Journal of Combinatorial Optimization, 10, 99–111, 2005
© 2005 Springer Science + Business Media, Inc. Manufactured in The Netherlands.

Maximizing Profits of Routing in WDM Networks

JIANPING LI* jianping@ynu.edu.cn, csjpli@cityu.edu.hk
Department of Mathematics, Yunnan University, Kunming 650091, P.R. China; Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong, P.R. China

KANG LI kangli@sdu.edu.cn
School of Information Science and Engineering, Shandong University, Jinan 250100, P.R. China

LUSHENG WANG lwang@cs.cityu.edu.hk
HAO ZHAO zhaoh@cs.cityu.edu.hk
Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong, P.R. China

Received June 9, 2004; Revised January 31, 2005; Accepted March 1, 2005

Abstract. Let $G = (V, E)$ be a ring (or chain) network representing an optical wavelength division multiplexing (WDM) network with k channels, where each edge e_j has an integer capacity c_j. A *request* $\{s_i, t_i\}$ is a pair of two nodes in G. Given m requests $\{s_i, t_i\}$, $i = 1, 2, \ldots, m$, each with a profit value p_i, we would like to design/route a k-colorable set of paths for some (may not be all) of the m requests such that each edge e_j in G is used at most c_j times and the total profit of the set of designed paths is maximized. Here two paths cannot have the same color (channel) if they share some common edge(s).

This problem arises in optical communication networks. In this paper, we present a polynomial-time algorithm to solve the problem when G is a chain. When G is a ring, however, the optimization problem is *NP*-hard (Wan and Liu, 1998), we present a 2-approximation algorithm based on our solution to the chain network. Similarly, some results in a bidirected chain and a bidirected ring are obtained.

Keywords: minimum-cost flow, routing, path coloring, approximation algorithm

1. Introduction

Optical networks are major network technology in data communications. Wavelength division multiplexing (WDM) is a common technique that divides a single fiber into a number of channels according to different wavelengths (Raghavan and Upfal, 1994). In a WDM network, to set up a connection for a request, a path between the two nodes of the request is selected and the same wavelength is assigned to every edge in the path. In some cases, different wavelengths must be used if two paths share a common edge. This requirement is known as the *wavelength-continuity* constraint. Current optical technologies impose limitations on the number of available wavelengths per fiber (Schrijver et al., 1998; Khanna, 1997; Wilfong and Winkler, 1998). Typically, this number is typically between 20 and 100.

*Corresponding author.

Routing on WDM networks is an important problem in optical fiber communication. A typical topology in network design is a ring. Routing algorithms on rings have been studied extensively. Raghavan and Upfal studied routing techniques for different kinds of optical fiber networks including rings, trees and meshes (Raghavan and Upfal, 1994). The ring loading problem, where the wavelengths on edges in a path connecting the two nodes of a request do not have to be the same, has been studied in Schrijver et al. (1998), Khanna (1997), and Wilfong and Winkler (1998). The objective is to route the network to serve all the requests such that the ring load, i.e., the maximum number of times that an edge can be used, is minimized. Schrijver et al. (1998) provided a fast and simple algorithm for the weighted version of the ring loading problem, which achieves a load that is guaranteed to exceed the optimum by at most 3/2 times the maximum demand. Building on the work of Schrijver et al., Khanna showed that for any fixed $\varepsilon > 0$, there exists a polynomial-time approximation scheme (PTAS) for the weighted version of the same problem in Khanna (1997) which computes a solution that requires bandwidth at most $1 + \varepsilon$ times of the optimum. Wilfong and Winkler studied the directed ring loading problem in Wilfong and Winkler (1998) and then gave a polynomial-time algorithm to solve it. They also tackled the number of wavelengths required if the edges in a path connecting two nodes of a request must have the same wavelength.

When the number of wavelengths allowed in the network is fixed, it might be impossible to route the network to serve all requests. In this case, one may try to route the network to serve as many requests as possible. Wan and Liu studied this problem in Wan and Liu (1998). They proved that this problem is *NP*-hard for general network and for special networks including rings, trees and meshes, and they then gave a $\frac{e}{e-1}$-approximation algorithm to maximize the throughput, i.e., the number of requests that are served, where e is the base of natural logarithm, i.e. $e = 2.718\ldots$. Nomikos et al. (2003) gave a better approximation algorithm with the factor $\frac{3}{2}$. Adamy et al. (2002) studied a problem similar to our problem defined below, where a request is replaced by a fixed path along the clockwise direction and the k-colorable constraint is omitted. They obtained a polynomial-time algorithm.

In this paper, we study the weighted version, where each request has a profit, and the problem is to select a set of requests with maximum total profit. In particular, we first consider the chain and ring networks. We then consider bidirected chains and rings. Up to the recent material, no results are known for the problem concerning the objective to maximize total profits of requests in the literature.

For convenience, we assume that a chain or a ring consists of n nodes labelled clockwise as $1, 2, \ldots, n$. For a chain network, its edge set is $E = \{e_j = (j, j+1) : 1 \le j < n\}$. For a ring network, its edge set is $E = \{e_j = (j, j+1) : 1 \le j < n\} \cup \{e_n = (n, 1)\}$. Similarly, a bidirected chain or a bidirected ring consists of n vertices labelled clockwise as $1, 2, \ldots, n$. For a bidirected chain network, its arc set is $A = E' \cup E''$, where $E' = \{e'_j = (j, j+1) : 1 \le j < n\}$ and $E'' = \{e''_j = (j+1, j) : 1 \le j < n\}$. For a bidirected ring network, its arc set is $A = E' \cup E''$, where $E' = \{e'_j = (j, j+1) : 1 \le j < n\} \cup \{e'_n = (n, 1)\}$ and $E'' = \{e''_j = (j+1, j) : 1 \le j < n\} \cup \{e''_n = (1, n)\}$.

A request $\{s, t\}$ is a pair of two nodes in the network (chain or ring). Let $\mathcal{P} = \{P_1, \ldots, P_q\}$ be a set of q paths (directed paths). The set $\mathcal{P}$ is *k-colorable* if we can assign k colors to the paths (directed paths) in $\mathcal{P}$ such that any two paths (directed paths) sharing any common edge (arc in the same direction) possess different colors. For a subset $\mathcal{Q} \subseteq \mathcal{P}$ and each $j\ (1 \le j \le n)$, the *load* $L(\mathcal{Q}, e_j)$ of edge e_j corresponding to $\mathcal{Q}$ is the number of paths in $\mathcal{Q}$ that contains the edge e_j, i.e., $L(\mathcal{Q}, e_j) = |\{P \in \mathcal{Q} : P$ contains the edge $e_j\}|$. A subset $\mathcal{Q} \subseteq \mathcal{P}$ is called *k-feasible*, i.e., $\mathcal{Q}$ satisfies the *k-feasibility*, if $\mathcal{Q}$ is k-colorable and no load exceeds the capacity of that edge, i.e., $L(\mathcal{Q}, e_j) \le c_j$ for all $1 \le j \le n$. Similarly in the directed network (bidirected chain or bidirected ring), for a subset $\mathcal{Q} \subseteq \mathcal{P}$ and each $j\ (1 \le j \le n)$, the *load* $L(\mathcal{Q}, e'_j)$ (or $L(\mathcal{Q}, e''_j)$) of arc e'_j (or e''_j) corresponding to $\mathcal{Q}$ is the number of directed paths in $\mathcal{Q}$ that contains the arc e'_j (or e''_j), i.e., $L(\mathcal{Q}, e'_j) = |\{P \in \mathcal{Q} : P$ contains the arc $e'_j\}|$ and $L(\mathcal{Q}, e''_j) = |\{P \in \mathcal{Q} : P$ contains the arc $e''_j\}|$. A subset $\mathcal{Q} \subseteq \mathcal{P}$ is called *k-feasible*, i.e., $\mathcal{Q}$ satisfies *k-feasibility*, if $\mathcal{Q}$ is k-colorable and no load exceeds the capacity of that arc, i.e., $L(\mathcal{Q}, e'_j) \le c'_j$ and $L(\mathcal{Q}, e''_j) \le c''_j$ for all $1 \le j \le n$.

The problems are as follows:

Maximizing Profits in Chains (MPC)

INSTANCE: a set C of k colors, a chain $G = (V, E)$, a set of m paths $\mathcal{P} = \{P_1, \dots, P_m\}$, a 'capacity' function $c : E \to \mathcal{Z}^+$ and a 'profit' function $p : \mathcal{P} \to \mathcal{R}^+$;

QUESTION: Find a subset $\mathcal{Q} \subseteq \mathcal{P}$ to satisfy the k-feasibility such that the objective $\sum_{P_i \in \mathcal{Q}} p_i$ is maximum.

Maximizing Profits in Rings (MPR)

INSTANCE: a set C of k colors, a ring $G = (V, E)$, a set of m requests $D = \{\{s_1, t_1\}, \dots, \{s_m, t_m\}\}$, a 'capacity' function $c : E \to \mathcal{Z}^+$ and a 'profit' function $p : D \to \mathcal{R}^+$;

QUESTION: Design a routing $\mathcal{Q}$ consisting of some paths, corresponding to some requests, such that $\mathcal{Q}$ satisfies the k-feasibility and the objective $\sum_{P_i \in \mathcal{Q}} p_i$ is maximum.

The **Maximizing Throughput in Rings** problem (MTR) is a special case of the MPR problem, where each $c_j = k$ and the profit value for each request is 1.

Maximizing Profits in Bidirected Rings (MPBR)

INSTANCE: a set C of k colors, a bidirected ring $G = (V, A)$, a set of m directed requests $D = \{(s_1, t_1), \dots, (s_m, t_m)\}$, a 'profit' function $p : D \to \mathcal{R}^+$ and two positive integers c', c'';

QUESTION: Design a routing $\mathcal{Q}$ consisting of directed some paths, corresponding to some directed requests, such that $\mathcal{Q}$ satisfies the k-feasibility and the objective $\sum_{P_i \in \mathcal{Q}} p_i$ is maximum.

Here, a directed path for the directed request (s_i, t_i) in the MPBR problem is always from s_i to t_i, either clockwise or counterclockwise.

In this paper, we present a polynomial-time algorithm for the MPC problem, a 2-approximation algorithm for the MPR problem and a 2-approximation algorithm for the MPBR problem.

The rest of the paper is organized as follows. In Section 2, we present a polynomial algorithm to solve the MPC problem optimally. A 2-approximation algorithm for the MPR problem is given in Section 3. A 2-approximation algorithm for the MPBR problem is designed in Section 4. We conclude our work in the final section.

2. An exact algorithm for routing in chains

In this section, we study the MPC problem. Since the request $\{s_i, t_i\}$ uniquely determines a path P_i in a chain network, we can use P_i and $\{s_i, t_i\}$ exchangeably in this section.

Our approach is to transform the MPC problem into the *minimum-cost s-t flow* problem. It is an extension of the Carlisle-Lloyd algorithm (Carlisle and Lloyd, 1995).

Definition 1 (The minimum-cost s-t flow problem). Given a digraph $D = (V, A; s, t; c, p)$, $s, t \in V$, a 'capacity' function $c : A \to \mathcal{Q}^+$, a 'cost' function $p : A \to \mathcal{Q}$ and a value k, find an s-t flow f of value k subject to $0 \leq f \leq c$ that minimizes *cost*(f), where *cost*(f) $= \sum_{a \in A} p(a) f(a)$.

There are many polynomial algorithms to solve the problem, see, for instance, Schrijver (2003). Here, we need the Tarjan's result to solve the minimum-cost s-t flow problem, the algorithm in details can be found in Tarjan (1983).

Theorem 1 (Tarjan, 1983). *If a digraph $D = (V, A; s, t; c, p)$ is acyclic and capacities are all integers, then an integer minimum-cost s-t flow of value k can be found in time $\mathcal{O}(|A| + kS(n))$, even if negative costs are permitted, where $S(n)$ is the running time of any algorithm for finding shortest paths in graphs with $\mathcal{O}(n)$ edges.*

2.1. Constructing an acyclic network

For any instance $\mathcal{I}$ of the MPC problem, we may assume $s_i < t_i$ for each $1 \leq i \leq m$ in this section. Denote $\mathcal{P} = \{P_i \mid P_i$ is the path to connect two nodes of the request $\{s_i, t_i\}$, $1 \leq i \leq m\}$ and $x_j = L(\mathcal{P}, e_j)$ for each $1 \leq j < n$. Let $p_{\max} = \max\{p_i : 1 \leq i \leq m\}$.

The new directed network is constructed as $N = (V, A; s, t; c, p)$, where V contains n nodes as $1, 2, \ldots, n$, the source s is the first node 1 in V and the sink t is the last node n, and A, c and p are defined in three cases:

Step 1: For $j = 1, 2, \ldots, n-1$, put a 'clique-arc' $e_j = (j, j+1)$ into A with capacity k and cost *zero*, i.e., $c(e_j) = k$ and $p(e_j) = 0$;

Step 2: For $i = 1, 2, \ldots, m$, construct an 'interval-arc' $e_i^* = (s_i, t_i)$ into A with capacity *one* and cost $-p_i$, i.e., $c(e_i^*) = 1$ and $p(e_i^*) = -p_i$;

Step 3: For $j = 1, 2, \ldots, n-1$, only if $\min\{k, x_j\} > c_j$, construct a 'dummy arc' $e_j^{**} = (j, j+1)$ into A with capacity $\min\{k, x_j\} - c_j$ and cost $-p_{\max}m - 1$, i.e., $c(e_j^{**}) = \min\{k, x_j\} - c_j$ and $p(e_j^{**}) = -p_{\max}m - 1$.

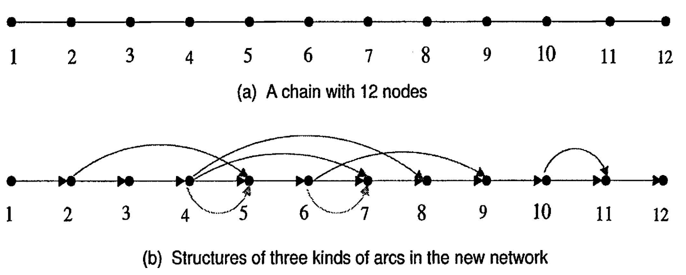
1
2
3
4
5
6
7
8
9
10
11
12
(a) A chain with 12 nodes
(b) Structures of three kinds of arcs in the new network

Figure 1. The construction of the new network.

Figure 1 gives an example to construct an acyclic network. In figure 1(a), the set of requests is $\{\{2, 5\}, \{4, 7\}, \{4, 8\}, \{6, 9\}, \{10, 11\}\}$, $k \geq 3$ and $c(e) = 2$ for any edge e in the chain. In figure 2, five arcs (2, 5), (4, 7), (4, 8), (6, 9) and (10, 11) are 'interval-arc's, two arcs (4, 5) and (6, 7) are 'dummy arc's, and the others are 'clique-arc's.

$N = (V, A; s, t; c, p)$ contains n nodes and $n + m + \sum_{j=1}^{n-1} \max\{0, |\{j : \min\{k, x_j\} > c_j\}|\} \leq 2n + m$ arcs, where n is the number of nodes and m is the number of requests on G. Obviously, the time required to construct $N = (V, A; s, t; c, p)$ is $\mathcal{O}(2n + m)$.

Lemma 2. *Let OPT be the profit of an optimal solution to an instance $\mathcal{I}$ of the* MPC *problem and $N = (V, A; s, t; c, p)$ the directed network constructed from $\mathcal{I}$. The cost value c(f) of the minimum-cost s-t integer flow f of value k in $N = (V, A; s, t; c, p)$ is*

$$c(f) = -OPT - (p_{\max} m + 1) \sum_{j-1}^{n-1} \max\{\min\{k, x_j\} - c_j, 0\}.$$

Moreover, given an optimal solution for the MPC *problem, we can construct a minimum-cost s-t integer flow of value k in $N = (V, A; s, t; c, p)$ in polynomial time and vice versa.*

Proof: Suppose that we have an optimal solution with profit *OPT* to an instance $\mathcal{I}$ of the MPC problem, i.e., $\mathcal{P}_{OPT} = \{P_{s_{i_1}, t_{i_1}}, \ldots, P_{s_{i_r}, t_{i_r}}\}$ is an optimal solution to an instance $\mathcal{I}$ of the MPC problem such that $OPT = \sum_{(s_i, t_i) \in P_{OPT}} p_i$. Denote $P_{OPT} = \{(s_{i_1}, t_{i_1}), \ldots, (s_{i_r}, t_{i_r})\}$. Let $\mathcal{P} = \{P_1, \ldots, P_m\}$ be the set of paths, corresponding to the set of m requests. For each $1 \leq j < n$, denote $x_j = L(\mathcal{P}, e_j)$ and $z_j = L(\mathcal{P}_{OPT}, e_j)$.

Now, we can construct a function $f : A \to \mathcal{R}^+$ as follows:

(i) For an 'interval-arc' $e_i^* = (s_i, t_i) (i \in \{1, 2, \ldots, m\})$, if $e_i^* \in P_{OPT}$, define $f(e_i^*) = 1$, and if $e_i^* \notin P_{OPT}$, define $f(e_i^*) = 0$;
(ii) For a 'dummy-arc' $e_j^{**} = (j, j + 1) (j \in \{1, 2, \ldots, n - 1\})$, by construction, $\min\{k, x_j\} > e_j$, define $f(e_j^{**}) = \min\{k, x_j\} - c_j$;
(iii) For a 'clique-arc' $e_j = (j, j + 1) (j \in \{1, 2, \ldots, n - 1\})$, if $\min\{k, x_j\} > c_j$, define $f(e_j) = k - (\min\{k, x_j\} - c_j) - z_j$, and if $\min\{k, x_j\} \leq c_j$, define $f(e_j) = k - z_j$.

It is easy to check that f is an s-t integer flow of value k and its cost value is

$$\begin{aligned}
cost(f) &= \sum_{e\in A} p(e)f(e)\\
&= \sum_{i=1}^{m} p((s_i,t_i))f((s_i,t_i)) + \sum_{e_j^{**}:\min\{k,x_j\}>c_j} p(e_j^{**})f(e_j^{**}) + \sum_{j=1}^{n-1} p(e_j)f(e_j)\\
&= \sum_{e_i^*\in P_{OPT}} p(e_i^*)f(e_i^*) + \sum_{e_j^{**}:\min\{k,x_j\}>c_j} p(e_j^{**})f(e_j^{**}) + \sum_{j=1}^{n-1} 0\cdot f(e_j)\\
&= \sum_{e_i^*\in P_{OPT}} (-p_i) + \sum_{e_j^{**}:\min\{k,x_j\}>c_j} (-p_{\max}m-1)(\min\{k,x_j\}-c_j)\\
&= -\sum_{e_i^*\in P_{OPT}} p_i + (-p_{\max}m-1)\sum_{e_j^{**}:\min\{k,x_j\}>c_j} (\min\{k,x_j\}-c_j)\\
&= -OPT - (p_{\max}m+1)\sum_{j=1}^{n-1} \max\{\min\{k,x_j\}-c_j,0\}
\end{aligned}$$

where the first and second equalities depend on the definition of *cost*(f), the third equality comes from the facts $f(e_i^*)=0$ for $e_i^*\notin P_{OPT}$ and $p(e_j)=0$ for $e_j=(j,j+1)$, the fourth equality is from the facts $p(e_i^*)=-p_i$, $f(e_i^*)=1$ for $e_i^*\in P_{OPT}$ and $p(e_j^{**})=-p_{\max}m-1$, $f(e_j^{**})=\min\{k,x_j\}-c_j$ for e_j^{**}. Since the second item in the last equality is a constant, the optimal profit *OPT* to an instance $\mathcal{I}$ of the MPC problem implies that the cost value *cost*(f) of flow f of value k is smallest. So f is a minimum-cost s-t integer flow of value k in N.

Conversely, if we have a minimum-cost s-t integer flow f on N, since the cost per unit of each 'dummy-arc' $e_j^{**}=(j,j+1)$ in the network N is $-p_{\max}m-1$, then each 'dummy-arc' e_j^{**} must be chosen in the minimum-cost s-t integer flow f to possess its flow value $\max\{\min\{k,x_j\}-c_j,0\}$ through that 'dummy-arc' by algorithm MPC, and the set of 'interval-arc's are partitioned into two subsets: one consists of 'interval-arc's of value *zero* and the other consists of 'interval-arc's of value *one*, which imply that the latter 'interval-arc's in the flow f consist of the arc set $P_{OUT}=\{(s_{i_1},t_{i_1}),\ldots,(s_{i_r},t_{i_r})\}$ (and corresponding to the set of paths). So the cost value of minimum-cost s-t integer flow f of value k is as follows

$$cost(f) = -\sum_{l=1}^{r} p_{i_l} - (p_{\max}m+1)\sum_{j=1}^{n-1}\max\{\min\{k,x_j\}-c_j,0\}.$$

For each arc (s_{i_l},t_{i_l}) in P_{OUT}, we can construct a path $P_{s_{i_l},t_{i_l}}$ to connect s_{i_l} and t_{i_l}, and denote $\mathcal{P}_{OUT}=\{P_{s_{i_1},t_{i_1}},\ldots,P_{s_{i_r},t_{i_r}}\}$. The paths in $\mathcal{P}_{OUT}$, whose corresponding 'interval-arc's belong to the identical path sharing a unit of flow from the source s to the sink t, are assigned the same color, then the paths in $\mathcal{P}_{OUT}$ can be checked to be k-colorable in time $\mathcal{O}(m)$ (Carlisle and Lloyd, 1995) and the profit of $\mathcal{P}_{OUT}$ is $OUT=\sum_{l=1}^{r} p_{i_l}$. Hence we can obtain

$$
\begin{aligned}
OUT &= \sum_{l=1}^{r} p_{i_l} \\
&= -cost(f) - (p_{\max} m + 1) \sum_{j=1}^{n-1} \max\{\min\{k, x_j\} - c_j, 0\}
\end{aligned}
$$

Since the second item in the last equality is a constant, the cost value $cost(f)$ of minimum-cost s-t integer flow f of value k implies that the set of paths $\mathcal{P}_{OUT}$ is an optimal solution to the instance $\mathcal{I}$ of the MPC problem. □

2.2. *The whole algorithm*

Using the algorithm for minimum-cost s-t flow problem, we can solve the MPC problem in polynomial time. The complete algorithm is described below:

Algorithm: Maximizing Profits in Chains (MPC)

INPUT: a set C of k colors, a chain $G = (V, E)$, a set of m paths $\mathcal{P} = \{P_1, \ldots, P_m\}$, a 'capacity' function $c : E \to \mathcal{Z}^+$ and a 'profit' function $p : \mathcal{P} \to \mathcal{R}^+$;
OUTPUT: Find a subset $\mathcal{Q} \subseteq \mathcal{P}$ to satisfy the k-feasibility such that the objective $\sum_{P_i \in \mathcal{Q}} p_i$ is maximum.
Step 1: Construct the directed network $N = (V, A; s, t; c, p)$;
Step 2: Compute the minimum-cost s-t flow in $N = (V, A; s, t; c, p)$;
Step 3: Construct the set $\mathcal{Q}$ of 'interval-arc's of value *one*, i.e., $\mathcal{Q} = \{e_i^* = (s_i, t_i) : f(e_i^*) = 1, 1 \le i \le m\}$;
Step 4: For all 'interval-arc's in $\mathcal{Q}$, assign the same color to the arcs that belong to the identical path sharing a unit of flow from the source s to the sink t. (There are at most k colors used since the flow f has its value k)

End of MPC

Theorem 3. *Algorithm* MPC *solves the* MPC *problem in* $\mathcal{O}(k(2n + m))$ *time, where n is the number of nodes, m is the number of requests and k is the number of colors.*

Proof: Lemma 2 implies the correctness of algorithm MPC. We now analyse the complexity of algorithm MPC.

In step 1, it needs $\mathcal{O}(2n + m)$ time. The acyclic network N consists of n nodes and at most $2n + m$ arcs.

In step 2, we can construct a minimum-cost s-t integer flow f of value k on the acyclic network N in $\mathcal{O}(k(2n + m))$ time.

In step 3, the construction of the set of 'interval-arc's with flow value *one* needs $\mathcal{O}(m)$ time.

In step 4, the set of $\mathcal{Q}$ is checked to be k-colorable in $\mathcal{O}(m)$ time.

Thus, algorithm MPC runs in $\mathcal{O}(n+m)+\mathcal{O}(k(2n+m))+\mathcal{O}(m)+\mathcal{O}(m) = \mathcal{O}(k(2n+m))$ time. □

3. Approximation algorithm for MPR

In this section, we consider the MPR problem. The main difficulty here is that there are two choices, either clockwise or counterclockwise, to serve a request. Since the MPR problem is a generalization of the MTR problem, the *NP*-hardness of the MTR problem (Wan and Liu, 1998) implies that the MPR problem is also *NP*-hard. We give a 2-approximation algorithm here. The basic idea is to convert the ring problem to the chain problem by removing an appropriate edge from the ring, and then use the algorithm for chains to solve the problem.

We first consider the fact: if n is *odd*, we can insert a new node $n+1$ on the edge $(n, 1)$ to obtain two new edges $e_n = (n, n+1)$ and $e_{n+1} = (n+1, 1)$ having the same capacity as the original edge $(n, 1)$. Then the two rings have the same solutions to the MPR problem. Without loss of generality, we now assume that n is *even* so as to give a simple proof. For convenience, let OPT be an optimal solution to the MPR problem and OPT_{e_j} the set of paths in OPT that pass through the edge e_j for each j $(1 \le j \le n)$. Set $OPT^c_{e_j} = OPT - OPT_{e_j}$. In addition, we treat OPT, OPT_{e_j} and $OPT^c_{e_j}$ as both the set and its profit, respectively.

To obtain the 2-approximation algorithm, we shall use algorithm MPC as a subroutine, and the following lemma will play an important role in the design of our algorithm.

Lemma 4. *Let OPT be the profit of an optimal routing to an instance $\mathcal{I}$ of the MPR problem. Then there exists an optimal routing $\mathcal{P}_{OPT} = \{P_{s_{i_1},t_{i_1}}, \ldots, P_{s_{i_r},t_{i_r}}\}$ such that $OPT_{e_j} \le \frac{OPT}{2}$ for some j $(1 \le j \le n)$.*

Proof: We prove the lemma by contradiction. Suppose $OPT_{e_j} > \frac{OPT}{2}$ for all $1 \le j \le n$. Consider an optimal routing $\mathcal{P}_{OPT} = \{P_{s_{i_1},t_{i_1}}, \ldots, P_{s_{i_r},t_{i_r}}\}$ to an instance $\mathcal{I}$ of the MPR problem that satisfies the two following properties:

(i) the length of any longest path in $\mathcal{P}_{OPT}$ is the shortest among all optimal routings and
(ii) under condition (i), $\mathcal{P}_{OPT}$ has the minimum number of the longest paths.

If the length of the longest paths in the routing $\mathcal{P}_{OPT}$ is at most $\frac{n}{2}$, then we have $OPT_{e_l} + OPT_{e_{\frac{n}{2}+1}} \le OPT$. This implies $\min\{OPT_{e_1}, OPT_{e_{\frac{n}{2}+1}}\} \le \frac{OPT}{2}$, contradicting our assumption.

We now consider the case where the length of the longest path in $\mathcal{P}_{OPT}$ is more than $\frac{n}{2}$. Without loss of generality, we assume that the longest path $P_{s_{i_r},t_{i_r}}$ connecting the request $\{s_{i_r}, t_{i_r}\}$ along the clockwise direction passes through both edges $e_{s_{i_r}} = (s_{i_r}, s_{i_r}+1)$ and $e_{t_{i_r}-1} = (t_{i_r}-1, t_{i_r})$, and that t_{i_r} is before s_{i_r} on the ring along clockwise direction.

It follows from our assumption that $\min\{OPT_{e_{s_{i_r}}}, OPT_{e_{t_{i_r}}}\} > \frac{OPT}{2}$ and that $P_{s_{i_r},t_{i_r}}$ is a longest path, we conclude that there must exist a path $P_{s_{i_j},t_{i_j}}$ in $\mathcal{P}_{OPT}$ that connects the request $\{s_{i_j}, t_{i_j}\}$ and passes through both edges $e_{s_{i_r}}$ and $e_{t_{i_r}}$. (Otherwise, if $P_{s_{i_j},t_{i_j}}$ does not exist, then any path in $\mathcal{P}_{OPT}$ containing edge $(s_{i_r}, s_{i_r}+1)$ does not contain edge $(t_{i_r}, t_{i_r}+1)$ and vice versa. This shows that the intersection of two sets $OPT_{e_{s_{i_r}}}$ and $OPT_{e_{t_{i_r}}}$ is empty. Thus, we have $OPT \ge OPT_{e_{s_{i_r}}} + OPT_{e_{t_{i_r}}} > \frac{OPT}{2} + \frac{OPT}{2} = OPT$. But, this is impossible).

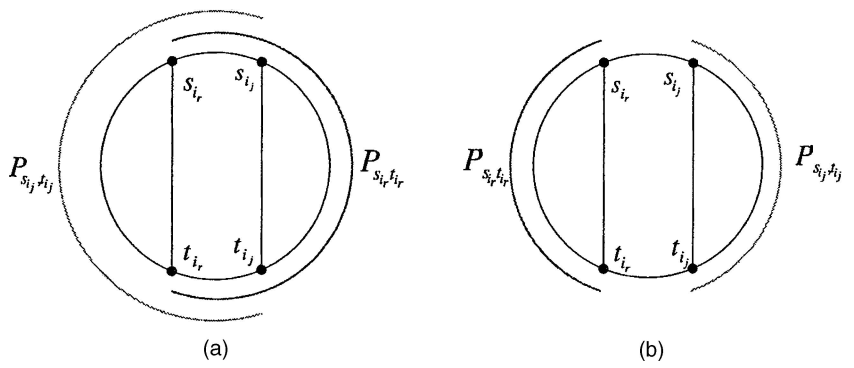

Figure 2. Transformation from $P_{s_{i_r},t_{i_r}}$, $P_{s_{i_j},t_{i_j}}$ to $P'_{s_{i_r},t_{i_r}}$, $P'_{s_{i_j},t_{i_j}}$.

The two nodes s_{i_j}, t_{i_j} must be located in the path $P_{s_{i_r},t_{i_r}}$. By the assumption that $P_{s_{i_r},t_{i_r}}$ is the longest path, $P_{s_{i_j},t_{i_j}}$ must connect the request $\{s_{i_j}, t_{i_j}\}$ along the counterclockwise direction. See figure 2(a). The two paths $P_{s_{i_r},t_{i_r}}$ and $P_{s_{i_j},t_{i_j}}$ cover the whole ring G.

Now, we can construct two new paths: $P'_{s_{i_r},t_{i_r}}$ connects the request $\{s_{i_r}, t_{i_r}\}$ along the counterclockwise direction and $P'_{s_{i_j},t_{i_j}}$ connects the request $\{s_{i_j}, t_{i_j}\}$ along the clockwise direction. See figure 2(b). It is easy to see that both lengths of these paths $P'_{s_{i_r},t_{i_r}}$ and $P'_{s_{i_j},t_{i_j}}$ are at most the length of $P_{s_{i_r},t_{i_r}}$ *minus* one by our assumption that the length of longest path $P_{s_{i_r},t_{i_r}}$ is more than $\frac{n}{2}$. So, we can obtain a new optimal routing consisting of the paths $P'_{s_{i_r},t_{i_r}}$, $P'_{s_{i_j},t_{i_j}}$ and others from $\mathcal{P}_{OPT} - \{P_{s_{i_r},t_{i_r}}, P_{s_{i_j},t_{i_j}}\}$. The new routing has either a smaller number of longest paths or the length of the longest path in the new routing is reduced. This contradicts the choice (i) or (ii) of the routing $\mathcal{P}_{OPT}$.

Thus, we complete the proof of this lemma. □

Using algorithm MPC as a subroutine, we can design a 2-approximation algorithm to the MPR problem as follows:

Algorithm: Maximizing Profits in Rings (MPR)

```
INPUT: an instance $\mathcal{I}$ of the MPR problem;
OUTPUT: a $k$-feasible solution $\mathcal{Q}$ such that the objective $\sum_{P_i \in \mathcal{Q}} p_i$ is maximized.
Step 1: For each $j = 1$ to $n$ do
    Step 1.1 Remove the edge $e_j = (j, j+1)$, and then obtain a chain $L_j$ with the source
        $s = j + 1$ and the sink $t = j$;
    Step 1.2 Apply algorithm MPC to $L_j$, then we get an optimal routing consisting of
        paths $\mathcal{Q}_j = \{P_{j_1}, \ldots, P_{j_q}\}$ corresponding to the request set $D_j = \{\{s_{j_1}, t_{j_1}\}, \ldots,$
        $\{s_{j_q}, t_{j_q}\}\}$ in $L_j$;
    endfor
Step 2 Choose a routing $\mathcal{Q}_{j^*}$ among these $n$ routings $\mathcal{Q}_j$ to maximize profit, i.e., $\sum_{P_i \in \mathcal{Q}_{j^*}}$
    $p_i = \max_{1 \le j \le n}\{\sum_{P_i \in \mathcal{Q}_j} p_i\}$;
```

Step 3 Export the routing $\mathcal{Q}_{j^*}$, corresponding to the set $\mathcal{D}_{j^*}$, and the value $OUT = \sum_{P_i \in \mathcal{Q}_{j^*}} p_i$.
End of MPR

It is easy to see that algorithm MPR runs in $\mathcal{O}(kn(2n + m))$ time. So we can present our main result concerning the MPR problem.

Theorem 5. *Algorithm* MPR *is a* 2*-approximation algorithm for the* MPR *problem and it runs in* $\mathcal{O}(kn(2n+m))$ *time, where n is the number of nodes of the ring G, m is the number of requests on G and k is the number of colors.*

Proof: Let OUT be the feasible solution by algorithm MPR. Using Lemma 4, we can choose an optimal solution OPT with its value OPT to satisfy $OPT_{e_j} \leq \frac{OPT}{2}$ for some $j(1 \leq j \leq n)$, here we may simply treat OUT (OPT, respectively) as both the feasible solution (optimal solution, respectively) and its profit in this proof.

Now, removing the edge e_j and deleting all paths from OPT_{e_j}, we obtain the chain L_j and the feasible routing $OPT^c_{e_j}$ with its profit $OPT^c_{e_j}$ to this chain L_j, and by the choice of e_j, we get $OPT^c_{e_j} \geq \frac{OPT}{2}$. Utilizing algorithm MPC in algorithm MPR to maximize profit on L_j with the source $j + 1$ and the sink j, we can get $OUT \geq OPT^c_{e_j}$. So we reach $OUT \geq \frac{OPT}{2}$, i.e., $\frac{OPT}{OUT} \leq 2$.

This completes the proof of Theorem 5. □

4. Approximation algorithm for MPBR

In this section, we give a 2-approximation algorithm for the MPBR problem, where all capacities of arcs in the identical direction are the same, i.e., $c'_j = c'$ and $c''_j = c''$ hold for all $1 \leq j \leq n$. Similar to the MPR problem, we can choose a directed path from s_i to t_i in one of the two choices, either clockwise or counterclockwise, to serve the directed request (s_i, t_i). The basic ideas are: (i) delete the two arcs e'_n and e''_n from the bidirected ring and use the algorithm for chains to solve the problem, (ii) use a greedy algorithm to compute a set of directed paths containing either e'_n or e''_n, and (iii) select the better solution obtained in (i) and (ii).

Our approximation algorithm to the MPBR problem is as follows:

Algorithm: Maximizing Profits in Bidirected Rings (MPBR)

INPUT: an instance $\mathcal{I}$ of the MPBR problem;
OUTPUT: a k-feasible solution $\mathcal{Q}$ such that the objective $\sum_{P_i \in \mathcal{Q}} p_i$ is maximized.
Step 1 Remove the two arcs $e'_n = (n, 1)$ and $e''_n = (1, n)$, and then obtain the bidiected chain L;
Step 2 Set $D_1 = \{(s_i, t_i) \in D : s_i < t_i \text{ and } 1 \leq i \leq m\}$ and $D_2 = \{(s_i, t_i) \in D : s_i > t_i \text{ and } 1 \leq i \leq m\}$;

```
Step 3 Use algorithm MPC with the set of directed requests $D_1$, the arc set $E' = \{e'_j = (j, j+1) : 1 \le j < n\}$, the source $s = 1$, the sink $t = n$ and arc capacity $c'$. Let $\mathcal{Q}_1$ be the set of optimal directed paths obtained;
Step 4 Use algorithm MPC with the set of directed requests $D_2$, the arc set $E'' = \{e''_j = (j+1, j) : 1 \le j < n\}$, the source $s = n$, the sink $t = 1$ and arc capacity $c''$. Let $\mathcal{Q}_2$ be the set of optimal directed paths obtained;
Step 5 Set $\mathcal{Q} = \mathcal{Q}_1 \cup \mathcal{Q}_2$, where $\mathcal{Q}_1 = \{P_{j_1}, \ldots, P_{j_{q_1}}\}$ and $\mathcal{Q}_2 = \{P_{j_{q_1+1}}, \ldots, P_{j_{q_1+q_2}}\}$;
Step 6 Choose $\min\{k, c''\}$ requests in $D_2$ possessing the heaviest profits and use the corresponding directed paths, $\mathcal{P}_1$, containing the arc $e'_n = (n, 1)$ to route;
Step 7 Choose $\min\{k, c''\}$ requests in $D_1$ possessing the heaviest profits and use the corresponding directed paths, $\mathcal{P}_2$, containing the arc $e''_n = (1, n)$ to route;
Step 8 Set $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$, where $\mathcal{P}_1 = \{P_{i_1}, \ldots, P_{i_{\min\{k,c'\}}}\}$ and $\mathcal{P}_2 = \{P_{i_{\min\{k,c'\}+1}}, \ldots, P_{i_{\min\{k,c'\}+\min\{k,c''\}}}\}$;
Step 9 Output the better solution of $\mathcal{P}$ and $\mathcal{Q}$.
End of MPBR
```

Theorem 6. *Algorithm* MPBR *is a* 2*-approximation algorithm for the* MPBR *problem.*

Proof: Let OPT be an optimal set to an instance $\mathcal{I}$ of the MPBR problem and $OUT = \max\{\sum_{P_i \in \mathcal{P}} p_i, \sum_{P_j \in \mathcal{Q}} p_j\}$ the output profit value obtained from algorithm MPBR. OPT^1_c and OPT^1_{cc} represent the sets of clockwise and counterclockwise directed paths in OPT, respectively, where no path passes through the arc e'_n or the arc e''_n. Similarly, OPT^2_c and OPT^2_{cc} represent the sets of clockwise and counterclockwise directed paths in OPT, respectively, where each path passes through either the arc e'_n or the arc e''_n. Again, for convenience, OPT, OPT^1_c, OPT^1_{cc}, OPT^2_c and OPT^2_{cc} are also used as the total profits of all paths in OPT, OPT^1_c, OPT^1_{cc}, OPT^2_c and OPT^2_{cc}, respectively. So OPT is partitioned into four disjoint subsets, i.e., $OPT = OPT^1_c \cup OPT^1_{cc} \cup OPT^2_c \cup OPT^2_{cc}$.

From Theorem 3, we can obtain $OPT^1_c \le \sum_{l=1}^{q_1} p_{j_l} = \sum_{P_{j_l} \in \mathcal{Q}_1} p_{j_l}$ and $OPT^1_{cc} \le \sum_{l=q_1+1}^{q_1+q_2} p_{j_l} = \sum_{P_{j_l} \in \mathcal{Q}_2} p_{j_l}$. So we get

$$OPT^1_c + OPT^1_{cc} \le \sum_{P_{j_l} \in \mathcal{Q}_1} p_{j_l} + \sum_{P_{j_l} \in \mathcal{Q}_2} p_{j_l} = \sum_{P_{j_l} \in \mathcal{Q}} p_{j_l} \le OUT \tag{1}$$

Now, we want to show that the greedy technique in steps 6 and 7 ensures $OUT^2_c \le \sum_{P_{i_l} \in \mathcal{P}_1} p_{i_l}$ and $OPT^2_{cc} \le \sum_{P_{i_l} \in \mathcal{P}_2} p_{i_l}$. Let $OPT^2_c = \{P_{i'_1}, \ldots, P_{i'_{p1}}\}$. Without loss of generality, we may assume that $p_{i_1} \ge p_{i_2} \ge \ldots \ge p_{i_{\min\{k,c'\}}}$ and $p_{i'_1} \ge p_{i'_2} \ge \ldots \ge p_{i'_{p1}}$. Since OPT^2_c is a feasible solution to an instance $\mathcal{I}$ of the MPBR problem, we get $\min\{k, c'\} \ge p_1$. By the facts that (1) the request of each path in OPT^2_c must be in D_2, (2) every arc has the same capacity and (3) we choose the $\min\{k, c'\}$ requests in D_2 possessing the heaviest profits by the greedy algorithm in step 6, we can get $p_{i'_l} \le p_{i_l}$ for each $1 \le l \le p_1 (\le \min\{k, c'\})$. Thus, $\sum_{l=1}^{p_1} p_{i'_l} \le \sum_{l=1}^{p_1} p_{i_l} \le \sum_{l=1}^{\min\{k,c'\}} p_{i_l}$, i.e., $OPT^2_c \le \sum_{P_{i_l} \in \mathcal{P}_1} p_{i_l}$. Similarly, we can obtain $OPT_{cc}^2 \leq \sum_{P_{i_l} \in \mathcal{P}_2} p_{i_l}$. So we get

$$OPT_c^2 + OPT_{cc}^2 \leq \sum_{P_{i_l} \in \mathcal{P}_1} p_{j_l} + \sum_{P_{j_l} \in \mathcal{P}_2} p_{j_l} = \sum_{P_{j_l} \in \mathcal{P}} p_{j_l} \leq OUT \tag{2}$$

Combing the two inequalities (1) and (2), we obtain

$$\begin{aligned} OPT &= OPT_c^1 + OPT_{cc}^1 + OPT_c^2 + OPT_{cc}^2 \\ &\leq OUT + OUT \\ &= 2OUT \end{aligned}$$

This completes the proof of Theorem 6. □

5. Conclusion

In this paper, we study the problem of designing/routing some (may not be all) requests in WDM ring (and chain) networks, the objective is to maximize total profit of the different paths used, where each edge e_j in the networks is used at most c_j times and two paths cannot possess the same wavelength (or color) if they share some common edge(s). Using the minimum-cost flow technique, we design a polynomial-time algorithm for the MPC problem optimally, then we derive a 2-approximation algorithm for the MPR problem and also a 2-approximation algorithm for the MPBR problem.

Acknowledgments

The authors are grateful to three anonymous reviewers whose comments and suggestions have led to a substantially improved presentation for the paper.

Jianping Li is fully supported by the National Natural Science Foundation of China [Project No. 10271103], the Project-sponsored by SRF for ROCS, SEM, China, and Natural Science Foundation of Yunnan Province [Project No. 2003F0015M].

Lusheng Wang and Hao Zhao are fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. Cityu 1196/03E].

References

U. Adamy, C. Ambuehl, R.S. Anand, and T. Erlebach, "Call control in rings," in *Proceedings of the 29th International Colloquium on Automata, Languages, and Programming (ICALP 2002)*, LNCS 2380, Springer-Verlag, 2002, pp. 788–799.

M.C. Carlisle and E.L. Lloyd, "On the k-coloring of intervals," *Discrete Applied Mathematics*, vol. 59, no 3, pp. 225–235, 1995.

J. Edmonds and R.M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of Association for Computing Machinery*, vol. 19, pp. 248–264, 1972.

M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco, 1979.

S. Khanna, "A polynomial time approximation scheme for the SONET ring loading problem," *Bell Labs Tech. J.*, vol. 2, pp. 36–41, 1997.

C. Nomikos, A. Pagourtzis, and S. Zachos, "Minimizing request blocking in all-optical rings," in *Proceedings of IEEE INFOCOM*, San Francisco, CA, USA, 2003, pp. 1355–1361.

P. Raghavan and E. Upfal, "Efficient routing in all-optical networks," in *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*, 1994, pp. 134–143.

A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Springer: The Netherlands, 2003.

A. Schrijver, P. Seymour, and P. Winkler, "The ring loading problem," *Siam Journal on Discrete Mathematics*, vol. 11, no. 1, pp. 1–14, 1998.

R. Tarjan, *Data Structures and Network Algorithms*, SIAM, Philadelphia, PA, 1983.

P. Wan and L. Liu, "Maximal throughput in wavelength-routed optimal networks," in P. Wan, D. Du and P. Pardalos (eds.), *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* vol. 46: *Multichannel Optimal Networks—Theory and Practice*, 1998, pp. 15–26.

G. Wilfong and P. Winkler, "Ring routing and wavelength translation," in *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, San Francisco, California, 1998, pp. 333–341.