for the capacitated case. More importantly,

Table 4.4 Comparison of several pivoting rules on the second set of test problems

Problem-number	Optimal value	Number of iterations					
		1	2	3	4	5	6
1	2054059	290	254	255	254	251	261
2	1818866	351	261	257	261	267	268
3	1646007	322	259	258	259	249	255
4	1332598	310	265	264	265	264	263
5	1374153	366	273	274	273	268	267
6	2135438	610	391	390	391	395	391
7	1818475	486	393	397	393	419	405
8	1803322	604	417	433	417	396	399
9	1650449	529	405	408	405	410	408
10	1988555	807	439	431	439	420	460
11	4991	922	461	457	460	396	400
12	3843	816	434	434	434	391	390
13	3048	867	439	442	439	406	410
14	2392	844	464	441	464	388	386
15	2460	934	483	540	483	431	432
16	66644957	499	302	311	302	315	322
17	33296481	510	286	314	286	316	325
18	62451490	507	288	293	288	306	308
19	33296481	508	283	301	283	334	320
20	79562354	572	312	313	313	326	380
21	25214811	550	288	305	294	329	348
22	78868140	587	303	304	303	340	349
23	24765976	414	268	279	268	310	316
24	80022555	589	196	195	207	194	189
25	69302042	718	231	227	233	223	225
26	67799030	497	140	140	142	134	132
27	51296683	506	175	176	175	178	177
28	131264893	1625	608	607	608	598	615
29	114387763	2074	630	638	630	694	666
30	86559373	2515	591	600	591	622	620
31	80333340	2293	582	596	852	570	575
Total		24122	11121	11280	11142	11140	11252

on the second set rule 2 is the best one and on the first set it behaves slightly worse than rule 4, which is the best rule for set 2. This is nice, because rule 2 is the uncapacitated version of the MS rule, which requires the least amount of additional computational work. Since rule 4 requires in each iteration the calculation of P_{ij}, which is very time consuming, and rule 1 requires per iteration the same amount of computational work as rule 2, our first conclusion is that on the problems in the two sets rule 2 gives the best results. It might well be that the reason of the good behaviour of rule 2, also in the capacitated case, lies in the fact that the 'uncapacitated denominator' is a fairly good approximation of the capacitated one.

Note that with respect to the number of iterations, rule 1 is the worst rule on all problems. It is interesting to see that the difference in behaviour between rule 1 and the other rules differs significantly for the two sets. For the problems in the second rule 1 requires on the average more than two times more iterations than rule 2, whereas for the problems in the first set this ratio is about 1.1. The reason for this may be that the problems in the second set are 'very capacitated', the upper capacity bounds of the arcs lie in the interval [5, 10] for these problems, except for problem 16 which has the upper capacities in the interval [50, 100] (cf. Appendix), whereas the problems in the first set are not capacitated or capacitated with high upper capacity bounds, namely in the interval [16 000, 120 000].

References

[1] GLOVER F. and KLINGMAN, D., 'Recent Developments in Computer Implementation Technology for Network Flow Algorithms', Infor vol. 20, no. 4, 433—452, 1982.

[2] GLOVER, F., KARNEY, D. and KLINGMAN, D., 'Implementation and Computational Comparison of Primal, Dual and Primal-Dual Computer Codes for Minimum Cost Network Flow Problems', Networks, vol. 4, no. 3, 191—212, 1974.

[3] KARNEY, D. and KLINGMAN, D., 'Netgen Revisited: A Program for Generating Large Scale (Un)Capacitated Assignment, Transportation, and Minimum Cost Flow Problems', Research Report CCS 320, Center for Cybernetic Studies, The University of Texas at Austin, 1978.

[4] KLINGMAN, D. and GLOVER, F.: 'Tutorial: Networks', Report CBDA 118 of the Center for Business Decision Analysis, October 1984.

[5] LAWLER, E. L., 'Combinatorial Optimization: Networks and Matroids', Holt, Rinehart and Winston, 1976.

[6] LUENBERGER, D. G., 'Linear and Nonlinear Programming', Adison-Wesley Publ. Comp., 1974.

[7] PAPADIMITRIOU, C. H. and STEIGLITZ, K., 'Combinatorial Optimization, algorithms and complexity', Prentice-Hall, Unc., Englewood Cliffs, New Jersey, 1982.

[8] ROOS, C., 'New Pivoting Rule For The Dual Network Simplex Method', Report 86—17, Faculty of Mathematics and Informatics, Delft University of Technology, 1986.

Data of the first set of set problems

	Number of				Cost Range		Total supply	Transshipment		% of high cost	% of arcs capacitated	Upper bound range		Random no. seed
problem	nodes	sources	sinks	arcs	min	max		sources	sinks			min	max	
1	200	100	100	1300	1	100	10000	0	0	0	0	0	0	1350246
2	200	100	100	1500	1	100	10000	0	0	0	0	0	0	1350246
3	200	100	100	2000	1	100	100000	0	0	0	0	0	0	1350246
4	200	100	100	2200	1	100	100000	0	0	0	0	0	0	1350246
5	200	100	100	2900	1	100	100000	0	0	0	0	0	0	1350246
6	300	150	150	3150	1	100	150000	0	0	0	0	0	0	1350246
7	300	150	150	4500	1	100	150000	0	0	0	0	0	0	1350246
8	300	150	150	5155	1	100	150000	0	0	0	0	0	0	1350246
9	300	150	150	6075	1	100	150000	0	0	0	0	0	0	1350246
10	300	150	150	6300	1	100	150000	0	0	0	0	0	0	1350246
11	400	200	200	1500	1	100	200	0	0	0	0	0	0	1350246
12	400	200	200	2250	1	100	200	0	0	0	0	0	0	1350246
13	400	200	200	3000	1	100	200	0	0	0	0	0	0	1350246
14	400	200	200	3750	1	100	200	0	0	0	0	0	0	1350246
15	400	200	200	4500	1	100	200	0	0	0	0	0	0	1350246
16	400	8	60	1306	1	100	400000	0	0	30	20	16000	30000	1350246

17	400	8	60	1443	1	100	400000	0	0	30	20	16000	30000	1350246
18	400	8	60	1306	1	100	400000	0	0	30	20	20000	120000	1350246
19	400	8	60	2443	1	100	400000	0	0	30	20	20000	120000	1350000
20	400	8	60	1416	1	100	400000	5	50	30	40	16000	30000	1350246
21	400	8	60	2836	1	100	400000	5	50	30	40	16000	30000	1350246
22	400	8	60	1416	1	100	400000	5	50	30	40	20000	120000	1350245
23	400	8	60	2836	1	100	400000	5	50	30	40	20000	120000	1350246
24	400	4	12	1382	1	100	400000	0	0	30	80	16000	30000	1350246
25	400	4	12	2676	1	100	400000	4	0	30	80	16000	30000	1350246
26	400	4	12	1382	1	100	400000	0	0	30	80	20000	120000	1350246
27	400	4	12	2676	1	100	400000	0	0	30	80	20000	120000	1350246
28	1000	50	50	2900	1	100	1000000	0	0	0	0	0	0	1350246
29	1000	50	50	3400	1	100	1000000	0	0	0	0	0	0	1350246
30	1000	50	50	4400	1	100	1000000	0	0	0	0	0	0	1350246
31	1000	50	50	4800	1	100	1000000	0	0	0	0	0	0	1350246
32	1500	75	75	4342	1	100	1500000	0	0	0	0	0	0	1350246
33	1500	75	75	4385	1	100	1500000	4	0	0	0	0	0	1350246
34	1500	75	75	5107	1	100	1500000	0	0	0	0	0	0	1350246
35	1500	75	75	5730	1	100	1500000	0	0	0	0	0	0	1350246

Data of the second set of test problems

	Number of				Cost Range		Total supply	Transshipment		% of high cost	% of arcs capacitated	Upper bound range		Random no. seed
problem	nodes	sources	sinks	arcs	min	max		sources	sinks			min	max	
1	100	5	95	250	1	100	10000	0	0	80	100	5	10	12345678
2	100	10	90	300	1	100	10000	0	0	80	100	5	10	12345678
3	100	10	90	1000	1	100	10000	0	0	80	100	5	10	12345678
4	100	25	75	2000	1	100	10000	0	0	80	100	5	10	12345678
5	200	20	180	1200	1	100	20000	0	0	80	100	5	10	12345678
6	200	20	180	2200	1	100	20000	0	0	80	100	5	10	12345678
7	300	10	290	1000	1	100	30000	0	0	80	100	5	10	12345678
8	300	10	290	1000	1	1000	30000	0	0	80	100	5	10	12345678
9	300	40	260	1000	1	100	30000	0	0	80	100	5	10	12345678
10	400	40	360	5000	1	100	40000	0	0	80	100	5	10	12345678
11	400	7	370	2500	1	100	40000	0	0	80	100	5	10	12345678
12	400	10	350	2000	1	100	40000	0	0	80	100	5	10	12345678
13	400	20	350	2000	1	100	40000	0	0	80	100	5	10	12345678
14	400	20	350	3000	1	100	40000	0	0	80	100	5	10	12345678
15	200	7	170	500	1	100	2000	0	0	80	100	5	10	12354678
16	200	7	170	1600	1	100	2000	0	0	80	100	50	100	12345678
17	400	7	370	1000	1	100	4000	0	0	80	100	5	10	12345678
18	400	7	370	1000	1	1000	40000	0	0	80	100	5	10	12355678
19	400	7	370	1000	1	100	40000	0	0	80	100	5	10	12345678
20	200	7	170	1000	1	100	20000	0	0	80	0	5	10	12345678
21	400	7	370	2000	1	100	40000	0	0	80	0	6	10	12345678
22	200	7	170	500	1	100	20000	0	0	80	50	5	10	12345678
23	400	7	370	1000	1	100	40000	0	0	80	50	5	10	12345678