Integer Solution to the Ring Loading Problem with Demand Splitting

Young-Soo Myung

Department of Business Administration
Dankook University, Cheonan, Chungnam 330-714, Korea

ABSTRACT *In this paper, we consider a ring loading problem, which arises in the design of SONET bidirectional rings. We deal with the case where demands are allowed to be split and routed in two different directions. Even if integral demands are given, the optimal solution of the problem doesn't always have integral values. We present an efficient algorithm which produces an integral optimal solution.*

1. Introduction

The ring loading problem (RLP) is defined on an undirected ring network $R = (V, L)$ with a node set $V = \{1, 2, \cdots, n\}$ and a link set $L = \{(1,2), (2,3), \cdots, (n-1, n), (n, 1)\}$. Let K be the index set of selected origin-destination pairs of nodes. Then, for each $k \in K$, we are given r_k units of flow requirements (demands). For each $k \in K$, let $o(k)$ and $d(k)$, where $o(k) < d(k)$, denote its origin and destination nodes, respectively. The demand between $o(k)$ and $d(k)$ can be routed in either of two directions, clockwise and counter-clockwise. We say that a flow is routed in the *clockwise* (*counter-clockwise*) direction if a flow passes through the node sequence $\{o(k), o(k)+1, \cdots, d(k)-1, d(k)\}$ ($\{o(k), o(k)-1, \cdots, 1, n, \cdots, d(k)+1, d(k)\}$). Since all the constituent links of a ring have to be with the same capacity, the capacity of a ring is determined by the maximum of the traffic loads imposed on its links. Therefore, the ring capacity determination is dependent on how to route each flow requirement on the ring. The objective of the RLP is then to find an optimal routing which minimizes the maximum link load.

The RLP arises when designing synchronous optical network (SONET) bidirectional self-healing rings (SHRs). For more detalis on SONET SHRs, refer to Wu [7] and Cosares et al. [1]. In a bidirectional SHR, demands may or may not be allowed to be split between both directions. We thus have two kinds of the RLP, with and without demand splitting. For example, in the RLP without demand splitting (RLPWO), each demand must be entirely routed in either of the two directions. Recently, due to its practical significance, the RLP has been considered in many researches. Cosares and Saniee [2] have presented several heuristics for the RLPWO. Myung et al. [5] have developed an $O(n|K|)$ algorithm for the RLPW and an efficient approximation procedure for the RLPWO. Shyur et al. [6] have dealt with the integer version of the RLPW where demands split are restricted to integers. When the demands are integral, the optimal solution of the RLPW, in general, may be the multiples of half-integers. Lee and Chang [3] have developed an approximation algorithm for the integer case of the RLPW. Their algorithm produces an approximate solution whose objective value is at most one unit higher than the optimal value. Lee et al. [4] have derived a class of valid inequalities for the same problem and shown that the problem can be solved in polynomial time using the cuts. However, their algorithm is not shown to be a practical one. In this paper, we develop a very simple algorithm which produces an integral optimal solution for the problem.

2. An algorithm for the integer version of the ring loading problem with demand splitting

For each $k \in K$, let $L_k^+ = \{(i, i+1) \in L | o(k) \leq i < d(k)\}$ and $L_k^- = L \setminus L_k^+$. Then $L_k^+(L_k^-)$ denotes the set of links contained in the clockwise (counter-clockwise) direction path from $o(k)$ to $d(k)$. For each $l \in L$, let $K_l^+ = \{k \in K | l \in L_k^+\}$ and $K_l^- = \{k \in K | l \in L_k^-\}$. Then $K_l^+(K_l^-)$ is the index set of origin-destination pairs whose clockwise (counter-clockwise) direction path contains l. Note that $K_l^- = K \setminus K_l^+$.

For each $k \in K$, let's define variable x_k which denotes the amount of the total demand between $o(k)$ and $d(k)$ routed in the clockwise direction. Let $X = \{\mathrm{x} \in Z^{|K|} | 0 \leq x_k \leq r_k \text{ for each } k \in K\}$, and for a given $\mathbf{x} \in X$, let

$$g(\mathbf{x}, l) = \sum_{k \in K_l^-} x_k + \sum_{k \in K_l^-} (r_k - x_k) \text{ for each } l \in L.$$

Then $g(\mathbf{x}, l)$ denotes the load assigned to link l. Let $F(\mathbf{x}) = \max_{l \in L} g(\mathbf{x}, l)$, then our problem, the integer version of the RLPW can be represented as follows:

$$(P) \qquad z = \min_{\mathbf{x} \in X} F(\mathbf{x}).$$

Our algorithm consists of two main procedures. One is to produce a feasible solution to (P) and the other is to verify the optimality of the solution produced and, if it is not optimal, to construct an optimal solution based on the obtained solution. The first procedure is the same as Lee and Chang's [3]. Initially, all demands are routed in the clockwise direction. Then, for each $k \in K$, it examines whether rerouting all or a part of demand k in the counter-clockwise direction would decrease the ring capacity. Note that for a given $\mathbf{x} \in X$, if $\max_{l \in L_k^+} g(\mathbf{x}, l) \geq \max_{l \in L_k^-} g(\mathbf{x}, l) + 2$, rerouting demand k in the counter-clockwise direction decreases the resulting ring capacity while keeping the integrality condition. Demand k is rerouted until either all the demand is routed in the counter-clockwise direction or the rerouting amount reaches to $\lfloor (1/2) \times \{\max_{l \in L_k^+} g(\mathbf{x}, l) - \max_{l \in L_k^-} g(\mathbf{x}, l)\} \rfloor$ where $\lfloor \alpha \rfloor$ means a largest integer not greater than α.

We assume that the indices of K are ordered as follows: if $o(k_1) < o(k_2)$, then $k_1 < k_2$, and if $o(k_1) = o(k_2)$ and $d(k_1) > d(k_2)$, then $k_1 < k_2$. Such ordering of the origin-destination pairs of $|K|$ can be done within $O(|K| \log |K|)$ time. In the next section, we will show that the rerouting sequence is very important for our algorithm. Since demand k is examined at the kth iteration of the rerouting procedure, we use the same index k for indicating both an origin-destination pair and an iteration step. The first procedure of our algorithm is formally stated as follows:

Algorithm Phase 1;

begin
  [Initial assignment]
  for each $k \in K$ **do**
    $x_k^0 := r_k$;
  [Rerouting]
  for $k = 1, \cdots, |K|$ **do**
  begin
    $\delta := \max\{0, \max_{l \in L_k^+} g(\mathbf{x}^{k-1}, l) - \max_{l \in L_k^-} g(\mathbf{x}^{k-1}, l)\}$;
    $\Delta := \min\{\lfloor \delta/2 \rfloor, r_k\}$;
    $x_k^k := x_k^{k-1} - \Delta$;
    for $i = 1, \cdots, |K|$ and $i \neq k$ **do**
      $x_i^k := x_i^{k-1}$;
  end;
end;

To present the second procedure of our algorithm, we need the following notation. We assume that the links are ordered such that $l = (i, j) < l' = (i', j')$ if $i < i'$. For a given $\mathbf{x} \in X$, let $L(\mathbf{x}) = \{l \in L | g(\mathbf{x}, l) = F(\mathbf{x}) - 1 \text{ or } F(\mathbf{x})\}$. Recall that $\mathbf{x}^k$ denote the solution obtained after the rerouting step of Phase 1 is performed for $k \in K$, $\mathbf{x}^0 = \{r_1, \cdots, r_{|K|}\}$ and $\mathbf{x}^{|K|}$ is the solution which Phase 1 finally produces. Let $l_1 = \min L(\mathbf{x}^{|K|})$ and $l_2 = \max L(\mathbf{x}^{\hat{k}})$ where $\hat{k}$ is defined as follows:

$$\hat{k} = \begin{cases} \max\{k \in K | l_1 \in L_k^-\} & \text{if } \{k \in K | l_1 \in L_k^-\} \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

In the example given in Figure 1, $l_1 = (2, 3)$, $\hat{k} = 2$ and $l_2 = (4, 5)$. We also define a polyhedron $Y \subset R^{|K|}$ such that

$$\begin{aligned} Y \;=\; \{ & y \in R^{|K|} | \\ & y_k = 0,\ k \in K_{l_1}^+ \cap K_{l_2}^+ && (1) \\ & y_k = 0,\ k \in K_{l_1}^- \cap K_{l_2}^- && (2) \\ & 0 \leq y_k \leq r_k,\ k \in K^0 && (3) \\ & -r_k \leq y_k \leq 0,\ k \in K^r && (4) \\ & -x_k^{|K|} \leq y_k \leq r_k - x_k^{|K|},\ k \in K^b && (5) \\ & \sum_{k \in K_l^-} y_k - \sum_{k \in K_l^-} y_k \\ & \leq \left\lfloor \frac{F(\mathbf{x}^{|K|}) - 1 - g(\mathbf{x}^{|K|}, l)}{2} \right\rfloor \times 2.\ l \in L\} && (6) \end{aligned}$$

where $K^0 = \{k \in K : x_k^{|K|} = 0\}$, $K^r = \{k \in K : x_k^{|K|} = r_k\}$, and $K^b = \{k \in K : 0 < x_k^{|K|} < r_k\}$.

Now we describe the second procedure of our algorithm. This procedure starts with $\mathbf{x}^{|K|}$, the solution produced by Phase 1, and constructs an optimal solution of (P), denoted by $\mathbf{x}^*$.

Algorithm Phase 2;

begin
  select l_1 and l_2;
  if $\max\{g(\mathbf{x}^{|K|}, l_1), g(\mathbf{x}^{|K|}, l_2)\} = F(\mathbf{x}^{|K|})$
  then $\mathbf{x}^* = \mathbf{x}^{|K|}$;
  else obtain a vertex solution $y \in Y$;
    if no such solution exists, i.e., $Y = \emptyset$
    then $\mathbf{x}^* = \mathbf{x}^{|K|}$;
    else for $i = 1, \cdots, |K|$ **do**
      $x_i^* := x_i^{|K|} + y_i$;
end;

3. Proof of the validity of the algorithm

Before proving the optimality of our algorithm, we will present some preliminary results for the integer solutions $\mathbf{x}^k$, $k \in K$, which are produced through Phase 1.

Lemma 1 *For each $k = 1, 2, \cdots, |K|$, the following relations hold:*
(i) $L(\mathbf{x}^{k-1}) \subseteq L(\mathbf{x}^k)$;
(ii) $L(\mathbf{x}^{k-1}) \subseteq L_k^+$ if and only if $k \in K^0 \cup K^b$;
(iii) $k \in K^b$ if and only if $L(\mathbf{x}^k) \setminus L_k^+ \neq \emptyset$;
(iv) If $k \in K_{l_1}^+ \cap K_{l_2}^+$, then $k \in K^0$; and
(v) If $k \in K_{l_1}^- \cap K_{l_2}^-$, then $k \in K^r$.

Proof:
(i) follows the fact that once a link is included in $L(\mathbf{x}^k)$ for some k, it remains in $L(\mathbf{x}^{k+1})$. In rerouting step k, $L(\mathbf{x}^{k-1}) \subseteq L_k^+$ if and only if $\Delta > 0$ and thus (ii) holds. A part of demand k is rerouted, only when some link $l \in L_k^-$ newly becomes a member of $L(\mathbf{x}^k)$, that proves (iii). Suppose that $k \in K_{l_1}^+ \cap K_{l_2}^+$, that is, $l_1, l_2 \in L_k^+$. By the definitions of l_1 and l_2, $l_1 \leq \min L(\mathbf{x}^k) \leq \max L(\mathbf{x}^k) \leq l_2$. Since $l_1, l_2 \in L_k^+$, $L(\mathbf{x}^k) \subseteq L_k^+$. By (ii) and (iii), $k \in K^0$ because $L(\mathbf{x}^{k-1}) \subseteq L(\mathbf{x}^k) \subseteq L_k^+$. So (iv) holds.

Finally, we prove (v). Suppose that $k \in K_{l_1}^- \cap K_{l_2}^-$, i.e., $l_1, l_2 \in L_k^-$. Then the following three cases are possible.
Case 1. $\max L_k^+ < l_1$.
Since $l_1 = \min L(\mathbf{x}^{|K|})$, $L(\mathbf{x}^{|K|}) \cap L_k^+ = \emptyset$. From $L(\mathbf{x}^{k-1}) \subseteq L(\mathbf{x}^{|K|})$, $L(\mathbf{x}^{k-1}) \cap L_k^+ = \emptyset$, and thus $k \in K^1$ by (ii).
Case 2. $l_1 < \min L_k^+ \leq \max L_k^+ < l_2$.
We first show that $k - 1 \geq \hat{k}$. When $\hat{k} = 0$, it is obvious. Suppose that $\hat{k} \geq 1$. From $l_1 \in L_{\hat{k}}^+$, $o(k) > o(\hat{k})$, and thus $k > \hat{k}$ by our assumption on the indices of K. Since $l_2 \in L(\mathbf{x}^{\hat{k}}) \subseteq L(\mathbf{x}^{k-1})$, $L(\mathbf{x}^{k-1}) \not\subseteq L_k^+$ which implies $k \in K^1$ by (ii).
Case 3. $l_2 < \min L_k^+$.
In this case, it also holds that $k - 1 \geq \hat{k}$. So $k \in K^1$ as shown in Case 2. □

Lemma 2 *For any $\mathbf{x} \in X$,*

$$g(\mathbf{x}, l_1) + g(\mathbf{x}, l_2) \geq g(\mathbf{x}^{|K|}, l_1) + g(\mathbf{x}^{|K|}, l_2).$$

Proof:
Consider any $\mathbf{x} \in X$ and let $\Delta_k = x_k - x_k^{|K|}$ for each $k \in K$. Then, for each $l \in L$,

$$g(\mathbf{x}, l) - g(\mathbf{x}^{|K|}, l) = \sum_{k \in K_l^+} \Delta_k - \sum_{k \in K_l^-} \Delta_k. \qquad (7)$$

Summing two equations (7) for l_1 and l_2, we have the following results.

$$\begin{aligned} & g(\mathbf{x}, l_1) - g(\mathbf{x}^{|K|}, l_1) + g(\mathbf{x}, l_2) - g(\mathbf{x}^{|K|}, l_2) \quad (8) \\ = & \; 2 \sum_{k \in K_{l_1}^+ \cap K_{l_2}^+} \Delta_k - 2 \sum_{k \in K_{l_1}^- \cap K_{l_2}^-} \Delta_k. \quad (9) \end{aligned}$$

By (iv) and (v) of Lemma 1, $\Delta_k \geq 0$ for each $k \in K_{l_1}^+ \cap K_{l_2}^+$ and $\Delta_k \leq 0$ for each $k \in K_{l_1}^- \cap K_{l_2}^-$. Therefore, the lemma holds. □

Note that $l_1, l_2 \in L(\mathbf{x}^{|K|})$, that is, $g(\mathbf{x}^{|K|}, l) \geq F(\mathbf{x}^{|K|}) - 1$, for $l = l_1, l_2$. So, the following corollary holds.

Corollary 3 *For any $\mathbf{x} \in X$, $F(\mathbf{x}) \geq F(\mathbf{x}^{|K|}) - 1$ and if $\max\{g(\mathbf{x}^{|K|}, l_1), g(\mathbf{x}^{|K|}, l_2)\} = F(\mathbf{x}^{|K|})$, $\mathbf{x}^{|K|}$ is an optimal solution for (P).*

Moreover, Lemma 2 holds, even if X is replaced by $\widehat{X} = \{\mathbf{x} \in R^{|K|} | 0 \leq x_k \leq r_k \text{ for each } k \in K\}$. Therefore, the following relation also holds.

Corollary 4 *Let $z(\overline{P})$ denote the optimal objective value of the linear programming relaxation of (P) where X is replaced by $\widehat{X}$ in (P). Then, $F(\mathbf{x}^{|K|}) \leq z(\overline{P}) + 1$.*

Now we turn to our main proof.

Theorem 5 *The algorithm, which consists of Phase 1 and 2, produces an optimal solution for (P).*

Proof:
By Corollary 3, if $\max\{g(\mathbf{x}^{|K|}, l_1), g(\mathbf{x}^{|K|}, l_2)\} = F(\mathbf{x}^{|K|})$, $\mathbf{x}^{|K|}$ is an optimal solution for (P). To prove the remaining part, we need the following two claims.

Claim 1 *If $Y = \emptyset$, then $\mathbf{x}^{|K|}$ is an optimal solution to (P).*

Proof:
Suppose that $\mathbf{x}^{|K|}$ is not an optimal solution to (P). Then, there must exist $\mathbf{x} \in X$ such that $F(\mathbf{x}) = F(\mathbf{x}^{|K|}) - 1$. Let $y_k = x_k - x_k^{|K|}$, for each $k \in K$. Then (3), (4) and (5) hold by the definitions of K^0, K^r and K^b. By Corollary 3, $g(\mathbf{x}^{|K|}, l_1) = g(\mathbf{x}^{|K|}, l_2) = F(\mathbf{x}^{|K|}) - 1$. From Lemma 2 and the fact that $g(\mathbf{x}, l) \leq F(\mathbf{x}^{|K|}) - 1$ for all $l \in L$, $g(\mathbf{x}, l_1) = g(\mathbf{x}, l_2) = F(\mathbf{x}^{|K|}) - 1$ and

$$\begin{aligned} & g(\mathbf{x}, l_1) - g(\mathbf{x}^{|K|}, l_1) + g(\mathbf{x}, l_2) - g(\mathbf{x}^{|K|}, l_2) \\ = & \; 2 \sum_{k \in K_{l_1}^+ \cap K_{l_2}^+} y_k - 2 \sum_{k \in K_{l_1}^- \cap K_{l_2}^-} y_k \\ = & \; 0. \end{aligned}$$

By (iv) and (v) of Lemma 1, $y_k \geq 0$ for each $k \in K_{l_1}^+ \cap K_{l_2}^+$, and $y_k \leq 0$ for each $k \in K_{l_1}^- \cap K_{l_2}^-$.

Therefore, $y_k = 0$ for all $k \in K_{l_1}^+ \cap K_{l_2}^+$ and $k \in K_{l_1}^- \cap K_{l_2}^-$, that is, (1) and (2) hold.

Now we show that (6) also holds. Note that for each $l \in L$,

$$\sum_{k \in K_l^-} y_k - \sum_{k \in K_l^-} y_k = g(\mathbf{x}, l) - g(\mathbf{x}^{|K|}, l)$$
$$\leq F(\mathbf{x}^{|K|}) - 1 - g(\mathbf{x}^{|K|}, l).$$

So, it is sufficient to show that the left term of (6) always takes even number. The following relation shows this.

$$\begin{aligned} & g(\mathbf{x}, l) - g(\mathbf{x}^{|K|}, l) \\ = & \, g(\mathbf{x}, l) - g(\mathbf{x}^{|K|}, l) + g(\mathbf{x}, l_1) - g(\mathbf{x}^{|K|}, l_1) \\ = & \, 2 \sum_{k \in K_l^- \cap K_{l_1}^-} y_k - 2 \sum_{k \in K_l^- \cap K_{l_1}^-} y_k \end{aligned}$$

The first equality is due to the fact that $g(\mathbf{x}, l_1) = g(\mathbf{x}^{|K|}, l_1) = F(\mathbf{x}^{|K|}) - 1$. Therefore, $y \in Y$, that contradicts the fact that $Y = \emptyset$. □

Claim 2 *If $Y \neq \emptyset$, every vertex $y \in Y$ is integral.*

Proof:
If $Y \neq \emptyset$, Y has a vertex, since Y is a bounded polyhedron. Let $\hat{y}$ be any vertex of Y. Let $K^=$ denote the set of indices corresponding to $\hat{y}_k$'s, each of which satisfies at least one of the constraints (1), (2), (3), (4) and (5) at equality and $L^=$ denote the subset of L corresponding the members of (6) that are satified at equality by $\hat{y}$. Let A denote the constraint matrix given by (6) and $\hat{A}$ be the submatrix of A having its columns in $K \setminus K^=$ and its rows in $L^=$. Let b be the $|L| \times 1$ matrix whose elements, b_l, $l \in L$ are given as follows:

$$b_l = \left\lfloor \frac{F(\mathbf{x}^{|K|}) - 1 - g(\mathbf{x}^{|K|}, l)}{2} \right\rfloor \times 2$$
$$- \sum_{k \in K_l^- \cup K^=} \hat{y}_k + \sum_{k \in K_l^- \cup K^=} \hat{y}_k.$$

Note that all elements of b are either even or odd. Let $\hat{b}$ be the submatrix of b having its elements in $L^=$. Since $\hat{y}$ is a vertex of Y, $\hat{A}$ has full column rank and $\hat{y}$ is the unique solution of the equation system $\hat{A}y = b$. Let $|K \setminus K^=| = m$, A^* be the $m \times m$ submatrix of $\hat{A}$ obtained after deleting the redundant rows and b^* be the submatrix of $\hat{b}$ corresponding to the rows of A^*. Then $A^*y = b^*$ is equivalent to $\hat{A}y = \hat{b}$.

Now we show that $A^*y = b^*$ determines an integral solution. If $m \leq 1$, it trivially holds. Assume that $m \geq 2$ and apply Gaussian elimination to the equation system. A sequence of elementary row operations are performed iteratively so that A^* is transformed to an upper-triangular matrix. Let $(\tilde{A}, \tilde{b})$ denote the matrix which is obtained after performing Gaussian elimination. We will show that each element of the upper-triangular matrix $\tilde{A}$ is equal to 0, 1 or -1 and $\tilde{b}$ is an integer vector. In the first iteration, the first row of (A^*, b^*), is added (or subtracted) to (or from) the other rows so that the first element becomes 0 in those rows. Then, every element in the second and subsequent rows is equal to 0, 2 or -2 and b_j^*, $j > 1$ has even number. Divide every element in the rows other than the first one by 2 and add (or subtract) the second row to (or from) the third and subsequent rows so that the first and second elements set equal to 0 for those rows. We continue this process until we obtain $(\tilde{A}, \tilde{b})$ as defined. Consequently, all variables $\hat{y}_k$ for $k \in K \setminus K^=$ should have integer values. Note that the other variables y_k for $k \in K^=$ are assumed to have integer values. □

Let y be any vertex solution of Y, then $\mathbf{x}^{|K|} + y$ is an integer solution, $\mathbf{x}^{|K|} + y \in X$ (by (3), (4) and (5)) and $F(\mathbf{x}^{|K|} + y) \leq F(\mathbf{x}^{|K|}) - 1$ (by (6)). This completes the proof. □

References

[1] S. Cosares, D.N. Deutsch, I. Saniee and O.J. Wasem, "SONET tookit: a decision support system for designing robust and cost-effective fiber-optic networks", *Interfaces* **25** (1995) 20-40.

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

[3] C.Y. Lee and G. Chang, "Balancing loads on SONET rings with integer demand splitting", Working paper, KAIST (1996).

[4] K. Lee, K. Park, S. Park and S.-B. Kim, "Ring loading problem with integer demand splitting", Working paper, KAIST (1996).

[5] Y.-S. Myung, H.-G. Kim and D.-W. Tcha, "Optimal load balancing on SONET bidirectional rings", to appear in *Operations Research.*

[6] C.C. Shyur, Y.M. Wu and C.H. Chen, "A capacity comparison for SONET self-healing ring networks", *IEEE GLOBECOM* **3** (1993) 1574-1578.

[7] T.H. Wu, *Fiber network service survivability*, Artech House, 1992.