Wavelength management in WDM rings to maximize the number of connections*

Ioannis Caragiannis

Research Academic Computer Technology Institute &
Department of Computer Engineering and Informatics
University of Patras, 26500 Rio, Greece

Abstract. We study computationally hard combinatorial problems arising from the important engineering question of how to maximize the number of connections that can be simultaneously served in a WDM optical network. In such networks, WDM technology can satisfy a set of connections by computing a route and assigning a wavelength to each connection so that no two connections routed through the same fiber are assigned the same wavelength. Each fiber supports a limited number of w wavelengths and in order to fully exploit the parallelism provided by the technology, one should select a set connections of maximum cardinality which can be satisfied using the available wavelengths. This is known as the *maximum routing and path coloring* problem (maxRPC).
Our main contribution is a general analysis method for a class of iterative algorithms for a more general coloring problem. A lower bound on the benefit of such an algorithm in terms of the optimal benefit and the number of available wavelengths is given by a *benefit-revealing linear program*. We apply this method to maxRPC in both undirected and bidirected rings to obtain bounds on the approximability of several algorithms. Our results also apply to the problem maxPC where paths instead of connections are given as part of the input. We also study the profit version of maxPC in rings where each path has a profit and the objective is to satisfy a set of paths of maximum total profit.

1 Introduction

Combinatorial problems arising from high speed communication networks utilizing the Wavelength Division Multiplexing (WDM) technology have received significant attention since the mid 90's. Such networks connect nodes through optical fibers. Each fiber can simultaneously carry different data streams provided that each stream is carried on a different wavelength. In order to fully exploit the capabilities of these networks, the same wavelength has to be used along a path so that all necessary processing is performed on the optical domain and slow opto-electronic conversions are avoided [20]. Given connection requests (i.e., transmitter-receiver pairs), the WDM technology establishes communication by

* This work was partially supported by the European Union under IST FET Integrated Project 015964 AEOLUS.

finding a path from each transmitter to the corresponding receiver and assigning a wavelength to each path so that paths crossing the same fiber are using different wavelengths. The number of available wavelengths (i.e., the available optical bandwidth) is limited, so they have to be used efficiently.

The underlying WDM network can be modeled as an undirected graph assuming that each fiber can handle transmissions in both directions. WDM networks can also be modeled by bidirected graphs (i.e., directed graphs which contain a directed link (u, v) if and only if it contains (v, u)). Here, we assume that each fiber is dedicated to transmitting in one direction.

We use the term *connection* to denote a pair of a transmitter and a receiver that wish to communicate. Given a set of connections, a *routing* is a set of paths connecting each transmitter to the corresponding receiver. The *load* of a set of paths is the maximum number of paths crossing any link of the network. The load of a set of paths gives a lower bound on the number of wavelengths that are necessary to satisfy them. Combinatorial problems of interest are those which aim either to minimize the number of wavelengths (intuitively, we may think of the wavelengths as *colors*) for a set of connections or to maximize the number of connections that can be satisfied (also called *benefit*) given a limitation w on the number of available wavelengths. Formally, we define the following problems that abstract the most interesting engineering questions in WDM networks.

Routing and path coloring (RPC). Given a set of connections R on a network, find a routing P of R and a coloring of P with the minimum number of colors. When paths instead of connections are given on input, we have the *path coloring* (PC) problem.

Maximum routing and path coloring (maxRPC). Given a positive integer w and a set of connections R on a network, find a subset of R of maximum cardinality which has a routing that can be colored with at most w colors. When paths instead of connections are given on input, we have the *maximum path coloring* (maxPC) problem.

For general networks, the above problems have been proved to be hard to approximate. In particular, problems PC and maxPC are in general equivalent to minimum graph coloring and maximum independent set, two problems which are very unlikely to have efficient approximation algorithms. Although these results are disappointing from the practical point of view, the topologies deployed by the telecommunication industry are much simpler; trees, rings, and mesh-like planar networks are the most popular ones. In such networks, much better approximations are feasible. For example, in trees where unique paths correspond to connections, the above problems have algorithms which approximate the optimal solution within a constant factor (e.g., see [2, 6–8]).

In this paper we focus on ring networks. Problem PC in rings is also known as *circular arc coloring* and has received significant attention in the literature (e.g., [3, 10, 13, 14, 22, 23]). It has been proved to be NP-hard in [10]. The best general approximation algorithm has approximation ratio 3/2 [13] while better approximation algorithms exist in the case where the load of the set of paths is polylogarithmic on the ring size [14, 15] or/and the minimum number of paths required to cover the whole ring is not very small [3, 23]. Observe that the ring is the simplest topology where routing decisions can be part of the problems. Problem RPC is also known to be NP-hard [7]. The best known approximation algorithm has approximation ratio slightly smaller than 2 [5], while an algorithm with approximation ratio approaching $\frac{3}{2} + \frac{1}{2e} \approx 1.68394$ when the optimal number of colors is substantially large compared to the size of the ring has been presented in [15].

For general values of w, problems maxPC and maxRPC are NP-hard; their NP-hardness follow by the NP-hardness of problems PC and RPC, respectively. When $w = 1$, maxRPC is actually the problem of computing a maximum number of connections that can be routed through link-disjoint paths. Awerbuch et al. [1] (see also [24]) show that algorithms that iteratively call link-disjoint paths algorithms to compute solutions to maxRPC with arbitrary w have slightly worse approximation ratio than the ratio of the algorithm that is used to compute link-disjoint paths. In undirected and bidirected rings, link-disjoint paths of maximum size can be computed in polynomial-time yielding iterative algorithms with approximation ratio $\frac{e}{e-1} \approx 1.58198$ [24] (see also the discussion in [18]). The best known approximation algorithms for maxRPC (and maxPC) in undirected rings has approximation ratio 3/2 [17, 18] while an 11/7-approximation algorithm for maxRPC in bidirected rings is presented in [18]. Interesting variants of the maxRPC and maxPC problems are their profit versions where connections (or paths) are associated with non-negative profits and the objective is to color with at most w colors a set of connections (or paths) with the maximum total profit. A simple iterative algorithm has approximation ratio 1.58198 in this case (the proof follows by extending the analysis of [1, 24]). Li et al. [16] present a 2-approximation algorithm for a related problem which also has arbitrary load constraints on the links of the ring.

The aim of this paper is to improve the known approximability bounds for maxRPC and maxPC in rings. Before presenting our results, we give a brief overview of the ideas in [17, 18]. The algorithm of [18] for the undirected case of maxRPC (similar ideas are used in [17] for maxPC) actually applies two different algorithms on the input instance and outputs the best among the two solutions. The first algorithm colors some of the connections on input with at most w colors by using the same color in at most two connections. This is done by a maximum matching computation on the graph which represents the compatibility of pairs of connections. Of course, this may lead to inefficient solutions if w is very small compared to the size of the optimal solution. In order to handle this case, another simple maxRPC algorithm is used whose performance increases as w decreases. This algorithm simply ignores one link of the ring and routes all connections so that no path uses this link. In this way an instance of maxPC on a chain network is obtained. maxPC in chains can be solved optimally in polynomial time by an algorithm of Carlisle and Lloyd [4]. By simple arguments, this second algorithm will color at most w connections less than those colored in an optimal solution of the original maxRPC instance. The same idea can be used in bidirected rings to color at most $2w$ connections less than those colored in an optimal solution.

In our algorithms, we also use this last algorithm to handle instances in which the optimal solution is much larger than w. We will refer to this algorithm as algorithm CL. In order to handle the most difficult case of large w, we will exploit iterative algorithms. Their main advantage compared to the maximum matching algorithm of [18] is that they can color more than two connections with the same color if this is feasible. We consider not only the *basic iterative algorithm* that iteratively computes link-disjoint paths but also more involved algorithms. We show that even the basic iterative algorithm combined with algorithm CL has approximation ratio $18/13 \approx 1.38462$ and $60/41 \approx 1.46341$ in undirected and bidirected rings, respectively, significantly improving the $\frac{e}{e-1}$ bound of [1, 24] and the ratios of the algorithms in [17, 18]. More involved iterative algorithms that use local search algorithms for computing *set packings* are proved to achieve approximation ratios $4/3$ and $719/509 + \epsilon \approx 1.41257$, respectively. We also study the profit version of maxPC and we present an algorithm based on linear programming and randomized rounding [19] with approximation ratio $1+\frac{4}{3e} \approx 1.49015$, improving on the 1.58198 bound obtained by a simple iterative algorithm. Again, we use as a subroutine an algorithm of Carlisle and Lloyd [4] for solving the profit variant of maxPC in chains.

For the analysis of the algorithms for the non-profit version of the problems, we develop a new technique which is quite general and could be applied to many other contexts where we are given a set of elements together with subsets of elements that can be assigned the same color and the objective is to color the maximum number of elements using no more than w colors. In particular, we present the *benefit-revealing LP lemma* which provides lower bounds on the performance of iterative algorithms for such problems in terms of the size of the optimal solution, w, and the objective value of a linear program. This technique is motivated by studies of greedy-like algorithms for facility location problems [12] but, in contrast to [12], benefit-revealing LPs do not directly yield any bound on the approximation factor; this requires some additional case analysis.

The rest of the paper is structured as follows. In Section 2 we present the $\mathsf{maxColoring}$ problem which generalizes problems maxRPC and maxPC, define a class of iterative $\mathsf{maxColoring}$ algorithms, and present the benefit-revealing LP lemma for analyzing their performance. In Section 3, we present our maxRPC algorithms. The profit version of maxPC is studied in Section 4. Due to lack of space, most of the proofs have been omitted from this extended abstract.

2 Iterative algorithms for the maxColoring problem

The problems maxRPC and maxPC can be thought of as special cases of the $\mathsf{maxColoring}$ problem defined as follows. We are given an integer w, a set V and a set $\mathcal{S}$ of subsets of V called *compatible sets* ($\mathcal{S}$ is closed under subsets). The objective is to compute a subset of w disjoint sets of $\mathcal{S}$ whose union contains a maximum number of elements of V. In other words, we are seeking for an assignment of colors to as many elements of V as possible so that at most w different colors are used and, for each color, the set of elements colored with this color is a compatible set. The compatible sets can be given either explicitly or implicitly. For example, for $w = 1$ and by defining the compatible sets to be the independent sets of a graph, the problem is identical to the maximum independent set problem. In maxRPC instances, the compatible sets are all those sets of connections which have a routing so that the corresponding paths do not share the same fiber (i.e., sets of link-disjoint paths).

The maxColoring problem is strongly related to the problem of computing a compatible set of maximum size. A simple iterative algorithm repeatedly (i.e., w times) includes a compatible set of maximum size that does not contain elements that are contained in compatible sets selected before. Awerbuch et al. [1] (see also [24]) have shown that, using an algorithm that computes a compatible set of size at most ρ times smaller than the size of the maximum compatible set, the corresponding iterative algorithm has approximation ratio at most $\frac{1}{1-\exp(-1/\rho)}$. Even in the case where a compatible set of maximum size can be computed in polynomial time (this is trivial if all compatible sets are given explicitly), the approximation ratio of the iterative algorithm is $\frac{e}{e-1} \approx 1.58198$. In general, this bound is best possible. A maxColoring algorithm with strictly better approximation ratio could be executed repeatedly to approximate minimum set cover within a factor of $\alpha \ln n$ for some constant $\alpha < 1$, contradicting a famous inapproximability result due to Feige [9].

In this paper, we are interested in solutions of instances of the maxColoring problem when a compatible set of maximum size can be computed in polynomial time. We study the class of iterative maxColoring algorithms which try to accommodate elements of V by computing as many as possible disjoint compatible sets of the maximum size. This involves solving instances of the *k-set packing problem*. An instance of k-set packing consists of a set of elements V, a set $\mathcal{S}$ of subsets of V each containing exactly k elements, and the objective is to compute a maximum number of disjoint elements of $\mathcal{S}$. A solution to this problem is called a k-set packing. A k-set packing is called *maximal* if it cannot be augmented by including another set of $\mathcal{S}$ without loosing feasibility. An iterative maxColoring algorithm works as follows:

INPUT: An integer w, a set V and a set of compatible sets $\mathcal{S} \subseteq 2^V$.
OUTPUT: At most w disjoint sets T_1, T_2, ..., of $\mathcal{S}$.

1. Set $F := V$, $\mathcal{T} := \mathcal{S}$, $i := 1$ and denote by k the size of the largest compatible set in $\mathcal{T}$.
2. While $i \le w$ or $F \ne \emptyset$ do:
   (a) Compute a maximal k-set packing Π among the sets of $\mathcal{T}$ of cardinality k.
   (b) If $\Pi \ne \emptyset$ then
      i. Denote by I_0, I_1, ..., I_{t-1} the compatible sets in Π and set $F' := \cup_{j=0}^{\min\{w-i,t-1\}} I_j$.
      ii. For $j := 0, ..., \min\{w-i, t-1\}$, set $T_{i+j} := I_j$.
      iii. Set $F := F \setminus F'$, $\mathcal{T} := \mathcal{T} \setminus \cup_{S \in \mathcal{T}: F' \cap S \ne \emptyset} S$ and $i := i + t$.
   (c) Set $k := k - 1$.

The algorithm that iteratively computes a compatible set of maximum size (henceforth called the *basic iterative algorithm*) can be thought of as an algorithm belonging to the above class of algorithms. In step 2a, it computes a maximal k-set packing by iteratively computing compatible sets of size k and removing from F the elements in the compatible sets computed. Since including a compatible set of size k may force at most k compatible sets of an optimal k-set packing to be excluded from the solution, this algorithm has approximation ratio k for solving the k-set packing problem. Using different methods for computing k-set packings, we obtain different algorithms. When the maximum compatible set has constant size κ, computing a maximal k-set packing can be done using a local search algorithm. Consider the set S of all compatible sets of V of size κ. A local search algorithm uses a constant parameter p (informally, this is an upper bound on the number of local improvements performed at each step) and, starting with an empty packing Π, repeatedly updates Π by replacing any set of $s < p$ sets of Π with $s + 1$ sets so that feasibility is maintained and until no replacement is possible. This algorithm is analyzed in [11].

Theorem 1 (Hurkens and Schrijver [11]). *The local search algorithm that computes k-set packings by performing at most p local improvements at each step has approximation ratio at most $\frac{k(k-1)^r-k}{2(k-1)^r-k}$ if $p = 2r - 1$ and $\frac{k(k-1)^r-2}{2(k-1)^r-2}$ if $p = 2r$.*

The next lemma relates the approximation ratio of iterative algorithms with the approximation ratio of the k-set packing algorithms used in step 2a.

Lemma 1 (Benefit-revealing LP). *Let* Alg *be an iterative* maxColoring *algorithm that uses ρ_k-approximation algorithms for computing maximal k-set packings in step 2a. Consider the execution of* Alg *on a* maxColoring *instance $(V, \mathcal{S}, w)$ and let $\mathcal{OPT} \subseteq V$ be an optimal solution for this instance. If* Alg *terminates by including elements of compatible sets of size $k = t$, then, for any $\lambda > t$, its benefit is at least*

$$\left(1 - \frac{t}{\lambda + 1}\right) |\mathcal{OPT}| + \left(t - \lambda + \frac{\lambda t}{\lambda + 1} + Z^*_{\lambda,t}\right) w$$

where $Z^*_{\lambda,t}$ is the maximum objective value of the following linear program

$$\begin{aligned}
\text{maximize} \quad & \left(1 - \frac{t}{\lambda + 1}\right) \sum_{i=t+1}^{\lambda-1} (\lambda - i)\gamma_i + \sum_{i=t+1}^{\lambda} \delta_i + \sum_{i=t+1}^{\lambda} \beta_i \\
\text{subject to:} \quad & \left(1 - \frac{t}{\lambda + 1}\right) \sum_{i=t+1}^{j-1} \gamma_i + \delta_j \le 1 - \frac{t}{j}, j = t + 2, ..., \lambda \\
& \left(1 - \frac{t}{\lambda + 1}\right) \sum_{i=t+1}^{\lambda-1} \gamma_i + \delta_\lambda + \beta_\lambda \le 1 - \frac{t}{\lambda + 1} \\
& \left(1 - \frac{t}{\lambda + 1}\right) \gamma_j + \delta_{j+1} - \delta_j - \beta_j \ge 0, j = t + 1, ..., \lambda - 1
\end{aligned}$$

$$\frac{j}{\rho_j}\left(1-\frac{t}{\lambda+1}\right)\sum_{i=t+1}^{j-1}\gamma_i+\frac{j}{\rho_j}\delta_j+\beta_j\leq\frac{j-t}{\rho_j}, j=t+1,...,\lambda$$
$$\gamma_j\geq 0, j=t+1,...,\lambda-1$$
$$\delta_j,\beta_j\geq 0, j=t+1,...,\lambda$$

Lemma 1 can be extremely helpful for the analysis of the performance of iterative algorithms on instances of maxColoring where a maximum compatible set can be computed in polynomial time and, additionally, the ratio $|\mathcal{OPT}|/w$ is upper-bounded by a (small) constant. For these instances, the $\frac{e}{e-1}$ bound for the basic iterative algorithm following by the analysis in $[1, 24]$ can be improved. The new proofs are not particularly complicated and they require solving a few simple linear programs.

3 Applications to maxRPC

In the case of maxRPC, the ratio of the size of the optimal solution over the number of available wavelengths is not bounded in general. Hopefully, very simple algorithms are efficient when this ratio is large while iterative algorithms are proved to be efficient for small values of this ratio through the benefit-revealing LP analysis. So, all the maxRPC algorithms we describe in the section have the same structure. They execute algorithm CL and an iterative algorithm on the input instance and output the best among the two solutions.

We denote by CL-I the algorithm obtained by combining algorithm CL with the basic iterative algorithm that iteratively computes compatible sets of connections on undirected rings. The approximation ratio of algorithm CL-I is stated in the next theorem.

Theorem 2. *Algorithm* CL-I *has approximation ratio at most* $18/13$ *for* maxRPC *in undirected rings.*

Note that this is already an improvement to the 3/2-approximation algorithm of [18]. Next we further improve the bound of Theorem 2 by using another simple iterative algorithm. For $k\geq 4$, algorithm I&3LS computes maximal k-set packings in the naive way (i.e., by mimicking the basic iterative algorithm). Maximum 2-set packings among compatible sets of 2 connections are computed using maximum matching computation while a 2-approximation algorithm is used to compute maximal 3-set packings among compatible sets of connections of size 3 (i.e., a local search algorithm performing 2 local improvements at each step). Algorithm CL-I&3LS simply calls both algorithms CL and I&3LS on the input instance and outputs the best among the two solutions.

Theorem 3. *Algorithm* CL-I&3LS *has approximation ratio at most* $4/3$ *for* maxRPC *in undirected rings.*

Proof. Consider the application of algorithm CL-I&3LS on an instance of problem maxRPC consisting of a set of connections on an undirected ring supporting w

wavelengths. Denote by $\mathcal{OPT}$ an optimal solution. If $w \leq |\mathcal{OPT}|/4$, algorithm CL computes a solution of size at least $\frac{3}{4}|\mathcal{OPT}|$. We will also show that when $w \geq |\mathcal{OPT}|/4$, algorithm I&3LS computes a solution of size at least $\frac{3}{4}|\mathcal{OPT}|$.

We may assume that algorithm I&3LS has used all the w wavelengths when it terminates (if this is not the case, then algorithm I&3LS has optimal benefit). We distinguish between three cases depending whether the last wavelength is assigned to compatible sets of connections of size at least 3, 2, or 1. If all wavelengths are assigned to connections in compatible sets of size at least 3, then the benefit of algorithm I&3LS is at least $3w \geq \frac{3}{4}|\mathcal{OPT}|$.

For $\lambda = 3$ and $t = 2$, the benefit-revealing LP is simply to maximize $\delta_3 + \beta_3$ subject to $\frac{3}{2}\delta_3 + \beta_3 \leq \frac{1}{2}$ with $\delta_3, \beta_3 \geq 0$. This is trivially maximized to 1/2 which yields that the benefit of algorithm I&3LS when it terminates by assigning the last wavelength to a compatible set of 2 connections is at least $\frac{1}{2}|\mathcal{OPT}| + w \geq \frac{3}{4}|\mathcal{OPT}|$.

For $\lambda = 3$ and $t = 1$, the benefit-revealing LP is

$$\begin{aligned} \text{maximize} \quad & \frac{3}{4}\gamma_2 + \delta_2 + \delta_3 + \beta_2 + \beta_3 \\ \text{subject to} \quad & \frac{3}{4}\gamma_2 + \delta_3 - \delta_2 - \beta_2 \geq 0 \\ & \frac{3}{4}\gamma_2 + \delta_3 + \beta_3 \leq \frac{3}{4} \\ & \frac{3}{4}\gamma_2 + \delta_3 \leq \frac{2}{3} \\ & \delta_2 + \beta_2 \leq \frac{1}{2} \\ & \frac{9}{8}\gamma_2 + \frac{3}{2}\delta_3 + \beta_3 \leq 1 \\ & \gamma_2, \delta_2, \delta_3, \beta_2, \beta_3 \geq 0 \end{aligned}$$

which is maximized to 5/4 for $\gamma_2 = 2/3$, $\delta_2 = \delta_3 = 0$, $\beta_2 = 1/2$, and $\beta_3 = 1/4$. Hence, we obtain that the benefit of algorithm I&3LS when it terminates by assigning the last wavelength to a single connection is at least $\frac{3}{4}|\mathcal{OPT}|$. □

Next we present algorithms that improve the 11/7 approximation bound of [18] in bidirected rings. We denote by bCL-I the algorithm obtained by combining algorithm CL with the basic iterative algorithm that iteratively computes compatible sets of connections on bidirected rings. Its approximation ratio is stated in the next theorem.

Theorem 4. *Algorithm* bCL-I *has approximation ratio at most* 60/41 *for* maxRPC *in bidirected rings.*

We can exploit local search algorithms for computing set packings among compatible sets of connections. Algorithm I&7LS uses the naive iterative algorithm to compute k-set packings for $k \geq 8$, while it uses the $k/2+\epsilon$-approximation local search algorithms to compute k-set packings for $k \in \{4, 5, 6, 7\}$. Optimal

3-set packings among compatible sets of connections of size 3 in bidirected rings are easy to compute using a maximum matching computation while 2-set packing is trivial since any set of 2 connections in a bidirected ring is a compatible set. Algorithm bCL-I&7LS simply calls both algorithms CL and I&7LS on the input instance and again outputs the best among the two solutions.

Theorem 5. *Algorithm* bCL-I&7LS *has approximation ratio at most* $719/509+\epsilon$ *for* maxRPC *in bidirected rings.*

Note that we have made no particular attempt to design k-set packing algorithms among compatible sets of connections in rings with better approximation guarantees than those of the general set packing algorithms analyzed in [11]. Although, in general, improving the bounds in [11] is a long-standing open problem this may be easier by exploiting the particular structure of the ring. An algorithm for 4-set packing among compatible sets of 4 connections with approximation ratio strictly smaller than 2 would immediately yield an iterative algorithm with approximation ratio strictly better than 4/3 for maxRPC in undirected rings. Similar improvements could be possible in bidirected rings as well.

4 Approximating the profit version of maxPC

By adapting the algorithms presented in Section 3 to work with paths instead of connections, we can obtain the same approximation bounds with those in Theorems 2 and 3 for the maxPC problem as well. Both results improve the 3/2-approximation algorithm of [17].

In the following we consider the profit version of the maxPC where together with each path we are given a non-negative profit and the objective is to select a w-colorable set of paths (or, equivalently, w disjoint compatible sets of paths) of maximum total profit. Again, we use two algorithms and pick the best solution. The first algorithm essentially mimics an algorithm of Carlisle and Lloyd [4] for the profit version of maxPC in chains applied to the paths not traversing a particular link e_0 of the ring. The second algorithm solves a linear programming relaxation of the problem maxPC and obtains a feasible integral solution by applying randomized rounding.

Given a set of paths P on a ring, denote by $\mathcal{I}$ the set of all compatible sets of paths in P. The problem can be expressed as the following integer linear program.

$$
\begin{aligned}
\text{maximize} \; & \sum_{p\in P} c_p \sum_{I\in\mathcal{I}:p\in I} y_I \\
\text{subject to} \; & \sum_{I\in\mathcal{I}:p\in I} y_I \leq 1, \forall p \in P \\
& \sum_{I\in\mathcal{I}} y_I \leq w \\
& y_I \in \{0,1\}, \forall I \in \mathcal{I}
\end{aligned}
$$

Although the above ILP has an exponential number of variables, we can solve its linear programming relaxation (obtained by relaxing the integrality constrained to $0 \le y_I \le 1$) by transforming it to a multicommodity flow problem. Denote by P_0 the subset of P containing the paths traversing link e_0. Consider the following network $N = (V(N), E(N))$ having two special nodes s and t, two nodes s_p and t_p for each path $p \in P_0$ and one node v_p for each path $p \in P \setminus P_0$. The nodes s and t have capacity w while all other nodes in $V(N)$ have unit capacity. For each pair of compatible paths p, q such that $p \in P_0$ and $q \in P \setminus P_0$, $E(N)$ contains the directed edges (s_p, v_q) and (v_q, t_p). For any two compatible paths $p, q \in P \setminus P_0$ such that path p is met prior to q when we walk clockwise on the ring starting from edge e_0, $E(N)$ contains the directed edge (v_p, v_q). For any path $p \in P \setminus P_0$, $E(N)$ contains the two directed edges (s, v_p) and (v_p, t). A directed path from s to t in N corresponds to a compatible set of paths in $P \setminus P_0$, while a directed path from node s_p to node t_p corresponds to a compatible set of paths containing the path $p \in P_0$. Denote by U_0 (resp. V_0) the set of nodes s (resp. t) and s_p (resp. t_p) for each $p \in P_0$.

Now, the maxPC problem with profits is equivalent to computing flows for each commodity (the flow for commodity corresponding to a node $u \in U_0$ has to be carried to the corresponding node in V_0) such that the capacity constraints are not violated (i.e., the flow entering/leaving any node in $V(N) \setminus \{s, t\}$ is at most 1 and the flow entering node t or leaving node s is at most w), the total flow of all commodities is at most w, and the quantity $\sum_{p \in P} c_p \sum_{u \in U_0} f_{v_p}^{(u)}$ is maximized. By $f_{v_p}^{(u)}$ we denote the flow for commodity corresponding to node $u \in U_0$ that is carried by the node v_p. In order to compute the values of the fractional variables in the solution of the LP relaxation, it suffices to decompose the flow of each commodity into flow paths and to set y_I equal to the flow carried by the flow path corresponding to compatible set I. The variables of compatible sets that correspond to flow paths carrying no flow are implicitly set to zero.

Denote by y^* the optimal solution to the LP relaxation of ILP. By ignoring the paths in P_0, we get an instance of the multicommodity flow problem with just one commodity. It can be easily seen that the constraint matrix of the corresponding LP is totally unimodular and, since the capacities are integral, this LP has an integral optimal extreme solution that can be computed in polynomial time [21]. In this way we obtain an integral feasible solution $\bar{y}$ for ILP which implicitly assigns zeros to all compatible sets that contain a path in P_0. Since this solution is optimal on the input instance consisting of the paths in $P \setminus P_0$, we obtain that the cost of $\bar{y}$ is

$$\sum_{p \in P \setminus P_0} c_p \sum_{I \in \mathcal{I}: p \in I} \bar{y}_I \ge \sum_{p \in P \setminus P_0} c_p \sum_{I \in \mathcal{I}: p \in I} y_I^* = \sum_{p \in P} c_p \sum_{I \in \mathcal{I}: p \in I} y_I^* - \sum_{p \in P_0} c_p \sum_{I \in \mathcal{I}: p \in I} y_I^*$$

In the following, we show how to obtain a good feasible solution for ILP by applying randomized rounding to its linear programming relaxation. The randomized rounding procedure works as follows. First, introduce dummy paths with zero profit that contain only e_0 into each compatible set I that does not contain any path in P_0 and whose variable has non-zero value in the fractional solution. Order the compatible sets $I \in \mathcal{I}$ whose fractional variable y_I^* is non-zero such that the compatible sets containing the same path of P_0 are consecutive. Let $I_1, I_2, ..., I_m$ be such an ordering. Let $W = \lceil \sum_{I \in \mathcal{I}} y_I^* \rceil$. Clearly, $W \leq w$. Pick W independent random variables $X_1, X_2, ..., X_W$ in range $(0, 1]$. Define

$$\mathcal{J} = \left\{ I_{j(t)} : \sum_{i=1}^{j(t)-1} y_{I_i}^* - t < X_t \leq \sum_{i=1}^{j(t)} y_{I_i}^* - t, \text{ for } t = 1, ..., W \right\}$$

So far, we have selected one compatible set for each of the w available colors. Since some of the paths may be contained in more than one compatible sets of $\mathcal{J}$, we apply the following procedure to guarantee that each path is contained in at most one compatible set. Consider each path that is contained in more than one compatible set of $\mathcal{J}$. Remove path p from all such compatible sets of $\mathcal{J}$ but one. Denote by $\mathcal{J}'$ the set of compatible sets obtained by the compatible sets of $\mathcal{J}$ in this way. Set $\hat{y}_I = 1$ for each $I \in \mathcal{J}'$ and $\hat{y}_I = 0$ for each $I \in \mathcal{I} \backslash \mathcal{J}'$.

Lemma 2. *The solution $\hat{y}$ obtained by applying the randomized rounding procedure on the optimal fractional solution y^* has expected cost*

$$E\left(\sum_{p \in P} c_p \sum_{I \in \mathcal{I}: p \in I} \hat{y}_I\right) \geq \left(1 - \frac{1}{e}\right) \sum_{p \in P} c_p \sum_{I \in \mathcal{I}: p \in I} y_I^* + \left(\frac{1}{e} - \frac{1}{4}\right) \sum_{p \in P_0} c_p \sum_{I \in \mathcal{I}: p \in I} y_I^*$$

Hence, we obtain that by selecting the best among the two solutions $\bar{y}$ and $\hat{y}$, we obtain a solution with expected total profit at least $\frac{3e}{3e+4}$ times the optimal profit. For any $\epsilon > 0$, we may repeat the method above $O(\frac{\log n}{\epsilon})$ times in order to obtain a solution with profit at least $\frac{3e}{3e+4+3e\epsilon}$ times the optimal profit, with probability at least $1 - 1/n$. The proof follows by a simple application of the Markov inequality. We obtain the following theorem.

Theorem 6. *For any $\epsilon > 0$, the algorithm described computes a $\left(1 + \frac{4}{3e} + \epsilon\right)$-approximate solution for the profit version of* maxPC *in rings in time polynomial on the input size and $1/\epsilon$.*

References

1. B. Awerbuch, Y. Azar, A. Fiat, S. Leonardi, and A. Rosen. Online competitive algorithms for call admission in optical networks. *Algorithmica*, 31(1), pp. 29-43, 2001.
2. I. Caragiannis, A. Ferreira, C. Kaklamanis, S. Perennes, and H. Rivano. Fractional path coloring with applications to WDM networks. In *Proceedings of the 29th International Colloquium on Automata, Languages, and Programming (ICALP '01)*, LNCS 2076, Springer, pp. 732-743, 2001.
3. I. Caragiannis and C. Kaklamanis. Approximate path coloring with applications to wavelength routing in WDM optical networks. In *Proceedings of the 21st Symposium on Theoretical Aspects of Computer Science (STACS '04)*, LNCS 2996, Springer, pp. 258-269, 2004.

4. M.C. Carlisle and E.L. Lloyd. On the k-coloring of intervals. *Discrete Applied Mathematics*, 59, pp. 225-235, 1995.
5. C. T. Cheng. Improved approximation algorithms for the demand routing and slotting problem with unit demands on rings. *SIAM Journal on Discrete Mathematics*, 17(3), pp. 384-402, 2004.
6. T. Erlebach and K. Jansen. The maximum edge-disjoint paths problem in bidirected trees. *SIAM Journal on Discrete Mathematics*, 14(3), pp. 326-366, 2001.
7. T. Erlebach and K. Jansen. The complexity of path coloring and call scheduling. *Theoretical Computer Science*, 255(1-2), pp. 33-50, 2001.
8. T. Erlebach, K. Jansen, C. Kaklamanis, M. Mihail, and P. Persiano. Optimal wavelength routing on directed fiber trees. *Theoretical Computer Science*, 221(1-2), pp. 119-137, 1999.
9. U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4), pp. 634-652, 1998.
10. M.R. Garey, D.S. Johnson, G.L. Miller, and C.H. Papadimitriou. The complexity of coloring circular arcs and chords. *SIAM Journal on Alg. Disc. Math.*, 1(2), pp. 216-227, 1980.
11. C. A. J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. *SIAM Journal on Discrete Mathematics*, 2(1), pp. 68-72, 1989.
12. K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *Journal of the ACM*, 50(6), pp. 795-824, 2003.
13. I. Karapetian. On coloring of arc graphs. *Dokladi of the Academy of Sciences of the Armenian SSR*, 70(5), pp. 306-311, 1980. (in Russian)
14. V. Kumar. An approximation algorithm for circular arc coloring. *Algorithmica*, 30(3), pp. 406-417, 2001.
15. V. Kumar. Approximating circular arc colouring and bandwidth allocation in all–optical ring networks. In *Proceedings of the 1st International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX '98)*, LNCS 1444, Springer, pp. 147–158, 1998.
16. J. Li, K. Li, L. Wang, and H. Zhao. Maximizing profits of routing in WDM networks. *Journal of Combinatorial Optimization*, 10, pp. 99-111, 2005.
17. C. Nomikos, A. Pagourtzis, S. Zachos. Satisfying a maximum number of pre-routed requests in all-optical rings. *Computer Networks*, 42, pp. 55-63, 2003.
18. C. Nomikos, A. Pagourtzis, S. Zachos. Minimizing request blocking in all-optical rings. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, 2003.
19. P. Raghavan and C.D. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7, pp. 365–374, 1987.
20. R. Ramaswami and K. Sivarajan. Optical networks: A practical perspective. *Morgan Kauffman Publishers*, 1998.
21. A. Schrijver. Theory of Linear and Integer Programming. *Wiley and Sons*, 1998.
22. A. Tucker. Coloring a family of circular arcs. *SIAM Journal of Applied Mathematics*, 29(3), pp. 493–502, 1975.
23. M. Valencia-Pabon. Revisiting Tucker's algorithm to color circular arc graphs. *SIAM Journal on Computing*, 32(4), pp. 1067-1072, 2003.
24. P.J. Wan and L. Liu. Maximal throughput in wavelength-routed optical networks. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, AMS, 46, pp. 15-26, 1998.