Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2006-04. Published by the Egerváry Research Group, Pázmány P. sétány 1/C, H–1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres. ISSN 1587–4451.

An Algorithm for Node-Capacitated Ring Routing

András Frank, Zoltán Király, and Balázs Kotnyek

May 9, 2006

An Algorithm for Node-Capacitated Ring Routing

András Frank*, Zoltán Király**, and Balázs Kotnyek***

Abstract

A strongly polynomial time algorithm is described to solve the node-capacitated routing problem in an undirected ring network.

Keywords: algorithm, routing, ring, multicommodity flow, Fourier-Motzkin elimination

1 Introduction

A *routing* is an (integral) packing of paths in a graph. A routing problem has the following input: a graph, nonnegative *demands* between the pairs of vertices, and *capacities* of some resources (edges or nodes). A solution of the routing problem is a routing where, for each pair of vertices, the number of paths connecting them in the routing is at least the demand, and no resource is used by more paths than its capacity.

Based on a theorem of Okamura and Seymour [4], the half-integral relaxation of the edge-capacitated routing problem in a ring network can be solved in polynomial time. In [1] a sharpening was described that gave rise to a polynomial time algorithm for the routing problem in rings. In [2], the node-capacitated routing problem in ring networks was considered along with its fractional relaxation, the node-capacitated multicommodity flow problem. For the feasibility problem, Farkas' lemma provides a characterization for general undirected graphs asserting, roughly, that there exists such a flow if and only if the so-called distance inequality holds for every choice of distance functions arising from nonnegative node-weights. For ring networks, [2]

*Department of Operations Research and Egerváry Research Group of MTA-ELTE, Eötvös University, Pázmány P. s. 1/C. Budapest, Hungary, H-1117. This research was supported by the Hungarian National Foundation for Scientific Research, OTKA T037547 and by European MCRTN Adonet, Contract Grant No. 504438. The author was also supported by Ericsson Hungary.

**Department of Computer Science, CNL (Communication Networks Lab) and Egerváry Research Group of MTA-ELTE, Eötvös University, Pázmány P. s. 1/C. Budapest, Hungary, H-1117. This research was supported by the Hungarian National Foundation for Scientific Research, OTKA T037547 and by European MCRTN Adonet, Contract Grant No. 504438. The author was also supported by Ericsson Hungary.

***Department of Operations Research and Egerváry Research Group of MTA-ELTE, Eötvös University, Pázmány P. s. 1/C. Budapest, Hungary, H-1117. This research was supported by the Hungarian National Foundation for Scientific Research, OTKA T037547 and by European MCRTN Adonet, Contract Grant No. 504438. The author was also supported by Ericsson Hungary.

improved on this (straightforward) result in two ways. First, independent of the integrality of node-capacities and demands, it was shown that it suffices to require the distance inequality only for distances arising from (0-1-2)-valued node-weights, a requirement called the double-cut condition. Second, for integral node-capacities and demands, the double-cut condition was proved to imply the existence of a half-integral multicommodity flow. An algorithm was also developed in [2] to construct a half-integral routing or a violating double-cut. A half-integral routing could then be used to construct a routing which, however, may have slightly violated the node-capacity constraint: the violation at each node was proved to be at most one. Such a routing may be completely satisfactory from a practical point of view, especially when the capacities are big. Nevertheless the problem to decide algorithmically whether or not there is a routing remained open in [2].

In the present work we develop a strongly polynomial algorithm to construct a routing in a node-capacitated ring network, if there is one. The algorithm is based on a rather tricky reduction to the edge-capacitated routing. The reduction can be carried out algorithmically with the help of the Fourier-Motzkin elimination. Though the FM-algorithm is not polynomial in general, we apply it to a specially structured matrix where it is even strongly polynomial. We include this version of the FM-algorithm in Section 6.

These problems may be important from both theoretical and practical point of view. For example, telecommunication network routing problems form the main source of practical demand of this type of problem. In particular, the present work was originally motivated by engineering investigations in the area of passive optical networks.

2 Notation

Let $G = (V, E)$ be a simple undirected graph called a *supply* graph and let $H = (V, F)$ be a so-called *demand* graph on the same node set. Suppose that a nonnegative demand value $h(f)$ is assigned to every demand edge $f \in F$ and a nonnegative capacity $g(e)$ is assigned to every supply edge $e \in E$. We will say that an integer-valued function g is **Eulerian** if $d_g(v) := \sum[g(e) : e \text{ incident to } v]$ is even for each node v.

By a path P we mean an undirected graph $P = (U, A)$ where $U = \{u_1, u_2, \ldots, u_n\}$, $A = \{e_1, \ldots, e_{n-1}\}$, and $e_i = u_i u_{i+1}$, $i = 1, \ldots, n-1$. The edge-set A of a path P is denoted by $E(P)$ while its node set by $V(P)$. Nodes u_1 and u_n are called the end-nodes of P while the other nodes of P are called internal nodes and their set is denoted by $I(P)$. We say that a path P connects its end-nodes and that P uses an edge e if $e \in E(P)$.

For a demand edge $f \in F$, let $\mathcal{P}_f$ denote the set of paths of G connecting the end-nodes of f and let $\mathcal{P} := \cup(\mathcal{P}_f : f \in F)$. By a *path-packing* we mean a function $x : \mathcal{P} \to \mathbb{R}_+$. A path P for which $x(P) > 0$ is said to belong to or determined by x. We say that x fulfills or meets the demand if

$$\sum[x(P) : P \in \mathcal{P}_f] = h(f)$$

holds for every $f \in F$. The occupancy $o_x(e)$ and $o_x(v)$ of a supply edge $e \in E$ and of a node $v \in V$ is defined, respectively, by

$$o_x(e) := \sum [x(P) : P \in \mathcal{P},\ e \in E(P)] \quad \text{and}$$

$$o_x(v) := \sum [x(P) : P \in \mathcal{P},\ v \in I(P)].$$

We stress that the paths ending at v are not counted in the occupancy of v. A path-packing x is called feasible with respect to the edge-capacity, or, in short, **edge-feasible** if

$$o_x(e) \leq g(e) \tag{1}$$

holds for every supply edge e.

Sometimes we are given a capacity function $c : V \to \mathbb{R}_+$ on the node set V rather than on E. A path-packing x is called feasible with respect to the node-capacity, in short, **node-feasible** if

$$o_x(v) \leq c(v) \tag{2}$$

holds for every node $v \in V$. Inequality (1) and (2) are called, respectively, the edge- and the node-capacity constraints.

The edge- or node-capacitated multicommodity flow problem, respectively, consists of finding an edge- or node-feasible path-packing fulfilling the demand. It is sometimes called the *fractional routing* problem. If x is required to be integer-valued, we speak of an integer multicommodity flow problem or a **routing** problem. If $2x$ is required to be integer-valued, we speak of a half-integral multicommodity flow problem or a *half-integral routing* problem. If each demand and each capacity is one, and x is also required to be (0–1)-valued, then we speak of an edge-disjoint or node-disjoint paths problem. That is, the edge-disjoint (node-disjoint) paths problem can be formulated as deciding if there is a path in G for each demand edge $f \in F$ connecting its end-nodes so that these $|F|$ paths are edge-disjoint (internally node-disjoint).

By a **ring** we mean an undirected graph $G = (V, E)$ where $V = \{v_1, v_2, \ldots, v_n\}$, $E = \{e_1, \ldots, e_n\}$, and $e_i = v_i v_{i+1}$, $i = 1, \ldots, n$. [Notation: $v_{n+1} = v_1$.] We will intuitively think that nodes of G are drawn in the plane in a counter-clockwise cyclic order. Note that for rings $\mathcal{P}_f$ has only two paths for any $f \in F$. The edge-capacitated half-integral version was solved earlier by Okamura and Seymour [4], while its integer-valued counter-part by the first named author of the present work [1]. (Actually, both results concerned graphs more general than rings.)

The node-capacitated routing problem for rings is the main concern of the present work. The solvability of the half-integral routing problem was completely characterized in [2]. This gave rise to a sufficient condition for the solvability of the routing problem, while the one of finding a necessary and sufficient condition and of constructing an algorithm to compute a routing, if one exists, remained open. In this paper, based on some of the ideas of [2], we develop a fully combinatorial, strongly polynomial solution algorithm running in $O(n^3)$. Although a good characterization may be derived with the help of the algorithm, its present form is not particularly attractive and finding an appropriate simplification requires further research.

Note that the node-capacitated routing problem is more general than the edge-capacitated one. To see this, put a new node on each edge, and assign the original capacity of the edge to this new node.

3 The cut condition for edge-capacitated routing

For a subset $X \subset V$, the set of edges of G with exactly one end-node in X is called a cut of G. The *capacity* of a cut (with respect to a capacity function g) is the sum of the capacities of supply edges in the cut and is denoted by $d_g(X)$. The *demand* on a cut (with respect to a demand function h) is the sum of the demands of demand edges with exactly one end-node in X and is denoted by $d_h(X)$.

A conceptually easy necessary condition for the solvability of the edge-capacitated multicommodity flow problem is the so-called *edge-cut criterion* which requires that the demand on any cut cannot exceed its capacity, that is,

$$d_g(X) \geq d_h(X) \tag{3}$$

for all subset $X \subset V$. Specializing the theorem of H. Okamura and P. D. Seymour [4] for rings, it states that if $g + h$ is Eulerian then this condition is also sufficient for the solvability of the routing problem, even if it is required only for simple cuts.

THEOREM 1 (Okamura and Seymour). *If G is a ring, g and h are integer-valued and $g + h$ is Eulerian, then the edge-capacitated routing problem has a solution if and only if the* **cut condition**

$$\begin{aligned} g(e_i) + g(e_j) \geq L_h(e_i, e_j) \\ \text{for all } 1 \leq i < j \leq n \end{aligned} \tag{4}$$

holds, where $L_h(e_i, e_j)$ denotes the total demand on the cut determined by e_i and e_j.

Note that there are polynomial algorithms [6][7][5] for this edge-capacitated ring routing problem. Based on the algorithm in [6], the second named author developed an algorithm [3] with running time $O(n^2)$. Our method makes use of such an algorithm as a subroutine.

4 Node-capacitated ring routing

Suppose that $G = (V, E)$ is a ring endowed with an integral node-capacity function $c : V \to \mathbb{Z}_+$. We assume that the demand graph $H = (V, F)$ is complete. Let $h : F \to \mathbb{Z}_+$ be a nonnegative integral demand function. Our main goal is to describe an algorithm to construct a node-capacitated routing fulfilling the demands, if one exists.

It is well-known that in digraphs the node-disjoint version of the Menger's theorem can be derived form its edge-disjoint counterpart by a straightforward node-splitting technique. Therefore it is tempting to try to reduce the node-capacitated ring-routing problem to its well-solved edge-capacitated version. A transformation, however, which is as simple as the one in the Menger's theorem, is unlikely to exist since in the edge-capacitated multicommodity flow problem in rings the cut condition is necessary and sufficient for the solvability, while the node-capacitated version was shown in [2] to require a significantly more complex necessary and sufficient condition, the so-called *double-cut condition*. (A double-cut consists of a family of nodes, in which each node is allowed to occur in two copies.) Still, the approach of [2] showed that such a reduction, though not so cheaply, is possible. A further refinement of that approach will be the key to the present algorithm.

We start with an easy simplification that has appeared already in [2]. For completeness, we include its proof.

Claim 2. *Given c and h, there is an Eulerian demand function h' so that the routing problem is solvable with respect to c and h if and only if it is solvable with respect to c and h'.*

Proof. Let v_i and v_{i+1} be two subsequent nodes in the ring. As a one-edge path connecting v_i and v_{i+1} has no inner node, increasing the demand between them does not affect the solvability of the routing problem.

If $d_h(v)$ is not even everywhere, then there are two nodes v_i and v_j so that both $d_h(v_i)$ and $d_h(v_j)$ are odd and so that $d_h(v_l)$ is even for each l with $i < l < j$. We can then increase by one the h-values on all demand edges $v_iv_{i+1},\ v_{i+1}v_{i+2}\ ,\dots,\ v_{j-1}v_j$ and this way both $d_h(v_i)$ and $d_h(v_j)$ get even while all other $d_h(v_l)$-values remain even for $i < l < j$. By repeating this procedure, we obtain the desired Eulerian demand function h'. •

Therefore we will assume throughout that

$$h \text{ is Eulerian.} \tag{5}$$

Note that for an Eulerian demand function the demand on every cut is even.

THEOREM 3. *Let h be Eulerian and $g : E \to \mathbb{Z}_+$ an integral edge-capacity function such that g is Eulerian, g satisfies the cut condition and*

$$g(e_{i-1}) + g(e_i) \leq d_h(v_i) + 2c(v_i) \quad \text{for every } v_i \in V. \tag{6}$$

Then there is an edge-feasible routing (with respect to g) meeting the demand h, and every such routing is node-feasible. Conversely, if there is a node-feasible routing that meets h, then there is an Eulerian function $g : E \to \mathbb{Z}_+$ satisfying the cut condition and (6).

Proof. Let us start with the easier second part and let x be a node-feasible routing. For $e \in E$, define $g(e) := o_x(e)$, that is, intuitively, $g(e)$ denotes the number of paths using e. Obviously g satisfies the cut condition (as x is an edge-feasible routing with respect to g). Let v_i be any node of G and remember that $o_x(v_i) = \sum[x(P) : P \in \mathcal{P},\ \ v_i \in I(P)]$. The paths counted in this sum use both e_i and e_{i-1} while the paths determined by x that end at v_i use exactly one of e_i and e_{i-1}. Therefore $g(e_{i-1}) + g(e_i) = d_h(v_i) + 2o_x(v_i)$, and this implies (6) since $o_x(v_i) \leq c(v_i)$. Note that since $d_h(v_i)$ is even it also follows that $g(e_{i-1}) + g(e_i)$ is even.

To see the first part, let g be a function satisfying the hypothesis of the theorem. As both g and h are Eulerian, so is $g+h$, and hence Theorem 1 ensures the existence of an edge-feasible routing x (with respect to edge-capacity function g.) We want to show that x is node-feasible as well, with respect to node-capacity function c. To see this, let us consider an arbitrary node v_i.

Let α denote the sum of x-values of those paths ending at v_i and using e_i, and let β denote the sum of x-values of those paths ending at v_i and using e_{i-1}. Finally, let γ be the sum of x-values of those paths using v_i as an inner node.

Then $\alpha + \beta = d_h(v_i)$. We have $\gamma + \alpha \leq g(e_i)$ and $\gamma + \beta \leq g(e_{i-1})$. By combining these with (6), we obtain $\alpha+\beta+2\gamma \leq g(e_{i-1})+g(e_i) \leq d_h(v_i)+2c(v_i) \leq \alpha+\beta+2c(v_i)$, from which $\gamma \leq c(v_i)$ follows, that is, x is indeed node-feasible with respect to c. •

By Theorem 3, the node-capacitated ring-routing problem can be reduced to the edge-capacitated case if one is able to compute the capacity function g described in the theorem. To this end, introduce a nonnegative variable $z(i)$ for each $g(e_i)$. Consider the inequality system

$$z(i-1) + z(i) \leq d_h(v_i) + 2c(v_i) \quad \text{for every } v_i \in V \tag{7}$$

and

$$z(i) + z(j) \geq L_h(e_i, e_j) \quad \text{for every } 1 \leq i < j \leq n. \tag{8}$$

The parity requirement in Theorem 3 may be satisfied in two ways, since every righthand-side is even. Either every $z(i)$ is even or else every $z(i)$ is odd. The algorithm handles these alternatives separately.

The system (7) and (8) has a solution so that z is even-integer-valued if and only if the system

$$z'(i-1) + z'(i) \leq (d_h(v_i) + 2c(v_i))/2 \quad \text{for every } v_i \in V \tag{9}$$

and

$$z'(i) + z'(j) \geq L_h(e_i, e_j)/2 \quad \text{for every } 1 \leq i < j \leq n \tag{10}$$

has a nonnegative integral solution, and $z(i) = 2z'(i)$.

The system (7) and (8) has a solution so that z is odd-integer-valued if and only if the system

$$z''(i-1) + z''(i) \leq (d_h(v_i) + 2c(v_i) - 2)/2 \quad \text{for every } v_i \in V \tag{11}$$

and

$$z''(i) + z''(j) \geq (L_h(e_i, e_j) - 2)/2 \quad \text{for every } 1 \leq i < j \leq n \tag{12}$$

has a nonnegative integral solution, and $z(i) = 2z''(i) + 1$.

Both the inequality system described for z' and the one for z'' have the following form. The right-hand side is integral and each inequality contains at most two variables. In each inequality the coefficients of the variables has absolute value one. The integral solvability of such an inequality system can be decided in strongly polynomial time by a straightforward modification of the Fourier-Motzkin elimination algorithm. Therefore with two separate applications of the FM-algorithm we are able to decide if any of the systems {(9),(10)} and {(11),(12)} has a nonnegative integral solution and compute one if it exists. If none of these systems has such a solution then we may conclude that the original node-capacitated ring-routing problem has no solution. If one of them has a solution, we can calculate the appropriate (either odd-integer-valued or even-integer-valued) vector z satisfying inequalities (7) and (8), and determine a routing with respect to edge-capacities $g(e_i) = z(i)$. We may use the $O(n^2)$ algorithm of [3] for this purpose, so together with the FM algorithm described in the next sections the total running time is $O(n^3)$. By Theorem 3 such an edge-feasible routing exists and it is also node-feasible.

For completeness, we include the original FM-algorithm and then we derive its strongly polynomial variation for "simple" matrices.

5 Fourier-Motzkin elimination

The Fourier-Motzkin elimination is a finite algorithm to find a solution to a linear inequality system $Qx \leq b$, that is, to find an element of the polyhedron $R := \{x \in \mathbb{R}^n : Qx \leq b\}$. It consists of two parts. In the first part, it eliminates the components of x one by one by creating new inequalities. In the second part, it proceeds backward and computes the components of a member of R. Geometrically, one elimination step may be interpreted as determining the polyhedron obtained from R by projecting along a coordinate axis.

Let Q be an $m \times n$ matrix $(m \geq 1,\ n \geq 2)$. For any index set L of the rows of Q let $_LQ$ denote the corresponding submatrix of Q. The i'th row of Q is denoted by $_iq$. In order to find a solution to the system $Qx \leq b$, we may assume that the first column q_1 of Q is $(0, \pm 1)$-valued since multiplying an inequality by a positive constant does not effect the solution set. Let I, J, K denote the index sets of rows of Q for which the value $q_1(i)$ is $+1, -1$ or 0, respectively. Define a matrix $Q^{[1]}$ which contains all rows of ${}_KQ$, as follows. For every choice of indices $i \in I$ and $j \in J$, let ${}_iq +_j q$ be a row of $Q^{[1]}$ and let this row be denoted by ${}_{[ij]}q$. This means that in case I or J is empty, $Q^{[1]}$ is simply ${}_KQ$. In general $Q^{[1]}$ has $m - (|I| + |J|) + |I||J|$ rows. Note that the first column of $Q^{[1]}$ consists of zeros. The right-hand side vector $b^{[1]}$ is obtained analogously from b. Let $R^{[1]} := \{x : \ x(1) = 0, \ Q^{[1]}x \leq b^{[1]}\}$.

Proposition 4. *The projection of R along the first coordinate is $R^{[1]}$, that is, by turning zero the first component of any solution to*

$$Qx \leq b \tag{13}$$

yields a solution to

$$Q^{[1]}x \leq b^{[1]}, \tag{14}$$

and conversely, the first coordinate of any solution to (14) *may be suitably changed in order to get a solution to* (13).

Proof. The first part follows directly from the construction of $Q^{[1]}$ and $b^{[1]}$ since every row of $(Q^{[1]}, b^{[1]})$ is a nonnegative combination of the rows of (Q, b).

To see the second part, let z be a solution to (14). For a number α, let z_α denote the vector arising from z by changing its first component to α. If J is empty, that is, the first column of Q has no negative element, then by choosing α small enough, z_α will be an element of R. (Namely, $\alpha := \min_{i \in I}\{b(i) -_i q \cdot z\}$ will do.) Analogously, if I is empty, then α may be chosen suitably large. Suppose now that neither I nor J is empty. For any $i \in I, \ j \in J, \quad {}_iq \cdot z + {}_jq \cdot z = \ {}_{[ij]}q \cdot z \leq b(i) + b(j)$ implies that ${}_jq \cdot z - b(j) \leq b(i) - {}_iq \cdot z$ and hence

$$\max_{j \in J}\{{}_jq \cdot z - b(j)\} \leq \min_{i \in I}\{b(i) - {}_iq \cdot z\}. \tag{15}$$

Therefore there is a number α with

$$\max_{j \in J}\{{}_jq \cdot z - b(j)\} \leq \alpha \leq \min_{i \in I}\{b(i) - {}_iq \cdot z\}. \tag{16}$$

We claim that vector z_α is a solution to (13). Indeed, for an index $h \in K$, the first component of ${}_hq$ is zero, so ${}_hq \cdot z_\alpha = \ {}_hq \cdot z \leq b(h)$. If $h \in I$, that is, ${}_hq(1) = 1$, then the second inequality in (16) implies ${}_hq \cdot z_\alpha = \ {}_hq \cdot z + \alpha \leq b(h)$. Finally, if $h \in J$, that is, ${}_hq(1) = -1$, then the first inequality of (16) implies ${}_hq \cdot z_\alpha = \ {}_hq \cdot z - \alpha \leq b(h)$. •

6 Integral Fourier-Motzkin elimination for simple matrices

Let us call a $(0, \pm 1)$-valued matrix Q *simple* if each row contains one or two nonzero entries and the rows are distinct. Note that a simple matrix with n columns can have at most $2n + 2\binom{n}{2} + n(n-1) = 2n^2$ rows.

We show how the FM-elimination above may easily be turned to a strongly polynomial algorithm for computing an integral solution of $Qx \leq b$ in case Q is simple and b is integral.

As Q is simple, each row of $Q^{[1]}$ has at most two nonzero entries. If a row of $Q^{[1]}$ has exactly one nonzero element, then this element is ± 2 or ± 1, while a row with two nonzero entries is $(0, \pm 1)$-valued. $Q^{[1]}$ is not necessarily simple but it may easily be simplified without changing the integral solution set as follows.

First replace any inequality of type $2x(i) \leq \beta$ or $-2x(i) \leq \beta$ by $x(i) \leq \lfloor \beta/2 \rfloor$ or $-x(i) \leq \lfloor \beta/2 \rfloor$, respectively. Then, if some inequalities with identical lefthand sides remained, then we keep only the one defining the strongest inequality (that is, for which the corresponding righthand side is the smallest). Let $Q^{[2]}$ and $b^{[2]}$ denote the derived lefthand and righthand sides respectively.

Proposition 5. *Any integral solution to*

$$Qx \leq b \tag{17}$$

yields an integral solution to

$$Q^{[2]}x \leq b^{[2]}, \tag{18}$$

and conversely, the first coordinate of any integral solution to (18) *may be suitably changed to another integer* α *in order to get an integral solution to* (17).

Proof. We follow the lines of the proof of Proposition 4. The first part follows again from the construction. For the second part, observe that as the righthand and lefthand side of inequality (16) are now integers, we may choose α to be also an integer. •

This approach clearly gives rise to a strongly polynomial algorithm. With some care the following estimation can be given for the running time.

Proposition 6. *Calculating a suitable representation of* $Q^{[2]}$ *and* $b^{[2]}$ *as well as determining the appropriate* α *value takes* $O(n^2)$ *steps. This yields an* $O(n^3)$ *algorithm for finding an integral solution to* $Qx \leq b$ *if* Q *is simple, and if such a solution exists.*

As Q may have $2n^3$ entries, for calculating $Q^{[2]}$ in $O(n^2)$ steps we really need a more compact representation. Instead of storing Q and b, we store three n-by-n matrices PP, MM and PM. Set $PP(k,l) = b(i)$ if there is a row ${}_iq$ in Q, where $Q(i,k) = Q(i,l) = 1$ and consequently $Q(i,j) = 0$ for $k \neq j \neq l$; and set $PP(k,l) = \infty$ if no such row exists. Similarly, $MM(k,l) = b(i)$ if there is a row ${}_iq$ in Q, where $Q(i,k) = Q(i,l) = -1$ (and set $MM(k,l) = \infty$ otherwise); and $PM(k,l) = b(i)$ if there is a row ${}_iq$ in Q, where $Q(i,k) = 1,\ Q(i,l) = -1$ (and set $PM(k,l) = \infty$ otherwise). Observe, for example, that indices in J can be determined by examining the first column of MM and of PM. Using these quantities, $PP^{[2]}, MM^{[2]}$ and $PM^{[2]}$ corresponding to $Q^{[2]}$ and $b^{[2]}$ can be easily calculated in $O(n^2)$ time. First, initialize each entry to ∞ and copy the values corresponding to inequalities in K from matrices PP, MM and PM; and observe that the first rows and columns are irrelevant (they will be deleted before the next elimination). Then for all (at most $4n^2$) values $i \in I, j \in$

J, calculate (in constant time) the appropriate row of $Q^{[1]}$ and then the appropriate row of $Q^{[2]}$. Check the corresponding entry of $PP^{[2]}, MM^{[2]}$ or $PM^{[2]}$, and, if the resulting righthand side is smaller than the one stored there, then replace it.

Determining α is straightforward as using the first row and column of the stored matrices PP, MM and PM, the left- and righthand side of inequality (16) can be calculated in $O(n)$ time.

Summarizing, the overall complexity of the integral FM-elimination for simple matrices can be carried out in $O(n^3)$ steps, as we need n (the number of variables) eliminations. •

References

[1] A. Frank, *Edge-disjoint paths in planar graphs*, J. of Combinatorial Theory, Ser. B. No. 2 (1985), pp. 164-178.

[2] A. Frank, B. Shepherd, V. Tandon, and Z. Végh, *Node-capacitated ring routing,* Mathematics of Operations Research, 27, 2. (2002) pp. 372-383.

[3] Z. Király, *An $O(n^2)$ algorithm for ring routing,* Egres Technical Reports, TR-2005-10, see `http://www.cs.elte.hu/egres/`, (2005).

[4] H. Okamura and P.D. Seymour, *Multicommodity flows in planar graphs,* J. Combinatorial Theory, Ser. B, 31 (1981) pp. 75-81.

[5] H. Ripphausen-Lipa, D. Wagner and K. Weihe, *Survey on Efficient Algorithms for Disjoint Paths Problems in Planar Graphs*, DIMACS-Series in Discrete Mathematics and Theoretical Computer Science, Volume 20 on the "Year of Combinatorial Optimization" (W. Cook and L. Lovász and P.D. Seymour eds.), AMS (1995) pp. 295-354.

[6] A. Schrijver, P. Seymour, and P. Winkler, *The ring loading problem*, SIAM J. Discrete Math. Vol. 11, No 1. (1998) pp. 1-14.

[7] D. Wagner and K. Weihe, *A linear-time algorithm for edge-disjoint paths in planar graphs*, Combinatorica 15 (1995) pp. 135-150.