Journal of Interconnection Networks
Vol. 5, No. 2 (2004) 193–200

World Scientific
www.worldscientific.com

POLYNOMIAL TIME SOLVABILITY OF THE WEIGHTED RING ARC-LOADING PROBLEM WITH INTEGER SPLITTING

JINJIANG YUAN*
Department of Mathematics
Zhengzhou University
Zhengzhou, Henan 450052, P. R. China
Email: yuanjj@zzu.edu.cn

SANMING ZHOU†
Department of Mathematics and Statistics
The University of Melbourne
Parkville, VIC 3010, Australia
Email: smzhou@ms.unimelb.edu.au

Received 19 February 2004
Revised 11 June 2004

Abstract

In the Weighted Ring Arc-Loading Problem with Integer Splitting, we are given a set of communication requests each associated with a non-negative integer weight. The problem is to find a routing scheme such that the maximum load on arcs of the ring is minimized, subject to that the weight of each request may be split into two integral parts routed in two directions around the ring, where the load of an arc is the sum of parts routed through the arc. A pseudo-polynomial algorithm for this problem is implied by a result in [G. Wilfong and P. Winkler, Ring routing and wavelength translation, Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, 1998, 333-341]. By refining the rounding technique developed in the same paper, we prove that the problem can be solved in polynomial time.

Keywords: ring; routing; load; polynomial algorithm

1 Introduction

A communication network is usually modelled by a graph, in which nodes represent processors, memory modules or routers and edges represent bidirectional links. Given a network and a set D of communication requests, a fundamental problem is to design a transmission route (directed path) for each request such that high load on arcs/edges is avoided, where an *arc* is an edge endowed with a direction, the load of an arc is the number of routes traversing the arc in its direction, and the load of an edge is the number of routes traversing the edge in either direction. In general, if each request is associated with a non-negative integer *weight*, then the load of an arc is defined to be the total weight of those requests that are routed

*Supported by the National Natural Science Foundation of China under grant number 10371112.
†Correspondence author. Supported by a Discovery Project Grant (DP0344803) from the Australian Research Council.

through the arc in its direction, and the load of an edge is defined similarly. Practically, the weight of a request can be interpreted as the traffic demand or the size of the data to be transmitted. In both weighted and non-weighted cases, the *Arc/Edge Loading Problem* asks for a routing scheme such that the maximum load on arcs/edges is minimized. This problem, in both weighted and non-weighted cases, has been studied extensively in recent years, especially in the case where the network is a ring. For example, in [9] and its updated version [10], Schrijver, Seymour and Winkler provided a fast approximation algorithm to solve the Weighted Edge-Loading Problem for rings in the case where D is all-to-all. In [7] Myung gave a polynomial time algorithm to solve the Weighted Edge-Loading Problem for rings in the case where D is arbitrary and demands can be integrally split and routed in two directions around the ring. In [11] Wilfong and Winkler proved that, for an arbitrary set D of requests, the Non-weighted Arc-Loading Problem for rings can be solved in polynomial time. On the contrary, the Weighted Arc-Loading Problem is NP-hard [1, Theorem 9] and a polynomial-time approximation scheme (PTAS) has been obtained in [1, Theorem 13]. For more results on the ring loading problem, the reader is referred to [1, 2, 3, 5, 6, 8].

In this paper we study the Weighted Arc-Loading Problem for rings with integer splitting, that is, the weight of each request may be split into two integral parts routed clockwise and anticlockwise around the ring. By refining the rounding technique developed by Wilfong and Winkler [11, Section 2], we prove that this problem can be solved in polynomial time. We notice that this result is *not* implied by [11, Theorem 2.2]. Indeed, the method of [11, Section 2] applies to weighted case if we take a request (s_i, t_i) (where s_i is the source and t_i is the destination) with weight w_i as w_i non-weighted requests. Nevertheless, it gives a pseudo-polynomial algorithm but not a polynomial algorithm. This is because the linear programming relaxation in [11, Section 2] has $\sum_{i=1}^{m} w_i$ variables, but $\sum_{i=1}^{m}(\lceil \log w_i \rceil + 1)$ would contribute to the input size of our problem, where m is the number of requests. Also, in the proof of [11, Proposition 2] it may need $\sum_{i=1}^{m} w_i$ steps to get the desired flush routing.

To prove our main result, Theorem 1 in the next section, we will take (s_i, t_i) as one request instead of w_i requests. By doing so the linear programming involved will have m variables, and hence is solvable in time polynomial in m. Also, we need to examine only at most m pairs of parallel requests to obtain a new routing with desired property, see Lemma 3. However, with this treatment we have to deal with the complication in turning the new routing into an optimal routing, see the proof of Theorem 1. The polynomial time algorithm given in this paper is very much needed in practice since SONET (synchronous optical network) rings are widely used configuration nowadays in telecommunication.

2 Main result and notation

Let C_n be an n-node ring with nodes $v_0, v_1, ..., v_{n-1}$ labelled clockwise. We will view C_n as bidirectional, that is, each edge $\{v_k, v_{k+1}\}$ of C_n, $0 \le k \le n-1$, is taken as two *arcs*

$$a_k^+ = (v_k, v_{k+1}), \;\; a_k^- = (v_{k+1}, v_k)$$

with opposite directions, and data streams can transmit in either direction. *Here and in the following subscripts are taken modulo n.* A *request* on C_n is an ordered pair (s, t) of distinct nodes of C_n; it corresponds to a data stream to be sent from the *source* s to the *destination* t. We assume that data can be transmitted clockwise or anticlockwise on the ring. We will use $P^+(s,t)$ to denote the directed (s,t)-path clockwise around C_n, and $P^-(s,t)$ the directed (s,t)-path anticlockwise around C_n. Often a request (s,t) is associated with an integer weight $w \geq 0$; we denote this weighted request by $(s,t;w)$. Let

$$D = \{(s_1,t_1;w_1),(s_2,t_2;w_2),\ldots,(s_m,t_m;w_m)\}$$

be a set of integrally weighted requests on C_n. We assume that weights may be *split*, that is, for some integer x_i with $0 \leq x_i \leq w_i$, x_i amount of data is transmitted along $P^+(s_i,t_i)$ and the remaining $w_i - x_i$ amount is transmitted along $P^-(s_i,t_i)$. The vector $\mathbf{x} = (x_1,x_2,\ldots,x_m)$ determines uniquely a routing scheme for D, and vice versa. In the following we will call $\mathbf{x}$ an *integral routing* for the given request set D. In general, any real-valued vector $\mathbf{x} = (x_1,x_2,\ldots,x_m)$ with $0 \leq x_i \leq w_i$ for each i will be called a (generalized) *routing* for D. For each $k = 0,1,\ldots,n-1$, the *loads* on a_k^+ and a_k^- under $\mathbf{x}$ are defined to be

$$L(\mathbf{x},a_k^+) = \sum_{i:\ a_k^+ \in P^+(s_i,t_i)} x_i \tag{1}$$

$$L(\mathbf{x},a_k^-) = \sum_{i:\ a_k^- \in P^-(s_i,t_i)} (w_i - x_i) \tag{2}$$

respectively, where $a_k^+ \in P^+(s_i,t_i)$ means that $P^+(s_i,t_i)$ passes through a_k^+ and the notation $a_k^- \in P^-(s_i,t_i)$ is interpreted similarly. The maximum load on arcs of C_n, namely,

$$L(\mathbf{x}) = \max\left\{\max_{0\leq k\leq n-1} L(\mathbf{x},a_k^+),\ \max_{0\leq k\leq n-1} L(\mathbf{x},a_k^-)\right\} \tag{3}$$

is called the *load* of $\mathbf{x}$.

WEIGHTED RING ARC-LOADING PROBLEM WITH INTEGER SPLITTING (WRALP)

Instance Ring C_n and set $D = \{(s_1,t_1;w_1),(s_2,t_2;w_2),\ldots,(s_m,t_m;w_m)\}$ of integrally weighted requests on C_n.

Objective Find an integral routing $\mathbf{x}$ for D such that $L(\mathbf{x})$ is minimized.

The main result of this paper is the following theorem.

Theorem 1 *The* WEIGHTED RING ARC-LOADING PROBLEM WITH INTEGER SPLITTING *can be solved in polynomial time.*

3 Proof of Theorem 1

Clearly, the WEIGHTED RING ARC-LOADING PROBLEM WITH INTEGER SPLITTING can be formulated as the following integer linear programming:

$$\text{WRALP:}\ \begin{cases} \min\ L(\mathbf{x}) \\ 0 \leq x_i \leq w_i,\ \ 1 \leq i \leq m \\ x_i \text{ an integer},\ \ 1 \leq i \leq m. \end{cases}$$

To solve WRALP we will need an optimal solution of its linear programming relaxation:

$$\text{LPR:}\ \begin{cases} \min\ L(\mathbf{x}) \\ 0 \le x_i \le w_i,\ \ 1 \le i \le m. \end{cases}$$

Denote $W = \sum_{1\le i\le m} w_i$. Let α be a real number with $0 \le \alpha \le W$, and let LPR_α stand for the following linear programming:

$$\text{LPR}_\alpha\text{:}\ \begin{cases} \min\ L(\mathbf{x}) \\ \sum_{1\le i\le m} x_i = \alpha \\ 0 \le x_i \le w_i,\ \ 1 \le i \le m. \end{cases}$$

By the polynomial time solvability of linear programming [4], both LPR and LPR_α (for any rational number α) can be solved in polynomial time. In the following we will use L_{OPT}, L_* and L_α to denote the optimal objective values of WRALP, LPR and LPR_α, respectively.

Lemma 1 *As a function of α, L_α is convex on the interval* $[0, W]$.

Proof Let $\alpha, \beta \in [0, W]$, and let $\mathbf{x} = (x_1, x_2, ..., x_m)$ and $\mathbf{y} = (y_1, y_2, ..., y_m)$ be optimal solutions of LPR_α and LPR_β respectively. For any real number λ with $0 \le \lambda \le 1$ and each $k = 0, 1, \ldots, n-1$, we have $L(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}, a_k^+) = \lambda L(\mathbf{x}, a_k^+) + (1-\lambda)L(\mathbf{y}, a_k^+)$ by (1) and $L(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}, a_k^-) = \lambda L(\mathbf{x}, a_k^-) + (1-\lambda)L(\mathbf{y}, a_k^-)$ by (2). Thus, from (3) and by noting $L(\mathbf{x}) = L_\alpha$ and $L(\mathbf{y}) = L_\beta$, we have

$$\begin{aligned} L(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) &\le \lambda L(\mathbf{x}) + (1-\lambda)L(\mathbf{y}) \\ &= \lambda L_\alpha + (1-\lambda)L_\beta. \end{aligned}$$

Since $\lambda\mathbf{x} + (1-\lambda)\mathbf{y}$ is a feasible solution of $\text{LPR}_{\lambda\alpha+(1-\lambda)\beta}$, it follows that

$$L_{\lambda\alpha+(1-\lambda)\beta} \le \lambda L_\alpha + (1-\lambda)L_\beta$$

and hence L_α is convex as claimed. □

Note that feasible solutions of LPR are exactly routings for D. Following [11], a feasible solution $\mathbf{x} = (x_1, x_2, ..., x_m)$ of LPR is called a *flush routing* if $\sum_{1\le i\le m} x_i$ is an integer.

Lemma 2 *Let $\mathbf{x}^* = (x_1^*, x_2^*, ..., x_m^*)$ be an optimal solution of LPR, and let $\alpha^* = \sum_{1\le i\le m} x_i^*$. Let $\mathbf{x}$ be an optimal solution of $LPR_{\lfloor\alpha^*\rfloor}$ if $L_{\lfloor\alpha^*\rfloor} \le L_{\lceil\alpha^*\rceil}$, and an optimal solution of $LPR_{\lceil\alpha^*\rceil}$ otherwise. Then $\mathbf{x}$ is a flush routing and $L(\mathbf{x}) \le L_{\text{OPT}}$.*

Proof Clearly, $\mathbf{x}^*$ is an optimal solution of LPR_{α^*}. So we have

$$L_{\alpha^*} = L(\mathbf{x}^*) = L_* = \min\{L_\alpha : 0 \le \alpha \le W\}.$$

Since L_α is convex by Lemma 1, this implies

$$\min\{L_{\lfloor\alpha^*\rfloor}, L_{\lceil\alpha^*\rceil}\} = \min\{L_\alpha : \alpha \text{ is an integer and } 0 \le \alpha \le W\}.$$

But $\min\{L_\alpha : \alpha \text{ is an integer and } 0 \le \alpha \le W\} \le L_{\text{OPT}}$, so

$$\min\{L_{\lfloor\alpha^*\rfloor}, L_{\lceil\alpha^*\rceil}\} \le L_{\text{OPT}}$$

and the result follows. □

Since each of LPR, $\text{LPR}_{\lfloor\alpha^*\rfloor}$ and $\text{LPR}_{\lceil\alpha^*\rceil}$ can be solved in polynomial time, an immediate consequence of Lemma 2 is that a flush routing $\mathbf{x}$ with $L(\mathbf{x}) \le L_{\text{OPT}}$ can be found in polynomial time. This will be used in the proof of Theorem 1.

As in [11], two distinct requests (s_i, t_i) and (s_j, t_j) are said to be *parallel* if $P^+(s_i, t_i)$ and $P^-(s_j, t_j)$, or $P^+(s_j, t_j)$ and $P^-(s_i, t_i)$, intersect in at most one node. (In other words, (s_i, t_i) and (s_j, t_j) are parallel if either $P^+(s_j, t_j)$ is contained in $P^+(s_i, t_i)$ or $P^-(s_j, t_j)$ is contained in $P^-(s_i, t_i)$.) A request $(s_i, t_i; w_i)$ is said to be *split* by a routing $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ if $0 < x_i < w_i$.

Lemma 3 *Given a flush routing* $\mathbf{x} = (x_1, x_2, ..., x_m)$, *we can find in polynomial time a flush routing* $\mathbf{y} = (y_1, y_2, ..., y_m)$ *such that* $L(\mathbf{y}) \le L(\mathbf{x})$ *and no two parallel requests are both split by* $\mathbf{y}$.

Proof Suppose (s_i, t_i) and (s_j, t_j) are parallel requests which are both split by $\mathbf{x}$. Without loss of generality we may suppose that the directed paths $P^+(s_i, t_i)$ and $P^-(s_j, t_j)$ intersect in at most one node. If $x_i + x_j \le w_i$, then define

$$y_i = x_i + x_j, \;\; y_j = 0, \;\; y_k = x_k \;\text{ for }\; k \ne i, j;$$

and if $x_i + x_j > w_i$, then define

$$y_i = w_i, \;\; y_j = x_i + x_j - w_i, \;\; y_k = x_k \;\text{ for }\; k \ne i, j.$$

See Figure 1 for an illustration. In both cases, $\mathbf{y} = (y_1, y_2, \ldots, y_m)$ is a flush routing with $\sum_{1 \le i \le m} y_i = \sum_{1 \le i \le m} x_i$, and one of (s_j, t_j) and (s_i, t_i) is not split by $\mathbf{y}$. Also, under $\mathbf{y}$ the arcs in $P^+(s_i, t_i)$ and $P^-(s_j, t_j)$ have the same loads as before and other arcs have the same or reduced loads. Thus, $L(\mathbf{y}) \le L(\mathbf{x})$ and $\mathbf{y}$ splits less number of requests than $\mathbf{x}$. If there exist no parallel requests both split by $\mathbf{y}$, we are done; otherwise repeat the process above. After at most m such processes, we then get the desired flush routing, and the proof is complete. □

Proof of Theorem 1 Recall that the nodes of the ring C_n are labelled by $v_0, v_1, \ldots, v_{n-1}$ clockwise. By Lemma 3 and the comments after the proof of Lemma 2, we can find in polynomial time a flush routing $\mathbf{y} = (y_1, y_2, \ldots, y_m)$ such that $L(\mathbf{y}) \le L_{\text{OPT}}$ and no two parallel requests are both split by $\mathbf{y}$. In particular, for each $k = 0, 1, \ldots, n-1$, at most one request with source v_k is split. Hence the number of requests split by $\mathbf{y}$, which we denote by q, is at most n. Without loss of generality, we may suppose that the requests split by $\mathbf{y}$ are

$$(s_1, t_1; w_1), (s_2, t_2; w_2), \ldots, (s_q, t_q; w_q)$$

and that the sources $s_1, s_2, \ldots, s_q$ of them are ordered clockwise on the ring C_n. By the above-mentioned properties of $\mathbf{y}$, any two of such requests are not parallel. Thus, the destinations $t_1, t_2, \ldots, t_q$ of them

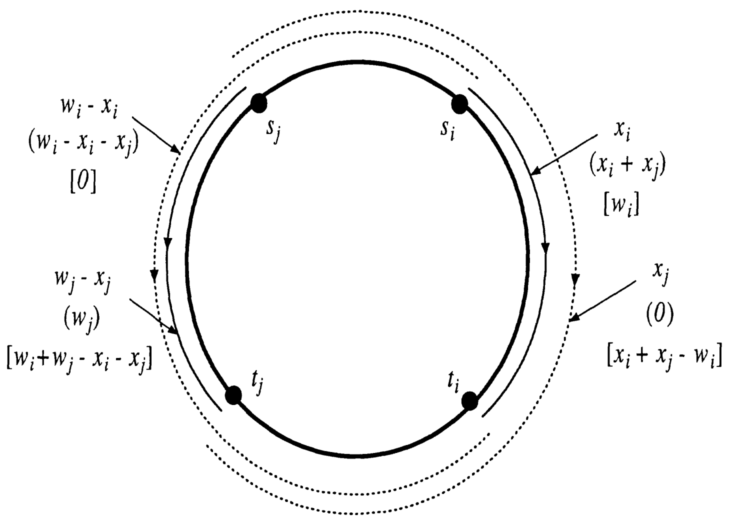

Figure 1: Proof of Lemma 3. The coordinates of $\mathbf{y}$ are in parentheses (when $x_i + x_j \leq w_i$) and brackets (when $x_i + x_j > w_i$).

must be in clockwise order as well. For i, j with $1 \leq i, j \leq q$, denote

$$[i,j]_q = \begin{cases} \{i, i+1, \ldots, j\}, & \text{if } i \leq j \\ \{1, 2, \ldots, q\} \setminus \{j+1, j+2, \ldots, i-1\}, & \text{if } i > j. \end{cases}$$

For each $k = 0, 1, \ldots, n-1$, if there exists i with $1 \leq i \leq q$ such that $a_k^+ \in P^+(s_i, t_i)$, then define k^+ (k^-, respectively) to be such a subscript i such that the length of $P^+(s_i, v_k)$ is maximum (minimum, respectively). See Figure 2 for an illustration. (Note that k^+ can be greater than, less than, or equal to k^-. For example, if t_q is between s_1 and t_1 and $v_k \neq t_q$ is between s_1 and t_q, then we have $k^+ > k^-$.) If there exists no i with $1 \leq i \leq q$ such that $a_k^+ \in P^+(s_i, t_i)$, then we simply define $[k^+, k^-]_q = \emptyset$. This latter case occurs precisely when, for some i, s_{i+1} is after t_i and $v_k \neq s_{i+1}$ is between t_i and s_{i+1} (subscripts of s and t modulo q) with respect to the cyclic order $v_0, v_1, \ldots, v_{n-1}$ of the nodes of C_n. In both cases, since the sources and the destinations of the requests split by $\mathbf{y}$ are ordered in the same direction, for each $i = 1, 2, \ldots, q$, we have

$$\begin{aligned} i \in [k^+, k^-]_q &\Longleftrightarrow a_k^+ \in P^+(s_i, t_i) \\ i \notin [k^+, k^-]_q &\Longleftrightarrow a_k^- \in P^-(s_i, t_i). \end{aligned} \tag{4}$$

Let $z_1 = \lfloor y_1 \rfloor$ if $y_1 - \lfloor y_1 \rfloor \leq 1/2$ and $z_1 = \lfloor y_1 \rfloor + 1$ otherwise, and define recursively

$$z_j = \begin{cases} \lfloor y_j \rfloor, & \text{if } 1 \leq j \leq q \text{ and } y_i - \lfloor y_i \rfloor - \sum_{1 \leq i \leq j-1}(z_i - y_i) \leq \frac{1}{2} \\ \lfloor y_j \rfloor + 1, & \text{if } 1 \leq j \leq q \text{ and } y_i - \lfloor y_i \rfloor - \sum_{1 \leq i \leq j-1}(z_i - y_i) > \frac{1}{2} \\ y_j, & \text{if } q+1 \leq j \leq m. \end{cases} \tag{5}$$

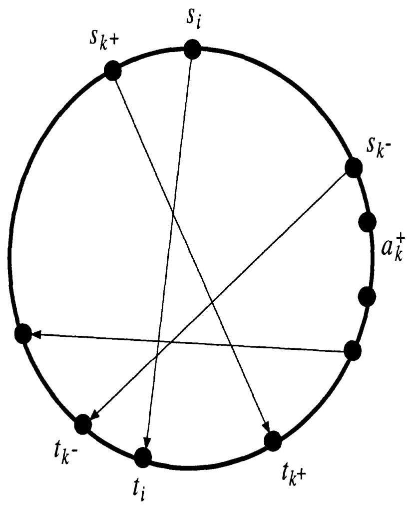

Figure 2: Proof of Theorem 1.

Then $\mathbf{z} = (z_1, z_2, \dots, z_m)$ is an integral routing for D. By induction one can prove that

$$-\frac{1}{2} \leq \sum_{1 \leq i \leq j} (z_i - y_i) < \frac{1}{2} \tag{6}$$

for each $j = 1, 2, \dots, q$. In particular, $-1/2 \leq \sum_{1 \leq i \leq q} (z_i - y_i) < 1/2$. However, $\sum_{1 \leq i \leq q} (z_i - y_i)$ is an integer because both $\mathbf{y}$ and $\mathbf{z}$ are flush routings. So we must have

$$\sum_{1 \leq i \leq q} (z_i - y_i) = 0. \tag{7}$$

For each $k = 0, 1, \dots, n-1$, from (4) we have

$$L(\mathbf{z}, a_k^+) = L(\mathbf{y}, a_k^+) + \sum_{i:\ i \in [k^+, k^-]_q} (z_i - y_i)$$

$$L(\mathbf{z}, a_k^-) = L(\mathbf{y}, a_k^-) + \sum_{i:\ i \notin [k^+, k^-]_q} (y_i - z_i).$$

From this and by using (7) we obtain

$$L(\mathbf{z}, a_k^+) - L(\mathbf{y}, a_k^+) = L(\mathbf{z}, a_k^-) - L(\mathbf{y}, a_k^-) = \sum_{i:\ i \in [k^+, k^-]_q} (z_i - y_i).$$

Again by (7) we have

$$\sum_{i:\ i \in [k^+, k^-]_q} (z_i - y_i) = \sum_{1 \leq i \leq k^-} (z_i - y_i) - \sum_{1 \leq i \leq k^+ - 1} (z_i - y_i) \tag{8}$$

no matter whether $k^+ \leq k^-$ or $k^+ > k^-$. However, from (6) the right hand side of (8) is strictly less than 1. Thus, we have

$$L(\mathbf{z}, a_k^+) - L(\mathbf{y}, a_k^+) = L(\mathbf{z}, a_k^-) - L(\mathbf{y}, a_k^-) < 1.$$

Since this is true for each $k = 0, 1, \ldots, n-1$, it follows that

$$L(\mathbf{z}) < L(\mathbf{y}) + 1 \leq L_{\text{OPT}} + 1.$$

But $L(\mathbf{z}) \geq L_{\text{OPT}}$ and $\mathbf{z}$ is an integral routing, so $L(\mathbf{z}) = L_{\text{OPT}}$ and $\mathbf{z}$ is an optimal routing for the WEIGHTED RING ARC-LOADING PROBLEM WITH INTEGER SPLITTING. Since $\mathbf{y}$ can be found in polynomial time, as mentioned earlier, from the proof above $\mathbf{z}$ can be found in polynomial time as well. □

The proof of Theorem 1 can be easily written as a polynomial time algorithm for producing an integral routing with minimum load. First, we solve LPR and get an optimal solution $\mathbf{x}^*$. Using Lemma 2 and solving $\text{LPR}_{\lfloor \alpha^* \rfloor}$ and $\text{LPR}_{\lceil \alpha^* \rceil}$, we can obtain a flush routing $\mathbf{x}$ with $L(\mathbf{x}) \leq L_{\text{OPT}}$. Applying the procedure detailed in the proof of Lemma 3, we then turn $\mathbf{x}$ into a flush routing $\mathbf{y}$ with $L(\mathbf{y}) \leq L_{\text{OPT}}$ such that no parallel requests are both split by $\mathbf{y}$. Finally, we define $\mathbf{z}$ as in (5), and this is an optimal integral routing for WRALP.

References

[1] L. Becchetti, M.D. Ianni and A. Marchetti-Spaccamela, "Approximation algorithms for routing and call scheduling in all-optical chains and rings," *Theoretical Computer Science*, **287** (2002), 429-448.

[2] T. Carpenter, S. Cosares and I. Saniee, "Demand routing and slotting on ring networks," Technical Report, Bellcore Preprint, 1997.

[3] S. Cosares and I. Saniee, "An optimization problem related balancing loads on SONET rings," *Telecommunications Systems*, **3** (1994), 165-181.

[4] L.G. Khachiyan, "A polynomial algorithm in linear programming," *Soviet Math. Doklady*, **20** (1979), 191-194.

[5] S. Khanna, "A polynomial-time approximation scheme for the SONET ring loading problem," *Bell Laboratories Technical Journal*, **2** (1997), 36-41.

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

[7] Y.S. Myung, "An efficient algorithm for the ring loading problem with integer demand splitting," *SIAM Journal on Discrete Mathematics*, **14** (2001), 291-298.

[8] Y.S. Myung and H.G. Kim, "On the ring loading problem with demand splitting," *Operations Research Letters*, **32** (2004), 167-173.

[9] A. Schrijver, P. Seymour and P. Winkler, "The ring loading problem," *SIAM Journal on Discrete Mathematics*, **11** (1998), 1-14.

[10] A. Schrijver, P. Seymour and P. Winkler, "The ring loading problem," *SIAM Review*, **41** (1999), 777-791.

[11] G. Wilfong and P. Winkler, "Ring routing and wavelength translation," Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms (San Francisco, CA, 1998), pp. 333-341.