

Minimizing Request Blocking in All-Optical Rings

Christos Nomikos

Department of Computer Science
University of Ioannina, Greece

Email: cnomikos@cs.uoi.gr

Aris Pagourtzis

Department of Computer Science and
Department of Mathematics
National Technical University of Athens

Email: pagour@cs.ntua.gr

Stathis Zachos

CIS Department, Brooklyn College, CUNY and
Department of Computer Science
National Technical University of Athens

Email: zachos@cs.ntua.gr

Abstract—In all-optical networks that use WDM technology it is often the case that several communication requests have to be blocked, due to bandwidth and technology limitations. Minimizing request blocking is therefore an important task calling for algorithmic techniques for efficient routing and wavelength assignment.

Here we study the problem for rings under both the undirected and the directed settings, corresponding to symmetric and one-way communication respectively. The problem in graph-theoretic terms can be formulated as the MAXIMUM ROUTING AND PATH COLORING PROBLEM. We present a *Chain-and-Matching* technique for routing requests and coloring the corresponding paths which gives constant approximations for both the undirected and the directed cases. For the undirected problem we obtain a $\frac{2}{3}$ -approximation algorithm; this corresponds to a considerable increase in the number of satisfied requests compared to the best known algorithm so far, due to Wan and Liu [1], that achieves a $1 - \frac{1}{e}$ ratio using iteratively a maximum edge-disjoint paths algorithm. For the directed case, we also introduce a *Balanced Matching* method which, combined with the Chain-and-Matching technique, gives a $\frac{7}{11}$ -approximation algorithm. This algorithm also improves upon the $(1 - \frac{1}{e})$ -approximation algorithm that can be obtained by extending the iterative method of [1].

I. INTRODUCTION

Wavelength Division Multiplexing (WDM) is a developing fiber-optic transmission technique that allows several connections to be realized over a single fiber, providing very high transfer rate. However, in practice the number of wavelengths per fiber is limited to a few hundreds. A further restriction is imposed by the need for all-optical networks, due to the prohibitive cost of opto-electronic converters; in this setting, a connection must use the same wavelength from one end to the other. Due to these limitations, it is often the case that not all pending communication requests can be satisfied. It is therefore important to minimize the number of blocked requests (e.g. see the open problems proposed by Mihail, Kaklamanis and Rao [2]). In other words, we would like to be able to satisfy a maximum number of communication requests.

The problem can be formulated in graph-theoretic terms as follows:

MAXIMUM ROUTING AND PATH COLORING PROBLEM (MAXRPC)

Input: a graph G , a set of pairs of nodes (requests) R and a number of available colors w .

Feasible solution: an assignment of paths (routing) to a subset of requests $A \subseteq R$ and a coloring of these paths with different colors for overlapping paths.

Goal: maximize $|A|$, thus satisfying a maximum number of requests.

In optical networks, there are usually two opposite-directed fibers between connected nodes. Communication may be symmetric or one-way. In the symmetric case, after connection is established, data are sent back and forth (cf. telephone line); we may thus consider communication requests and connecting paths undirected. Two undirected paths *overlap* if they pass through the same edge. In the one-way case, datagram messages are sent (cf. telegram service). We may therefore treat requests as ordered pairs of nodes; paths are directed and two paths overlap if they pass through the same edge in the same direction. Thus, for our problem there is an undirected and a directed version corresponding to symmetric and one-way communication respectively. To avoid confusion we will call the directed version DIRECTED MAXRPC.

A. Our results

We present efficient approximation algorithms for MAXRPC and DIRECTED MAXRPC in rings. Both algorithms are based on the *Chain-and-Matching* technique which combines optimum solutions for chain instances with appropriate coloring according to a matching between requests. The algorithm for MAXRPC achieves approximation ratio $\frac{2}{3}$ which is a considerable improvement of the best known ratio of $1 - \frac{1}{e}$ [1].

For DIRECTED MAXRPC we develop a *Balanced Matching* method which, combined with the Chain-and-Matching technique gives an efficient approximation algorithm with ratio $\frac{7}{11}$. This is better than the $1 - \frac{1}{e}$ ratio which is the best possible that can be achieved using the iterative method of [3], [4], [1]—at least under the currently known analysis for this method. We show that indeed the $1 - \frac{1}{e}$ ratio is achievable by giving an exact algorithm for DIRECTED MAXRPC with one available color. It turns out that this second algorithm might achieve a better approximation guarantee only for small number of wavelengths ($w \leq 43$) and that its time complexity is worse than the complexity of our Chain-and-Balanced-Matching algorithm. To the best of our knowledge these are the first results for the DIRECTED MAXRPC problem in rings.

B. Related work

The MAXRPC problem has been studied for several topologies. In chains it coincides with the problem “ k -coloring

of intervals” which is solved exactly [5], [6]. Wan and Liu [1] present $(1 - \frac{1}{e})$ -approximation algorithms for MAXRPC in rings and trees and constant approximation algorithm for meshes. Using ideas from [3], [4] they solve MAXRPC by successive calls to an algorithm for MAXRPC with one available color (also known as the Maximum Edge-Disjoint Paths problem—MEDP). Using the same technique, Erlebach and Jansen [7] provide a $(1 - \frac{1}{e})$ -approximation algorithm for DIRECTED MAXRPC in bounded degree trees and a 0.451-approximation algorithm for general trees. The on-line version of MAXRPC (on-line call admission problem) has been studied in [4] where they give a general technique to obtain a $(\rho + 1)$ -competitive algorithm for arbitrary number of wavelengths from a ρ -competitive algorithm for one wavelength. Nomikos and Zachos [6] study the version where the routing is prescribed and give a $\frac{2}{3}$ -approximation algorithm for rings.

The most studied routing and wavelength assignment problem is the minimization problem ROUTING-AND-PATH-COLORING (RPC) where the goal is to satisfy all requests with a minimum number of colors. The problem can be solved exactly in polynomial time for chains (an old result, see e.g. [8]) as well as for bounded-degree trees [9]. RPC for rings is NP-hard [10]; 2-approximation algorithms are presented in [11] (undirected case) and in [2] (directed case). A version in which the routing is given (PC) is also well-known as the *circular-arc coloring problem*. PC for rings is proved to be NP-hard by Garey et al. [12]; Karapetian [13] presents a $\frac{3}{2}$ -approximation algorithm.

Other related work includes multiple fiber models [14], [15], [16], [17], [18] and models that allow wavelength conversion [19], [20], [21]. Another interesting approach is to consider a random distribution of requests and estimate blocking probabilities (e.g., see [22], [23], [24]).

C. Notation and Preliminaries

Here n will denote the number of vertices and m the number of requests ($m = |R|$); note that the number of edges is also n , whereas the number of arcs in the directed version is $2n$. Let Π be any of our maximization problems. We say that an algorithm for Π achieves approximation ratio $\rho \in (0, 1)$ if it produces solutions of cardinality at least $\rho \cdot OPT$ where OPT is the cardinality of the optimal (largest) solution.

A *ring* is a graph that consists of a single (simple) cycle. A *chain* is a ring from which one edge has been removed. Note that in chains, routing is unique for each request, while in rings two paths are possible (clockwise and counterclockwise) for each request.

Once a routing is determined, for each edge e we use the term *load of e* , $L(e)$ for short, to denote the number of paths that share e . The load of an instance, L for short, is the maximum load among all edges of the instance. For the directed case the definitions are similar, using arcs instead of edges.

In chains, MAXRPC can be solved exactly; an $O(m + w)$ time algorithm was presented by Carlisle and Lloyd in [5]. It is

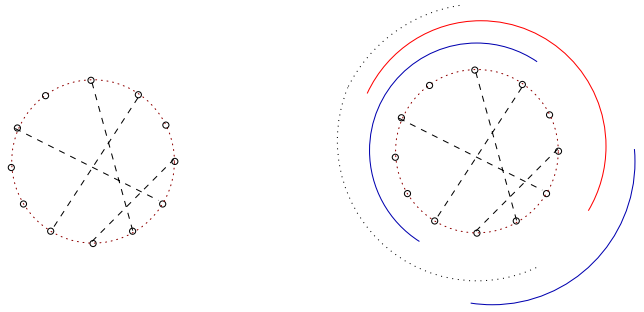


Fig. 1. An instance of MAXRPC in rings with 2 available colors and 3 satisfied requests. Paths corresponding to satisfied requests are represented by continuous lines.

well-known that for a chain instance of load L , it sufficient and necessary to use L colors to color all requests. If $w > L$ the algorithm in [5] colors all paths using exactly L colors. The algorithm works for the undirected version and the directed problem can easily be reduced to the undirected one. We therefore obtain the following fact.

Fact 1: Both undirected and directed versions of MAXRPC can be solved exactly in polynomial time if the input graph is a chain.

II. UNDIRECTED COMMUNICATION: A $\frac{2}{3}$ -APPROXIMATION ALGORITHM

MAXRPC is NP-hard even for rings, since there is a straightforward reduction from the decision version of ROUTING-AND-PATH COLORING (RPC) to the decision version of MAXRPC for any topology. NP-hardness for RPC in rings has been shown in [10].

An instance of MAXRPC in rings is shown in Figure 1.

In this section we present an approximation algorithm for MAXRPC in rings that produces solutions of cardinality at least $\frac{2}{3}OPT$. This algorithm illustrates our Chain-and-Matching technique on which the algorithm for DIRECTED MAXRPC (see Section III) is also based.

The algorithm is illustrated in Figure 2. The general idea is to transform the instance (Fig. 2(a)) to a chain instance (Fig. 2(b)) that can be solved optimally. We then take a second solution by making use of an appropriate matching (Fig. 2(c-d)). The final output is the largest cardinality solution.

The matching is taken over a *request compatibility graph* that is defined as follows. Each request corresponds to a geometric chord in the ring (circle). Two requests with crossing chords cannot be routed with non-overlapping paths. Note that in Figure 2(a) $\{2, 8\}$ crosses $\{1, 4\}$ while it does not cross $\{8, 6\}$. Two requests that do not cross are called *compatible*; two requests can be satisfied using the same color if and only if they are compatible. Incompatible and compatible requests are shown in Figure 3. The compatibility graph of a set of requests R is a graph $H = (R, E)$, in which the set of nodes is the set of requests R and the set of edges E contains all pairs of compatible requests.

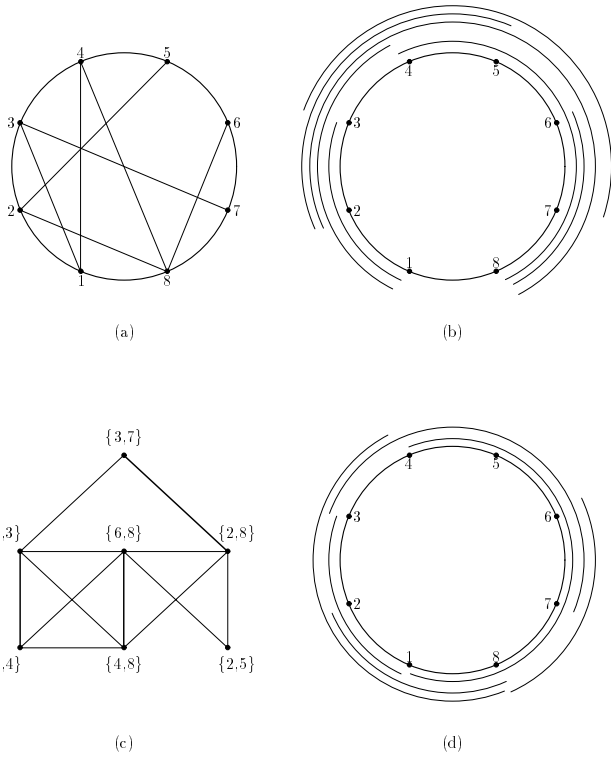


Fig. 2. (a) An instance of MAXRPC, requests are represented by chords. (b) Set of paths C routed avoiding edge $\{8, 1\}$. (c) A maximum matching on the request-compatibility graph H . (d) A routing by ALG_2 , according to the maximum matching.

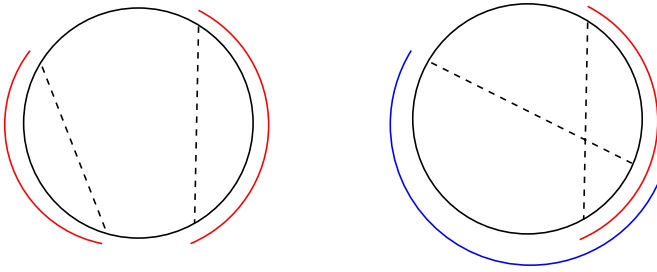


Fig. 3. Compatible (left) and incompatible (right) undirected requests.

ALG for MAXRPC

Input: ring G , set of requests R , number of colors w .

Output: paths and coloring for a subset of R .

Execute ALG_1 and ALG_2 independently. Output the largest cardinality solution.

ALG_1 Choose an edge e of G in an arbitrary way. Route requests so that paths avoid e . Let C denote the resulting set of paths. Color a maximum number of paths in C using an algorithm for chains (see Fact 1).

ALG_2 Find a maximum matching (of cardinality μ) in H , the request compatibility graph of R . Repeatedly route matched requests in a non-overlapping manner and color the two paths using the same color, until there are either no pairs left or no colors left.

Remark. A more sophisticated algorithm is possible, where the two solutions are combined: the chain solution is recolored according to an appropriate matching in such a way that the final solution is at least as good as the best between the chain and the matching solutions. The combination algorithm performs better in many cases but both algorithms achieve the same worst-case ratio. For the sake of clarity and brevity, we have chosen to present the simpler algorithm here.

Theorem 1: The above algorithm satisfies at least $\frac{2}{3}$ of the maximum number of simultaneously satisfiable requests.

Proof: Let SOL_1 (SOL_2) be the number of requests satisfied by ALG_1 (ALG_2) and SOL be the number of requests satisfied by ALG . Let also OPT denote the maximum number of requests in R that can be satisfied by any algorithm that uses w colors. Then the following hold:

(a) $SOL = \max(SOL_1, SOL_2)$.

(b) $OPT \leq SOL_1 + w$:

Consider any optimal solution A satisfying OPT requests. In A , the number of paths routed avoiding e (and colored) is at most SOL_1 , due to the optimality of the chain algorithm. In addition, A may only contain at most w paths routed through e , since these paths overlap.

(c) $OPT \leq SOL_1 + \mu$:

For an optimal solution A with requests routed through e and colored with a uniquely used color, there is another optimal solution B with such requests re-routed avoiding e (with the same coloring). The set of requests in B that are routed avoiding e (and colored) has cardinality at most SOL_1 , otherwise there would be a better solution for the chain instance of ALG_1 which would contradict the optimality of the chain algorithm used by ALG_1 . Thus, compared to SOL_1 , the optimal solution B may only contain additional satisfied requests routed through e and colored so that each one of them shares its color with a request routed avoiding e . Such pairs of requests correspond to adjacent vertices in the request compatibility graph H constructed by ALG_2 . Therefore, their number is at most μ .

(d) $OPT \leq SOL_1 + \min(\mu, w)$: from (b) and (c).

(e) $SOL_2 \geq 2 \min(\mu, w)$: ALG_2 routes and colors matched pairs of requests until there are either no more pairs or no more colors.

(f) $SOL \geq \frac{2}{3}OPT$: from (a), (d) and (e), we have

$$OPT \leq SOL_1 + \frac{1}{2}SOL_2 \leq \frac{3}{2}OPT.$$

Remark. Note also that $SOL_1 \geq w$ since all colors are used (otherwise, all requests are colored and $SOL = OPT$). Combined with (b) we get $SOL_1 \geq \frac{1}{2}OPT$, i.e. algorithm ALG_1 achieves approximation ratio $\frac{1}{2}$ for MAXRPC in undirected rings.

Complexity. Coloring paths in C can be performed in $O(m)$ time, using the chain algorithm of Carlisle and Lloyd [5], provided that paths are sorted according to their ending point. This sorting can be done in $O(m + n)$ time by bucket sort

(with one bucket for each vertex). The most expensive step of ALG for MAXRPC is the matching computation in ALG_2 , which can be done in $O(m^{2.5})$ time by using the algorithm in [25]. The remaining steps require $O(n + m)$ time. Thus, the total complexity is $O(n + m^{2.5})$.

III. DIRECTED COMMUNICATION: A $\frac{7}{11}$ -APPROXIMATION ALGORITHM

The DIRECTED MAXRPC problem corresponds to one-way communication, i.e. messages are sent only in one direction and any two connected nodes are linked with two arcs of opposite direction. For simplicity we will call these two opposite arcs an edge. Recall that in this case, two paths overlap only if they share a common *arc*. Thus, a clockwise and a counterclockwise path may always be colored with the same color. Therefore, w colors are available *per direction*.

This graph-theoretic description models realistic optical networks in which two opposite-directed fibers constitute a full-duplex link and each communication message travels only in one direction (datagram connectionless communication).

As in the undirected case, the DIRECTED MAXRPC for rings is an NP-hard problem. This can be proved using a simple reduction from the decision version of MAXRPC to the decision version of DIRECTED MAXRPC.

A. The Chain-and-Balanced-Matching Technique

The algorithm follows a similar strategy as the algorithm for undirected rings, using two subroutines, ALG_1 and ALG_2 . ALG_1 routes all paths avoiding an edge e (that is two opposite arcs) so that a chain instance occurs, whereas ALG_2 transforms request satisfaction to a matching problem. However, the request compatibility graph is now defined in a more sophisticated way. Namely, two requests can be satisfied using the same color in clockwise direction if and only if they can both be routed clockwise in a non-overlapping way (see Figure 4, bottom-left). This can be done if and only if their chords do not cross and in clockwise traversal the target of one request is followed by the source of the other; in this case we call the two requests *clockwise compatible*. Similarly we define *counterclockwise compatibility* (see also Figure 4, bottom-right). Incompatible requests are shown in top of Figure 4. Note that two requests can be both clockwise and counterclockwise compatible, if and only if they are symmetric, that is they are between the same pair of nodes but in opposite directions.

In order to simplify the algorithm presentation and analysis, we will assume that the set of requests does not contain any pair of symmetric requests; we will consider the modifications needed when symmetric requests appear at the end of this subsection. Under the above assumption, given a set of requests R , the request compatibility graph is an undirected graph $H = (R, E_{blue} \cup E_{red})$, where E_{blue} contains all pairs of clockwise compatible requests (blue edges) and E_{red} contains all pairs of counterclockwise compatible requests (red edges).

Coloring in ALG_2 is performed using a matching which is not necessarily maximum, but it is *balanced*. The idea behind

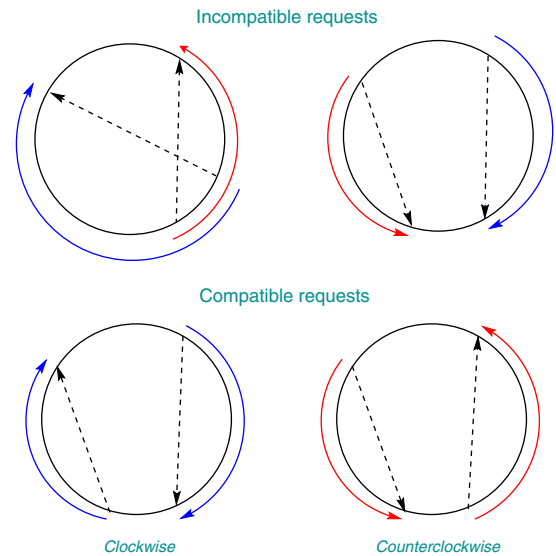


Fig. 4. Cases of incompatible and compatible directed requests.

balancing is that we can delete from a maximum matching a number of superfluous edges of one color, in order to increase, even by a smaller number, the edges of the other color. We require that the initial maximum matching is proper: a matching M of H is called *proper* if one of the following is true: (a) M consists of a single edge (b) M contains edges of both colors (c) all edges in H possess the same color. The next lemma shows that this requirement can be always satisfied:

Lemma 1: The request compatibility graph H of any set of directed requests R over a ring has a maximum matching which is proper.

Proof: Let M be a maximum matching in H which is not proper. W.l.o.g. we may assume that H contains at least one red edge (a, b) , M consists of only blue edges, and $|M| \geq 2$. Since M is maximum, at least one endpoint of (a, b) is matched. Assume that a is matched, that is $(a, a') \in M$. If b is not matched, then we can obtain a maximum matching with one red edge by replacing (a, a') by (a, b) . If however b is also matched, that is $(b, b') \in M$, then using the fact that (a, a') and (b, b') are blue and (a, b) is red, it is easy to prove that (a', b') is a blue edge of H . Hence, we can obtain a maximum matching with one red edge by replacing (a, a') and (b, b') by (a, b) and (a', b') . ■

Let M be a matching in H and M_{red} be a maximum matching in graph (R, E_{red}) . Consider the graph G resulting by superimposing M and M_{red} . The connected components of G are either cycles or chains, since every vertex has degree ≤ 2 . An edge e , which is common in M and M_{red} , belongs to a connected component that consists exactly of e . The remaining connected components alternate edges of M and M_{red} . Let $\mathcal{C}(M, M_{red})$ denote the set of all connected components in G , in which the number of edges that belong to M_{red} is greater than the number of red edges that belong to M . Notice that if $\mathcal{C}(M, M_{red}) = \emptyset$ then the number of red edges in M is maximum. Similar definitions can be given

using color *blue* instead of *red* in the above.

ALG for DIRECTED MAXRPC

Input: ring G , set of directed requests R , number of colors w .

Output: directed paths and coloring for a subset of R .

Execute ALG_1 and ALG_2 independently. Output the largest cardinality solution.

ALG_1 Choose an edge e of G in an arbitrary way. Route the requests so that connecting paths avoid e . Let C denote the resulting set of paths.

Color a maximum number of paths in C using an algorithm for directed chains (see Fact 1).

If free colors remain in a direction, re-route unsatisfied requests according to this direction and color the corresponding paths with these colors, one color for each path.

ALG_2 Construct the compatibility graph $H = (R, E_{blue} \cup E_{red})$, of R .

Find a maximum proper matching M in H .

Execute the *Balancing Procedure* to obtain a balanced matching M' .

Route matched pairs of requests according to the direction implied by the matching M' .

In each direction, repeatedly color two matched requests using one color, until there are either no more pairs or no more colors.

In each direction, if unused colors remain, route and color unsatisfied requests using one color for each.

Balancing Procedure

(W.l.o.g. we assume that M contains more blue edges than red edges. The other case is symmetric.)

Find a maximum red matching M_{red} and let $S = C(M, M_{red})$.

While M contains more than $w + 1$ blue edges and less than w red edges, and S is not empty, choose (arbitrarily) a connected component F in S , execute the Augmenting Step for F and delete F from S .

Augmenting Step

Let ν_1, ν_2 be the number of blue edges in M, F respectively. If $\nu_1 - \nu_2 < w$, then let F' be a chain of edges in F with $\nu_1 - w$ blue edges, which begins and ends with a blue edge. Otherwise let $F' = F$.

Delete from M all edges that belong to F' ; add to M all edges in F' that belong to M_{red} .

Lemma 2: Suppose that the augmenting step increases the number of red edges by a value k . Then the corresponding decrease in the number of blue edges is at most $k + 1$.

Proof: Let k_r be the number of edges in M_{red} that appear in F' . Then the number of edges in M that appear in F' is at most $k_r + 1$ (since F' alternates edges between M and M_{red}); a number $k_r - k$ of them are red. Consequently the number of blue edges in F' is at most $k_r + 1 - (k_r - k) = k + 1$. ■

Lemma 3: If the algorithm terminates due to $S = \emptyset$ then the number of red edges in M' is maximum.

Proof: Suppose that the i -th iteration of the while loop in the Balancing Procedure executes the Augmenting Step for the component F_i and let M_i be the resulting matching. Let $M_0 = M$. Observe that $C(M_{i+1}, M_{red}) = C(M_i, M_{red}) - \{F_i\}$. This implies that $C(M_{i+1}, M_{red}) = C(M, M_{red}) - \{F_0, \dots, F_i\}$. Consequently, if the algorithm terminates due to $S = \emptyset$ then $C(M', M_{red}) = \emptyset$. This means that the total

number of red edges in M' cannot be less than $|M_{red}|$. The lemma follows since M_{red} is a maximum red matching. ■

Theorem 2: The above algorithm satisfies at least $\frac{7}{11}$ of the maximum number of simultaneously satisfiable requests.

Proof: Let SOL_0 be the number of paths colored by the algorithm for directed chains, SOL_1 (SOL_2) be the number of requests satisfied by ALG_1 (ALG_2) and SOL be the number of requests satisfied by ALG . Let also OPT denote the maximum number of requests in R that can be satisfied by any algorithm that uses w colors per direction. Moreover, let $\mu, \mu_{blue}, \mu_{red}$ represent the size of the maximum matching in $H, (R, E_{blue}), (R, E_{red})$ respectively. Finally, assume that M contains μ_b blue and μ_r red edges ($\mu = \mu_b + \mu_r$) and M' contains μ'_b blue and μ'_r red edges.

Again, w.l.o.g. assume that neither ALG_1 nor ALG_2 satisfy all requests (otherwise $SOL = OPT$). Then the following hold:

- (a) $SOL_0 \leq SOL_1$.
- (b) $SOL = \max(SOL_1, SOL_2)$ and thus

$$SOL \geq SOL_i, \quad 0 \leq i \leq 2.$$

- (c) $SOL_1 \geq 2w$: all w colors, in both directions, are used since otherwise, all requests are colored—which contradicts the assumption.
- (d) $OPT \leq SOL_0 + 2w$:
Consider any optimal solution A satisfying OPT requests. In A the number of paths routed avoiding e (and colored) is at most SOL_0 , due to the optimality of the chain algorithm. In addition, A may only contain at most w paths per direction routed through e , since these paths overlap.
- (e) $SOL_1 \geq \frac{1}{2}OPT$: from (a), (c) and (d).
- (f) $OPT \leq SOL_1 + \mu$:

Following similar arguments as in (c) in proof of Theorem 1 we can show that, compared to SOL_1 , an optimal solution B may only contain additional satisfied requests routed through e and colored so that each request has the same color as a request routed avoiding e . Such pairs of clockwise (counterclockwise) routed requests correspond to adjacent vertices in graph H constructed by ALG_2 . Therefore, the number of these requests is at most μ .

- (g) $OPT \leq SOL_1 + \min(w, \mu_{blue}) + \min(w, \mu_{red})$:
By using separately for each direction the same arguments as in (d) and (f).
- (h) $OPT \leq SOL_1 + z$, $z = \min(\mu, \min(w, \mu_{blue}) + \min(w, \mu_{red}))$: from (f) and (g).
- (i) $SOL_2 \geq \frac{7}{4}z$:

W.l.o.g. we may assume that $\mu_b \geq \mu_r$.

Case 1: $\mu_b \leq w$. Then the augmenting step is never executed. All requests matched by $M' = M$ are satisfied, that is $SOL_2 \geq 2\mu \geq 2z$.

Case 2: $\mu_b > w$. Then obviously $\mu_{blue} > w$. Moreover the selection of F' guarantees that $\mu'_b \geq w$. We consider two subcases:

Case 2.1: $\mu'_r \geq \min(w, \mu_{red})$. Then

$$\begin{aligned} SOL_2 &\geq 2(\min(w, \mu'_b) + \min(w, \mu'_r)) \\ &\geq 2(w + \min(w, \mu_{red})) \\ &= 2(\min(w, \mu_{blue}) + \min(w, \mu_{red})) \\ &\geq 2z. \end{aligned}$$

Case 2.2: $\mu'_r < \min(w, \mu_{red})$. Then $\mu'_b = w + d$, where $d \in \{0, 1\}$. The balancing procedure in ALG_2 increases the number of red edges by at least $\frac{\mu_b - \mu'_b}{2}$, since in the worst case an execution of the augmenting step may replace two blue edges with one red edge. Therefore $\mu'_r \geq \mu_r + \frac{\mu_b - \mu'_b}{2} = \frac{2\mu_r + \mu_b - w - d}{2} \geq \frac{\mu_r + \mu_b - w}{2}$. To obtain the last inequality, we use the fact that $\mu_r \geq 1 \geq d$, which holds because M is a proper matching, and in this subcase $\mu_{red} > \mu'_r \geq 0$ and $\mu_b > w \geq 1$. Consequently,

$$\begin{aligned} SOL_2 &= 2w + 2\mu'_r + (w - \mu'_r) \\ &= 3w + \mu'_r \\ &\geq 3w + \frac{\mu_r + \mu_b - w}{2} \\ &= 3w + \frac{\mu - w}{2} \\ &= \frac{1}{2}(5w + \mu) \\ &\geq \frac{1}{2}\left(\frac{5}{2} \min(2w, \mu) + \min(2w, \mu)\right) \\ &= \frac{7}{4} \min(2w, \mu) \geq \frac{7}{4}z. \end{aligned}$$

(j) $SOL \geq \frac{7}{11}OPT$: from (b), (h) and (i) we have

$$OPT \leq SOL_1 + \frac{4}{7}SOL_2 \leq \frac{11}{7}SOL.$$

Remark. Note that here also, (e) implies $SOL_1 \geq \frac{1}{2}OPT$, i.e. algorithm ALG_1 achieves approximation ratio $\frac{1}{2}$ for DIRECTED MAXRPC in rings.

It remains to consider the general case, in which R may contain symmetric requests. In this case, the request compatibility graph is $H = (R, E_{blue} \cup E_{red} \cup E^*)$, where E^* is the set of edges that connect symmetric requests. Symmetric requests are both clockwise and counterclockwise compatible, that is edges in E^* can be colored either blue or red. We choose a color for each edge in E^* after the computation of a maximum matching M of H and before executing the balancing procedure. There are three cases:

- If M contains at least w edges of each color then no balancing is needed and therefore the coloring of E^* is irrelevant (e.g., we may color all edges in E^* red).
- If M contains at least w edges of one color (say blue) and less than w edges of the other color (red), then we color all uncolored edges red.
- Otherwise (M contains less than w edges of each color) we start coloring edges in $E^* \cap M$ blue until M contains exactly w blue edges or there are no more uncolored edges in

M ; in the former case we continue by coloring the remaining uncolored edges red.

If M is not a proper matching for the resulting graph we apply the transformation indicated in Lemma 1. Notice that for the above described general case the proof of Theorem 2 is similar, with μ_{blue}, μ_{red} representing the size of a maximum matching in $(R, E_{blue} \cup E^*), (R, E_{red} \cup E^*)$ respectively.

Complexity. The complexity of the algorithm is $O(m^{2.5})$. Notice that the balancing algorithm requires $O(m^{2.5})$ time to compute M_{red} and $O(m)$ time to perform the balancing.

B. Comparison with the Maximum Edge-Disjoint Paths technique

We now show that the idea of iterative application of an algorithm for the Maximum Edge-Disjoint Paths problem (MEDP) [1], [7] can be also applied here.

In our terminology we are looking for an algorithm that gives a maximum number of arc-disjoint paths; in other words, an algorithm that can route and color a maximum number of directed requests using one color in each direction. The iterative technique guarantees that if there is an exact algorithm for coloring with one color then there is an algorithm that colors with w colors which achieves an approximation ratio $1 - (1 - \frac{1}{w})^w \geq 1 - \frac{1}{e}$. This w -coloring algorithm consists of merely repeating the 1-coloring algorithm w times.

Indeed, the following algorithm solves exactly the DIRECTED MAXRPC problem for $w = 1$.

1-Coloring Algorithm for DIRECTED MAXRPC

Input: ring G , set of directed requests R .

Output: a maximum set of requests with corresponding paths that do not overlap (arc-disjoint paths).

Set $S = \emptyset$.

Repeat steps (a), (b), and (c) for each request $r \in R$:

(a) Route r clockwise with a path $p(r)$. Add r to S_r .

(b) Assign a path to each request so that $p(r)$ is avoided (if possible, otherwise remove the request). Find a maximum set of non-overlapping paths in each direction. Add the corresponding requests to S_r . If $|S_r| > |S|$ set $S = S_r$.

(c) For each request t do:

If t can be satisfied by a counterclockwise path $p(t)$ such that $p(t)$ and $p(r)$ share an edge (use opposite arcs), add r and t to $S_{r,t}$. Assign a path to each request so that common edges of $p(r)$ and $p(t)$ are avoided (if possible, otherwise remove the request). Find a maximum set of non-overlapping paths in each direction. Add the corresponding requests to $S_{r,t}$. If $|S_{r,t}| > |S|$ set $S = S_{r,t}$. Repeat steps (a), (b) and (c) using the symmetric directions. If a larger cardinality solution occurs set S to be this solution.

Output S .

It is not difficult to verify that the above algorithm examines all possible routings and therefore gives the maximum solution. We therefore have the following.

Corollary 1: An approximation ratio $1 - (1 - \frac{1}{w})^w > 1 - \frac{1}{e}$ can be achieved for the DIRECTED MAXRPC problem in rings, where w is the number of colors.

This iterative algorithm has a worse approximation guarantee than the Chain-and-Balanced-Matching algorithm for all $w > 43$ and it has a worse time complexity $O(wm^3)$. Therefore, it may be of some practical use only in very limited cases (mainly for small w).

IV. CONCLUSIONS – DISCUSSION

We studied the problem of satisfying a maximum number of communication requests in a ring and proposed algorithms for both the undirected and the directed versions of the problem.

By using the Chain-and-Matching technique we obtain a $\frac{2}{3}$ -approximation algorithm for MAXRPC; the best known ratio so far was $1 - \frac{1}{e}$ [1]. This represents a significant increase in the number of satisfied requests.

Combining this technique with a Balanced-Matching method we achieve a $\frac{7}{11}$ -approximation ratio for DIRECTED MAXRPC. To the best of our knowledge this is the first algorithm for the DIRECTED MAXRPC problem in rings. An interesting open question is whether this ratio can be further improved by a more accurate balanced matching method. In particular, Balanced Matching can be seen as an interesting open problem that arises from this work. Solving Balanced Matching exactly would result in a $\frac{2}{3}$ -approximation for the directed case as well.

Following a standard iterative method we have shown that we can obtain a second algorithm with approximation ratio $1 - (1 - \frac{1}{w})^w > 1 - \frac{1}{e}$. A comparison of the two algorithms shows that our $\frac{7}{11}$ algorithm outperforms the iterative one except, possibly, for few cases.

The Chain-and-Matching algorithms can be improved in order to perform better in the average. For example, we may use the maximum matching of ALG_2 to improve the solution of ALG_1 (i.e. a combination of ALG_1 and ALG_2 instead of a mere choice). This does not improve the worst-case approximation ratio but it certainly makes the algorithms more useful in practice.

On the other hand, sometimes the algorithm can be terminated earlier. For example, in ALG for MAXRPC, if ALG_1 satisfies more than $2w$ requests, ALG_2 does not need to be executed because $SOL_1 \geq SOL_2$. In any case, if a ratio $\frac{1}{2}$ is acceptable, implementing only ALG_1 suffices; the advantages are the simplicity of implementation and the linear time complexity of ALG_1 for all problems.

An interesting open problem is to determine upper bounds for the approximation ratios of MAXRPC and DIRECTED MAXRPC in rings, i.e. bounds on how close to 1 these ratios can be. As far as we know no such bounds have been shown yet.

REFERENCES

[1] P. J. Wan and L. Liu, "Maximal throughput in wavelength-routed optical networks," *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 46, pp. 15–26, 1998.

[2] M. Mihail, C. Kaklamanis, and S. Rao, "Efficient access to optical bandwidth," in *Proceedings of FOCS*, 1995, pp. 548–557.

[3] G. Cornuejols, M. Fisher, and G. Nemhauser, "Location of bank accounts to optimize float," *Management Science*, vol. 23, pp. 789–810, 1977.

[4] B. Awerbuch, Y. Azar, A. Fiat, S. Leonardi, and A. Rosen, "On-line competitive algorithms for call admission in optical networks," *Algorithmica*, vol. 31(1), pp. 29–43, 2001.

[5] M. C. Carlisle and E. L. Lloyd, "On the k -coloring of intervals," *Discrete Applied Mathematics*, vol. 59, pp. 225–235, 1995.

[6] C. Nomikos and S. Zachos, "Coloring a maximum number of paths in a graph," in *ICALP Satellite workshop on Algorithmic Aspects of Communication*, 1997.

[7] T. Erlebach and K. Jansen, "The maximum edge-disjoint paths problem in bidirected trees," *SIAM Journal on Discrete Mathematics*, vol. 14(3), pp. 326–366, 2001.

[8] S. Olariu, "An optimal greedy heuristic to color interval graphs," *Information Processing Letters*, vol. 37, pp. 21–25, 1991.

[9] C. Nomikos, "Path coloring in graphs," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, NTUA, 1997.

[10] T. Erlebach and K. Jansen, "The complexity of path coloring and call scheduling," *Theoretical Computer Science*, vol. 255(1-2), pp. 33–50, 2001.

[11] P. Raghavan and E. Upfal, "Efficient routing in all-optical networks," in *Proceedings of STOC*, 1994, pp. 134–143.

[12] M. Garey, D. Johnson, G. Miller, and C. Papadimitriou, "The complexity of coloring circular arcs and chords," *SIAM Journal on Algebraic and Discrete Methods*, vol. 1(2), pp. 216–227, 1980.

[13] I. A. Karapetian, "On the coloring of circular arc graphs," *Dokladi (Reports) of the Academy of Science of the Armenian Soviet Socialist Republic*, vol. 70(5), pp. 306–311, 1980, in Russian, English translation by D. Gamarnik, D. Williamson and N. Edwards in <http://www.orie.cornell.edu/~nedwards/wdm-routing/>.

[14] C. Nomikos, A. Pagourtzis, and S. Zachos, "Efficient coloring with applications in multiwavelength routing," in *ICALP Satellite workshop on Algorithmic Aspects of Communication*, 1997.

[15] —, "Routing and path multi-coloring," *Information Processing Letters*, vol. 80(5), pp. 249–256, 2001.

[16] G. Li and R. Simha, "On the wavelength assignment problem in multifiber WDM star and ring networks," in *Proceedings of INFOCOM*, 2000, pp. 1771–1780.

[17] L. Margara and J. Simon, "Wavelength assignment problem on all-optical networks with k fibres per link," in *Proceedings of ICALP*, 2000, pp. 768–779.

[18] —, "Decidable properties of all-optical networks," in *Proceedings of ICALP*, 2001, pp. 518–529.

[19] S. Subramaniam, M. Azizoglu, and A. K. Somani, "On optimal converter placement in wavelength-routed networks," *IEEE/ACM Transactions on Networking*, vol. 5(7), pp. 754–766, 1999.

[20] R. Ramaswami and G. Sasaki, "Multiwavelength optical networks with limited wavelength conversion," *IEEE/ACM Transactions on Networking*, vol. 6(6), pp. 744–754, Dec. 1998.

[21] V. Sharma and E. Varvarigos, "Limited wavelength translation in all-optical WDM mesh networks," in *Proceedings of INFOCOM*, 1998, pp. 893–901.

[22] R. A. Barry and P. A. Humblet, "Models of blocking probability in all-optical networks with and without wavelength changers," *IEEE Journal on Selected Areas in Communications*, vol. 14(5), pp. 858–867, 1996.

[23] Y. Zhu, G. N. Rouskas, and H. G. Perros, "A path decomposition approach for computing blocking probabilities in wavelength routing networks," *IEEE/ACM Transactions on Networking*, vol. 8(6), pp. 747–762, 2000.

[24] S. Ramesh, G. N. Rouskas, and H. G. Perros, "Computing blocking probabilities in multi-class wavelength routing networks with multicast calls," *IEEE Journal on Selected Areas in Communications*, vol. 20(1), pp. 89–96, 2002.

[25] S. Micali and V. Vazirani, "An $O(\sqrt{VE})$ algorithm for maximum matching in general graphs," in *Proceedings of FOCS*, 1980, pp. 17–27.