On-line survivable routing in WDM networks

Anteneh A. Beshir, Fernando A. Kuipers, Piet F.A. Van Mieghem
Network Architectures and Services
Delft University of Technology
Mekelweg 4, 2628 CD Delft
The Netherlands
Emails: {A.A.Beshir, F.A.Kuipers, P.F.A.VanMieghem}@tudelft.nl

Ariel Orda Department of Electrical
Engineering Technion,
Israel Institute of Technology
Haifa, Israel 32000
Email: ariel@ee.technion.ac.il

March 6, 2009

Abstract

In WDM networks, survivable routing and wavelength assignment (SRWA) involves assigning link-disjoint primary and backup lightpaths. In the on-line SRWA problem, a sequence of requests arrive and each request is either accepted or rejected based only on the input sequence seen so far. For special networks, we establish on-line algorithms with constant and logarithmic competitive ratios. It is not possible to obtain good competitive ratios in general topologies. Hence, we propose heuristic schemes and evaluate their performance by way of simulations. The building blocks in these schemes are 2-approximation algorithms (*MSA* and *ESA*) that we establish for the minimum disruption link-disjoint paths (MDLDP) problem. These approximations require far less memory and computation time than the best-known exact solution of the MDLDP problem. We use these three algorithms as heuristics for the on-line SRWA problem for infinite and finite duration requests and we show that, in terms of on-line performance, our algorithms do as well as (even at times better than) the exact algorithm of the MDLDP problem. We also provide an exact ILP formulation to solve the infinite duration off-line SRWA problem.

1 Introduction

In optical networks employing wavelength-division multiplexing (WDM), the enormous capacity of a fiber is divided into several non-overlapping wavelength channels that can transport data independently. These wavelength channels make up *lightpaths*, which are used to establish point-to-point optical connections that may span several fiber links without using routers. In wavelength-selective WDM networks, a lightpath connection between a source and a destination must have the same wavelength in all links along its route. In wavelength-interchanging WDM networks, the nodes have the capability to convert a wavelength at an incoming link to a different one at an outgoing link. Unfortunately, the high price of wavelength converters makes them less desirable. Therefore, in this paper we only focus on wavelength-selective networks.

In WDM networks, provisioning lightpaths involves not only routing, but also wavelength assignment and this problem is referred to as the *routing and wavelength assignment (RWA)* problem. Due to the tremendous amount of data transported, survivability, which is the ability to reconfigure and re-establish communication upon failure, is indispensable in WDM networks. Since in reality not all the links fail at the same time, we consider the *single-link failure model*, where at most a single link fails at any given time. The *survivable routing and wavelength assignment (SRWA)* problem is to assign, given a set of lightpath requests, link-disjoint primary and backup lightpaths to each request so that the total number of accepted requests is maximized.

Considering only one request, the SRWA problem is easily solved with Suurballe's algorithm [14] when the primary and backup lightpaths use the *same wavelength* (for different wavelengths it is NP-complete even for a single request [3]). However, in practice, lightpath requests arrive over time and the decision to accept or reject a request is made without any knowledge of future requests, yet maintaining the goal to maximize the total number of accepted requests. This version of the SRWA problem is known as *on-line SRWA*. An algorithm is said to be an on-line algorithm if, for any arbitrary input sequence σ, at any point in the sequence a decision is made based on the input seen so far and without any knowledge of the future. On the other hand, an off-line algorithm is assumed to know the whole input sequence. Thus, the performance of an on-line algorithm A can at best be as good as an optimal, but usually non-implementable, off-line algorithm OPT.

Definition 1 *An on-line algorithm A is said to be ρ-competitive if for any input sequence σ,*

$$\mathcal{B}(A,\sigma) \geq \frac{1}{\rho}\mathcal{B}(OPT,\sigma)$$

where $\mathcal{B}(X,\sigma)$ is the number of accepted requests by algorithm X for the input sequence σ. The smallest such ρ is called the competitive ratio *of the algorithm.*

In Section 2, we provide algorithms for the on-line SRWA problem with constant and logarithmic competitive ratios for specific networks. In Section 3, we introduce rerouting of lightpaths to improve the practical performance of on-line routing. To this end, we discuss a related problem called the *minimum disruption link-disjoint paths (MDLDP)* problem and provide two 2-approximation algorithms for solving it. An algorithm is a *2-approximation* algorithm for

MDLDP if for any request, the number of lightpaths rerouted by its solution is at most twice that of the optimal algorithm. In Sections 4 and 5, we employ these algorithms as heuristics to solve the on-line SRWA with rerouting problem for requests of infinite and finite duration, respectively.

2 On-line SRWA

The on-line survivable routing and wavelength assignment (SRWA) problem is defined as follows.

Problem 1 ***On-line SRWA****: The physical optical network is modeled as an undirected graph $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is a set of N nodes and $\mathcal{L}$ is a set of L links. Each fiber link has a set of W wavelengths, $\Lambda = \{w_1, w_2, \ldots, w_W\}$. A sequence of lightpath requests σ arrive over time. Each request $i \in \sigma$ is represented by (s_i, t_i), where $s_i, t_i \in \mathcal{N}$ are its source and destination nodes, respectively. The on-line SRWA problem is to allocate for each request link-disjoint primary and backup lightpaths such that (1) the same wavelength is used on all links of the primary and bakup lightpaths, (2) no two lightpaths having the same wavelength can share a link, and (3) the decision to accept or reject a request is based only on the input sequence seen so far. The objective is to maximize the number of accepted requests.*

Before addressing the on-line SRWA problem, we consider the on-line SRWA problem without survivability (on-line RWA) and other related problems that have been studied in the literature.

Problem 2 ***On-line Maximum Disjoint Paths (MDP) Problem****: Given are a graph $G(\mathcal{N}, \mathcal{L})$ and a sequence of requests. For each request (s_i, t_i), find a path P_i that connects s_i and t_i such that no two paths share the same link. The objective is to maximize the total number of accepted requests.*

The MDP problem is NP-complete [10]. Since lightpaths on the same wavelength are not allowed to share a link, the on-line MDP problem is equivalent to the on-line RWA problem with $W = 1$. Awerbuch *et al.* [7] have shown that if there is a ρ-competitive algorithm for the on-line MDP problem, then a $(\rho + 1)$-competitive algorithm can be obtained for the on-line RWA problem by employing the on-line MDP algorithm on each wavelength.

The on-line MDP problem has been widely studied in the literature. The $\Omega(N^a)$, where $a = \frac{2}{3}(1 - \log_4 3)$ lower bound given by Bartal *et al.* [8] for randomized algorithms shows that it is not possible to find a good competitive ratio for general networks. In fact, most of the work in the literature has been restricted to special networks such as lines, trees, lattices, tree of rings, etc.

Problem 3 ***On-line k Maximum Disjoint Paths (k-MDP) Problem****: Given are a graph $G(\mathcal{N}, \mathcal{L})$ and a sequence of requests. For each request (s_i, t_i), find k link-disjoint paths $P_{i1}, ..., P_{ik}$ that connect s_i and t_i such that no two paths* *of different requests share the same link. The objective is to maximize the total number of accepted requests.*

A simple upper-bound of any non-preemptive on-line algorithm for k-MDP is $O(\frac{L}{k})$. Suurballe's [14] algorithm ($k = 2$) has a competitive ratio equal to this upper-bound. For example in Figure 1, if the input sequence is (s, t) followed by (s, a_1), (a_1, a_2),..., (a_y, t), (s, b_1), (b_1, b_2),..., (b_y, t) and all links have equal cost, the off-line algorithm accepts $O(N)$ requests (i.e., all except the first), but the on-line algorithm accepts only the first two requests. Since in this example $L = O(N)$, the competitive ratio is of the same order as the upper-bound.

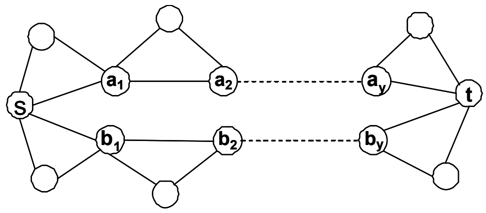

Figure 1: An example where Suurballe's algorithm attains the upper bound.

Using the same argument provided by Awerbach *et al.* [7], a (ρ+1)-competitive algorithm for the on-line SRWA problem ($W > 1$) can be derived from a ρ-competitive algorithm of the on-line 2-MDP problem. Hence, in the remainder of this section, we provide algorithms and corresponding competitive ratios for the on-line 2-MDP problem, which forms the basis for the on-line SRWA problem, in star-of-rings, tree-of-rings, and lattice networks. Even though these are simple networks, not only do they help us gain insight into the problem, but they are also used in real networks (e.g., the SURFnet network in the Netherlands resembles a star-of-rings[1]).

2.1 Star-of-rings network

Algorithm 1 *Star_Alg*(G, s, t)

- Accept a request if it is the first request so far that uses the ring(s) to which the source and destination nodes belong.
- Reject, otherwise.

Star_Alg(G, s, t) is 2-competitive if the number of rings is greater than 1. For a single ring, it is optimal. Figure 2 shows an example where *Star_Alg*(G, s, t) is 2-competitive for the input sequence (a, b), (b, c), (a, d). In this example, the on-line algorithm accepts only the first request, while the off-line algorithm accepts the last two requests.

[1] http://www.surfnet.nl/en

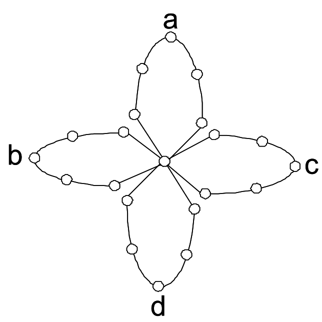

Figure 2: A star of rings containing four rings.

2.2 Tree-of-rings network

Algorithm 2 $Tree\_Alg(G, s, t)$

- Replace each ring by a single link so that the whole tree of rings is substituted by the underlying tree topology.
- Each 2-MDP request in the tree of rings is equivalent to a corresponding MDP request in the underlying tree.
- Use the algorithm of Awerbuch *et al.* [5], which has $O(\log N)$ competitive ratio for a tree of N nodes, to solve the on-line MDP problem.

From [5], it follows that $Tree\_Alg(G, s, t)$ is $O(\log \Upsilon)$-competitive, where Υ is the number of rings.

2.3 Lattice network

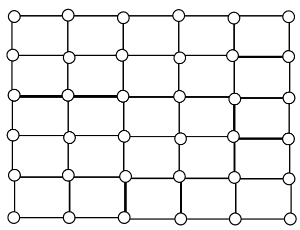

Figure 3: 6×6 lattice network.

The $O(\log N)$-competitive algorithm given by Kleinberg and Tardos [11] for the MDP problem can, with a slight modification, be used for solving the 2-

MDP problem with an $O(\log N)$-competitive ratio. Given an $N = n \times n$ lattice network, the outline of their algorithm is as follows.

- Classify each request as either "short" or "long," depending on the shortest distance $dist(s_i, t_i)$ (in terms of hopcount) between its source and destination nodes. A request is said to be short if $dist(s_i, t_i) < 16\gamma \log n$ for a given constant $\gamma > 1$; and long otherwise.
- Choose (randomly) to accept only short or only long calls.
- Create a "simulated network" whose vertices are subsquares of the original $n \times n$ lattice network and each of its links contain $O(\log n)$ links of the original network.
- Map the requests onto the simulated network (details in [11]).
- For long requests, a modified version of the AAP algorithm [4] is used to route the requests as shown in [11]. The modified AAP algorithm is given and validated in Appendix A.1.
- For short requests, a modified version of the algorithm from [11] is used. The modified algorithm is given and validated in Appendix A.2.

3 On-line SRWA with Rerouting

In Section 2, we provided algorithms for the on-line 2-MDP problem in specific networks, which can be used to derive corresponding algorithms for the on-line SRWA problem. Unfortunately, it is not possible to attain a good competitive ratio for general networks [8]. In this section, we explore the idea of rerouting lightpaths to improve performance. Although rerouting does not improve the competitive ratio, we show through simulations that it can increase the acceptance rate considerably. In wavelength-selective WDM networks, a rerouting procedure may be path rerouting (i.e., changing the route of a lightpath while keeping the wavelength), wavelength rerouting (i.e., changing the wavelength while keeping the path) or a combination of both. Compared to path rerouting, wavelength rerouting does not need extra path computation (as it retains the same path), facilitates control and, if the rerouted lightpath is moved to a vacant route on another wavelength, it incurs less traffic disruption [12]. We therefore focus on wavelength rerouting.

Generally, the wavelength rerouting problem is NP-complete [12]. The problem consists of solving the three possible scenarios presented below. The second and the third scenarios make the problem hard to solve.

1. When the lightpaths to be rerouted are on the same wavelength, they can be moved to vacant wavelengths in parallel without any conflict (since they do not share links). For example, in Figure 4(a), a new lightpath from node 1 to 5 can be accepted on wavelength w_2 by rerouting lightpath p_3 to w_1 and p_4 to w_3 in parallel.

2. When the lightpaths are on different wavelengths, moving to vacant wavelengths can be done sequentially while checking for conflicts. For example, in Figure 4(b), a new lightpath from node 1 to 5 can be accepted on w_1 by first rerouting p_4 to w_3 and then p_1 to w_2.

3. Moving to a vacant wavelength may not be sufficient, and it may be necessary to swap the wavelengths of lightpaths. For example, in Figure 4(c), a new lightpath from node 1 to 4 can be accepted on w_2 by swapping the wavelengths of p_2 and p_3.

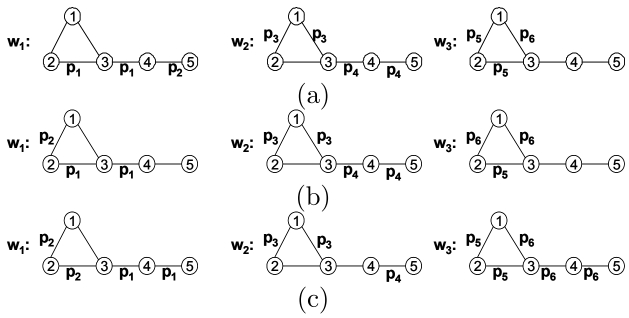

Figure 4: Different scenarios of wavelength rerouting: (a) moving to vacant, (b) sequential rerouting, and (c) swapping.

In the literature and the remainder of this paper, the term *wavelength rerouting* is used to refer to the reduced problem, i.e., assigning a lightpath by moving existing lightpaths on the same wavelength to vacant wavelengths in parallel. Xue [16] has shown that this problem can be solved in $O(WN \log N + WL)$ time.

On-line SRWA with wavelength rerouting involves assigning link-disjoint primary and backup lightpaths for new requests by rerouting, if necessary, already existing lightpaths. When rerouting lightpaths, the number of lightpaths rerouted should be kept to a minimum. This leads us to consider the minimum-disruption link-disjoint paths (MDLDP) problem. The MDLDP problem is NP-complete when the primary and backup lightpaths use different wavelengths. However, it is polynomially solvable for the same wavelength [15]. We consider the polynomially-solvable version.

Problem 4 ***Minimum Disruption Link-Disjoint Paths (MDLDP):*** *The physical optical network is modeled as an undirected graph $G(\mathcal{N}, \mathcal{L})$, where $N = |\mathcal{N}|$ and $L = |\mathcal{L}|$. Each fiber link has a set $\Lambda = \{w_1, w_2, \ldots w_W\}$ of W wavelengths. Given a request i, the MDLDP problem is to allocate on the same wavelength link-disjoint primary and backup lightpaths for request i, while minimizing the number of lightpaths to be rerouted.*

Wan and Liang [15] provided an $O(WL^5 \log N)$ exact algorithm for solving the MDLDP problem. We refer to this algorithm as WLA. WLA has a very high running time and requires a large amount of memory. This makes it less suitable, especially in an on-line setting where the algorithm has to be invoked whenever a new request arrives. We propose two 2-approximation algorithms with considerably less running time and memory requirements.

3.1 2-Approximation Algorithms for MDLDP

We provide two 2-approximation algorithms for MDLDP: MSA and ESA. MSA is a modified version of Suurballe's algorithm [14] with a running time of $O(WN \log N + WL)$ and ESA is an extended algorithm with a running time of $O(WN^2 \log N + WNL)$. This is a significant reduction from the $O(WL^5 \log N)$ running time of the exact WLA algorithm with at most twice as much lightpaths being rerouted.

In our notation, we use p to represent a lightpath and P to represent any path. A lightpath on wavelength w_i is said to be *reroutable*, if and only if all of its links are free on at least one other wavelength w_j, i.e., no lightpath is using these links on w_j. A path P from s to t is said to *traverse* a lightpath p if it shares at least one link with p. Let $\mathcal{P}_k$ be the set of lightpaths on wavelength w_k; $\mathcal{P}_k^{'} \subseteq \mathcal{P}_k$ be the set of reroutable lightpaths on wavelength w_k; $\mathcal{P}_k^{''} = \mathcal{P}_k \backslash \mathcal{P}_k^{'}$ be the set of non-reroutable lightpaths on wavelength w_k; and $\Lambda_{(i,j)}$ be the set of free wavelengths on fiber link (i, j).

We identify W subgraphs, $G_k = G(\mathcal{N}, \mathcal{L}_k)$, $\mathcal{L}_k = \{(i,j) \in \mathcal{L} \mid w_k \in \Lambda_{(i,j)}$ or $\exists\ p \in \mathcal{P}_k^{'}$ such that link (i, j) belongs to lightpath $p\}$. The cost of a link (i, j) in subgraph G_k is $cost_k(i, j) = \epsilon$, if (i, j) is a free link, where[2] $2N\epsilon < 1$; $cost_k(i, j) = 1$ otherwise. However, the cost $cost_k(P)$ of a path P in subgraph G_k is the sum of the cost of its free links and the number of *distinct* reroutable lightpaths traversed by P, i.e., multiple links belonging to a lightpath are counted only once. Thus, the shortest path between two nodes traverses the minimum number of reroutable lightpaths. Note that any lightpath that is traversed by the shortest path is encountered only once, i.e., in a single segment (of possibly multiple links).

In Step $1a$ of the MSA algorithm, we find the shortest path from s to t (using an algorithm such as the one given in [16]). In Step $1b$, the cost of all links belonging to lightpaths traversed by the shortest path is set to zero so that these links are preferred in the second path and the lightpaths are not counted twice. Similarly, the cost of free links on the shortest path is set to $-\epsilon$.

Theorem 1 *MSA is a 2-approximation algorithm for the MDLDP problem.*

Proof. Since the best solution is chosen after independently considering each wavelength, it suffices to consider only the wavelength that provides the best solution. Assume that for this wavelength, given that a solution of MSA that traverses a total of K lightpaths, there is an optimal solution that traverses

[2] Using such a cost, the longest possible link-disjoint paths made up of only free links have a total cost that is less than any link-disjoint pair of paths that cross a lightpath.

Algorithm 3 $MSA(G, s, t)$

1. For each G_k, $k = 1, \ldots, W$
   (a) In graph G_k, find the shortest path from s to t.
   (b) Graph $G_k^{'}$ is obtained by directing each link (i, j) of the shortest path from t to s, setting the cost of the free links on the shortest path as $cost_k(j, i) = -cost_k(i, j)$ and the cost of *all links* of lightpaths that are traversed by the shortest path to zero.
   (c) Find the shortest path from s to t in $G_k^{'}$.
   (d) If the shortest path exists in $G_k^{'}$, remove all the overlapping links between the two paths in G_k to obtain the solution.
2. Choose the best solution among all wavelengths.

Algorithm 4 $ESA(G, s, t)$

1. For each G_k, $k = 1, \ldots, W$
   (a) For each node $u \in \mathcal{N} \backslash \{s, t\}$:
      i. In graph G_k, find the shortest path P_{s-u} from s to u.
      ii. Graph $G_k^{'}$ is obtained from G_k by setting the cost of all links on P_{s-u} and each link belonging to lightpaths on P_{s-u} to infinity except for links of the lightpath (if any) in the last link of P_{s-u}. For the lightpath in the last link, all its links except the ones in P_{s-u} will have a cost of zero.
      iii. In graph $G_k^{'}$, find the shortest path P_{u-t} from u to t. If P_{s-u} and P_{u-t} share nodes, go to Step 1a-i if there are remaining nodes whose shortest paths have not been found, otherwise go to Step 1b. If P_{s-u} and P_{u-t} do not share nodes, the shortest path through u is found by concatenating the two.
      iv. Graph $G_k^{''}$ is obtained from G_k by directing each link (i, j) along the shortest path from t to s. The cost of free links on the shortest path is set to $cost_k(j, i) = -cost_k(i, j)$ and the cost of all links belonging to lightpaths on the shortest path is set to zero.
      v. In graph $G_k^{''}$, find the shortest path from s to t.
      vi. If the shortest path exists, remove all the overlapping links.
   (b) Choose the best solution among all nodes.
2. Choose the best solution among all wavelengths.

less than $\frac{K}{2}$ lightpaths, which would violate the claim of 2-approximation. Our intention is to prove that the assumption is wrong.

Let $\ell(P)$ represent the number of lightpaths traversed by a path P and $\ell(\{P_1, P_2\})$ represent the number of distinct lightpaths traversed by paths P_1 and P_2, where $\ell(\{P_1, P_2\}) \leq \ell(P_1) + \ell(P_2)$.

Let $\{P_1^*, P_2^*\}$ be the optimal solution. In MSA, let P_1 be the first shortest path that is obtained in Step 1a and P_2 be the second shortest path that is obtained in Step 1c.

Let $\mathcal{Q}$ be the set of alternating lightpaths of the optimal solution $\{P_1^*, P_2^*\}$, i.e., lightpaths with segments in both P_1^* and P_2^*. Let $\mathcal{S}$ be the set of links of lightpaths $p \in \mathcal{Q}$.

$\ell(\{P_1^*, P_2^*\}) < \frac{K}{2}$ implies that $\ell(P_1^*) < \frac{K}{2}$ and $\ell(P_2^*) < \frac{K}{2}$. Therefore, the first shortest path returned by MSA must have $\ell(P_1) < \frac{K}{2}$. Since $\ell(\{P_1, P_2\}) = K$, the second shortest path returned by MSA is assumed to have $\ell(P_2) > \frac{K}{2}$. However, MSA can find a path P_2 from the set of links of P_1^*, P_2^* and $\mathcal{S}$. If P_1 also contains any of these links, they are redirected in Step 1b of MSA and are assigned a cost of zero. Since no new lightpaths are added $\ell(P_2) < \frac{K}{2}$, which is a contradiction. ■

The 2-approximation is attained in the worst case when $\ell(P_1) = \ell(P_2) = \ell(\{P_1^*, P_2^*\})$ and P_1 and P_2 do not have common lightpaths as shown in Figure 5(a). $P_1 = \{s, 3, t\}$, $P_2 = \{s, 4, t\}$, $P_1^* = \{s, 1, 2, t\}$, and $P_2^* = \{s, 5, 6, t\}$; $\ell(\{P_1, P_2\}) = 2$ and $\ell(\{P_1^*, P_2^*\}) = 1$.

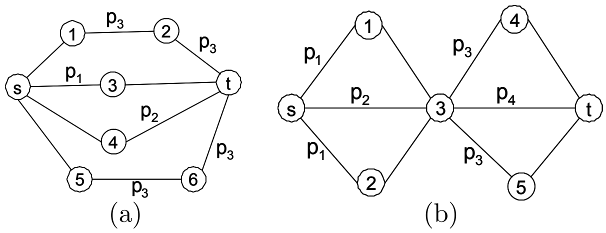

Figure 5: (a) A worst case for MSA that leads to a 2-approximation and (b) an example where ESA fails.

The example in Figure 5(a) can exactly be solved if P_1 leaves the source node through node 1 or node 5. We can achieve this by extending the MSA algorithm so that it checks the shortest path through any given node $u \in \mathcal{N} \backslash \{s, t\}$. As it can be seen later in Section 3.3, our extended algorithm ESA has a significantly improved performance in solving the MDLDP problem. But it fails for cases like the one in Figure 5(b), where $P_1 = \{s, 1, 3, t\}$, $P_2 = \{s, 2, 3, 5, t\}$, $P_1^* = \{s, 1, 3, 4, t\}$, and $P_2^* = \{s, 2, 3, 5, t\}$; $\ell(\{P_1, P_2\}) = 3$ and $\ell(\{P_1^*, P_2^*\}) = 2$.

In ESA, for each node $u \in \mathcal{N} \backslash \{s, t\}$, we find link-disjoint paths from s to t, where the first path is forced to go through u. In Step 1a-ii, the cost of all links on P_{s-u} and all links belonging to lightpaths on P_{s-u} (except those of the lightpath on the last link, if there is any) is set to infinity. This is to prevent the same links from being used again in P_{u-t} and to make sure that any lightpath in P_{s-t} is traversed in at most one segment. For the lightpath on the last link, since our interest is to find the shortest path from s to t through u, it can still be encountered *just* after u. Therefore, its links, except those in P_{s-u}, will have a cost of zero. In Step $1a$-iii, the shortest path from u to t is found. If P_{s-u} and P_{u-t} share nodes, then the algorithm does not proceed to finding the second shortest path. Instead, it skips to searching for the solutions of the remaining nodes. Once the path through u is found by concatenating P_{s-u} and P_{u-t}, the links on this path are directed from t to s in Step $1a$-iv. In Step $1b$, all the solutions are compared and the one that traverses the minimum number of lightpaths is chosen. If there are multiple solutions that traverse the same number of lightpaths, the one with the *smallest hopcount* is chosen. Since ESA includes MSA, it is at worst a 2-approximation algorithm.

3.2 Reroutability Status Update Procedure

Once a lightpath request is accepted and its link-disjoint lightpaths are determined, it affects the reroutability of other lightpaths. These lightpaths include the rerouted lightpaths, and lightpaths that are using the same link, but on different wavelengths. In addition, the reroutability of the new lightpaths has to be identified. Once a request is accepted, its primary and backup lightpaths are treated independently, i.e., each can be rerouted to a different wavelength independently of the other. Hence, as in [13], for each lightpath, we dynamically keep track of such information as its hopcount, its wavelength, how many of its links are free on other wavelengths and to which other wavelengths it can be rerouted to. This is done as follows.

1. When a new lightpath p is assigned without rerouting other lightpaths on wavelength w_k:
    - We create new reroutability status information for p, e.g., how many of its links are free on other wavelengths and the wavelengths it can be rerouted to. This takes $O(NW)$ time.
    - After checking whether p is reroutable or not, we assign the costs of its links on wavelength w_k. This takes $O(N)$ time.
    - In addition, the reroutability status information of lightpaths using the same fiber link, but on other wavelengths, should be updated. If q is such a lightpath, the number of its links that are free on wavelength w_k is decremented by one for each link that p and q have in common. Thus, if q was reroutable to wavelength w_k, it is not any more. Since, in the worst case, there are $O(NW)$ such lightpaths, this takes $O(NW)$ time.
2. When a new lightpath p is assigned by rerouting some lightpaths on wavelength w_k:
    - All the aforementioned operations are performed.

- If q is a rerouted lightpath, the costs of its links on the new wavelength, and its reroutability status on w_k should be updated. This takes $O(N)$ time and in the worst case $O(N)$ lightpaths are rerouted. Therefore, the total running time is $O(N^2)$.

3. When the holding time of lightpath p expires:

   - All its links on wavelength w_k will be free links and their cost is updated accordingly. This takes $O(N)$.
   - For any lightpath q that uses the same fiber link, but a different wavelength, the reroutability status information is updated. The number of its free links on wavelength w_k is increased by one and if this equals to the hopcount of q, then q is reroutable to w_k. This will take $O(NW)$ time.

The total running time of the reroutability update procedure is $O(N^2+NW)$. We employ this procedure when solving the on-line SRWA problem using the MDLDP algorithms.

3.3 Simulation Study

We proceed to compare our 2-approximation algorithms (MSA and ESA) with the exact algorithm (WLA) in solving the MDLDP problem. In order to simulate a wide range of possibilities, we generate dynamic traffic, where requests arrive according to a Poissonian distribution (arrival rate r) with exponential holding times of mean 1. For each request, we record the results of our algorithms in comparison to WLA. The *approximation ratio* represents the ratio of the number of lightpaths traversed by an approximation algorithm to the number of lightpaths traversed by WLA. It is averaged for all accepted requests over 10 iterations, each 5000 requests. The source and destination nodes are randomly selected with all nodes having equal probability of being selected.

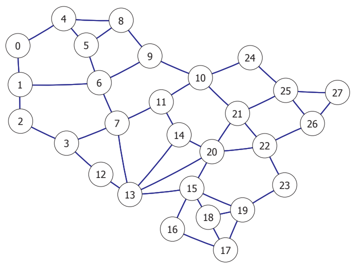

Figure 6: ARPANET network

We consider three networks: an ARPANET network (Figure 6), an Erdös-Rényi random network ($N = 50$, link density $\alpha = 0.2$), and a 7×7 lattice network, each with $W = 10$ wavelengths. In all our simulations, the approximation ratio attained by ESA never exceeded 1.00004. The approximation ratios of both ESA and MSA in comparison to the exact algorithm were much smaller than 2. Table 1 shows the average simulated approximation ratios of MSA, in terms of the number of lightpaths rerouted, when compared to WLA, which returns the exact number for a given request. Table 2 shows the approximation ratio of MSA for different wavelengths in the ARPANET network.

Table 1: Approximation ratios of the MSA algorithm in the three networks for different arrival rates (r) for $W = 10$.

ARPANET		RANDOM		LATTICE	
r	Approx. ratio	r	Approx. ratio	r	Approx. ratio
10	1.0105	20	1.0185	20	1.0208
15	1.0095	40	1.0209	30	1.0172
20	1.0055	60	1.0309	40	1.0147
25	1.0170	80	1.0326	50	1.0119
30	1.0128	110	1.0275	60	1.0093
35	1.0100	120	1.0169	70	1.0073

Table 2: Approximation ratios of the MSA algorithm for different wavelengths in the ARPANET network (W) for $r = 20$.

ARPANET	
W	Approx. ratio
2	1.0071
4	1.0066
6	1.0067
8	1.0084
10	1.0079
12	1.0050

In Sections 4 and 5, we use the aforementioned MDLDP algorithms to heuristically solve infinite and finite duration on-line SRWA, respectively. For each case, we compare the performances of MSA, ESA and WLA.

4 Infinite Duration On-line SRWA

In the infinite duration on-line SRWA problem, lightpaths stay indefinitely once they arrive. The off-line SRWA problem, where all the requests are known beforehand, can be described as a network flow problem as follows. We consider two cases: case 1 where both the primary and backup lightpaths have to use the same wavelength and case 2 where they can use different wavelengths.

Indices:

$i = 1, \dots, F$	ID of requests (F in total)
$w = 1, \dots, W$	ID of wavelengths
$\mathcal{L}$	Set of links
$\mathcal{N}$	Set of nodes
$\mathcal{N}(u)$	Set of nodes adjacent to node u

Variables (integers):

$\gamma_{i,w,u,v}$	is 1 (or -1 depending on the flow direction) if the primary or backup lightpaths of request i use wavelength w on link $(u, v) \in \mathcal{L}$; 0 otherwise.
$x_{i,w}$	*Case 1* (*same wavelength*): is 1 if request i is accepted and uses wavelength w; 0 otherwise. *Case 2* (*different wavelengths*): is 0 if neither the primary nor the backup lightpaths of request i are on wavelength w; 1 if either the primary or the backup lightpath of request i is on wavelength w; 2 if both the primary and the backup lightpaths of request i are on wavelength w.
y_i	is 1 if request i is accepted; 0 otherwise.

Objective:

The objective is to maximize the number of accepted requests.

$$\text{Maximize:} \quad \sum_{i=1}^{F} y_i$$

Constraints

Antisymmetry constraints: Since the graph is undirected, the flow is in both directions.

$$\gamma_{i,w,u,v} = -\gamma_{i,w,v,u} \qquad \forall (u, v) \in \mathcal{L}; 1 \le i \le F; 1 \le w \le W.$$

Conservation constraints: If a given node is not the source or destination of a given request, then any flow related to the request that enters the node has to leave the node.

$$\sum_{v \in \mathcal{N}(u)} \gamma_{i,w,u,v} = 0 \qquad \forall u \in \mathcal{N} \backslash \{s_i, t_i\}; 1 \le i \le F; 1 \le w \le W.$$

Capacity constraints: Only a single lightpath can use a given wavelength on a certain link.

$$\sum_{i=1}^{F} \gamma_{i,w,u,v} \le 1 \qquad \forall (u, v) \in \mathcal{L}; 1 \le w \le W.$$

Disjointedness constraints: The primary and the backup lightpaths of a request should be link-disjoint.

$$\sum_{w=1}^{W} \gamma_{i,w,u,v} \leq 1 \quad \forall (u,v) \in \mathcal{L}; 1 \leq i \leq F.$$

Equations

For the same wavelength: the number of lightpaths of a request on a given wavelength.

$$\sum_{v \in \mathcal{N}(s_i)} \gamma_{i,w,s_i,v} = 2 \cdot x_{i,w} \quad 1 \leq i \leq F; 1 \leq w \leq W.$$

$$\sum_{v \in \mathcal{N}(d_i)} \gamma_{i,w,v,d_i} = 2 \cdot x_{i,w} \quad 1 \leq i \leq F; 1 \leq w \leq W.$$

For the different wavelengths: the number of lightpaths of a request on a given wavelength.

$$\sum_{v \in \mathcal{N}(s_i)} \gamma_{i,w,s_i,v} = x_{i,w} \quad 1 \leq i \leq F; 1 \leq w \leq W.$$

$$\sum_{v \in \mathcal{N}(d_i)} \gamma_{i,w,v,d_i} = x_{i,w} \quad 1 \leq i \leq F; 1 \leq w \leq W.$$

A request is accepted if it has link-disjoint primary and backup lightpaths.

$$\sum_{w=1}^{W} x_{i,w} = 2y_i \quad 1 \leq i \leq F.$$

Solving the given ILP formulation for large networks and high number of requests is not feasible. Therefore, we use the algorithms of the MDLDP problem to solve the on-line SRWA problem sequentially. Clearly, this approach will not guarantee an optimal solution. However for small networks, we show that the results obtained by these algorithms are close to the optimal off-line solution (given by the ILP for case 2), which does not need rerouting. Tables 3 and 4 show comparisons, in terms of the number of rejected requests, of our algorithms (MSA and ESA), WLA and without rerouting (W/R) against the optimal ILP formulation for small random networks with link density α ($N = 10$ with 20 requests and $N = 12$ with 30 requests) and $W = 4$. We observe that rerouting performs better (though marginally, since the network is small and the number of requests are few) than without rerouting and our algorithms perform as good as (and at times better than) WLA.

Table 3: Number of rejected requests for $N = 10$ and 20 requests.

α	W/R	WLA	MSA	ESA	$Optimal$
0.2	15	14	14	14	12
0.3	13	12	12	11	7
0.4	8	6	6	6	3
0.5	7	6	6	7	3
0.6	6	5	5	4	1

Table 4: Number of rejected requests for $N = 12$ and 30 requests.

α	W/R	WLA	MSA	ESA	$Optimal$
0.2	23	22	22	22	21
0.3	25	25	25	25	24
0.4	12	11	11	10	6
0.5	5	3	3	3	2
0.6	4	3	3	3	0

5 Finite Duration on-line SRWA

Unlike infinite duration SRWA, finite duration SRWA requests arrive to and depart from the network over time. Thus, any two lightpaths can share requests as long as they do not overlap in time. Here also, we use the algorithms of MDLDP as heuristics to solve the finite duration on-line SRWA problem.

We use the same scenarios as in Section 3.3 for our simulations. Figures 7-12 show comparisons of the performance of our algorithms (MSA and ESA) with WLA in terms of the percentage of rejections. The given results are (a) for different number of requests with a constant arrival rate, and (b) for different arrival rates with constant number of requests. Figure 13 shows the percentage of rejection of MSA, ESA and WLA for different wavelengths. A comparison of these algorithms with the case of no rerouting (W/R) shows that rerouting of lightpaths decreases the percentage of rejections significantly. *In addition, we observe that both MSA and ESA perform similarly to WLA, which has much higher running time and memory requirements.* The need to have a small running time becomes more pronounced in an on-line setting, where the algorithm has to be invoked repeatedly whenever a request arrives.

6 Conclusions

In WDM optical networks, where lightpaths carry a tremendous amount of data, survivability is of paramount importance. In practice, lightpath requests arrive over time and a decision on whether to accept or deny a request should be made without any knowledge of the future requests. Therefore, it is necessary to have an on-line solution scheme with good performance to deal with survivable routing and wavelength assignment (SRWA). In this paper, we have

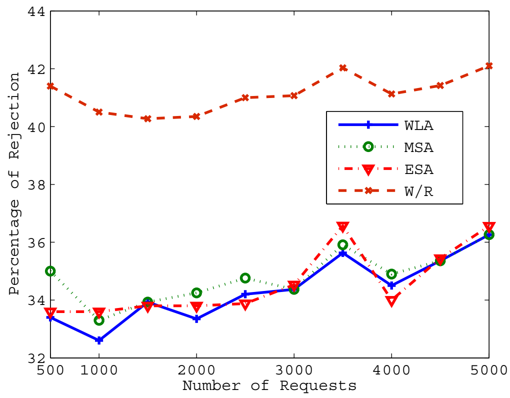

Figure 7: Rejection rates of MSA, ESA, WLA and without rerouting for different number of requests for the ARPANET network. ($W = 10$, $r = 40$)

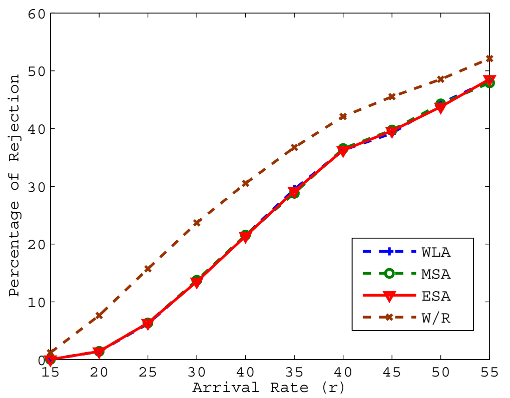

Figure 8: Rejection rates of MSA, ESA, WLA and without rerouting for different arrival rates of the ARPANET network. ($W = 10$ and 5000 requests)

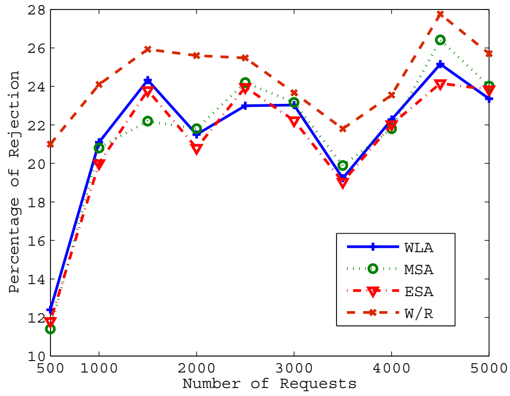

Figure 9: Rejection rate of MSA, ESA, WLA and without rerouting for different number of requests for the random network. ($N = 50$, $\alpha = 0.2$, $W = 10$, $r = 120$)

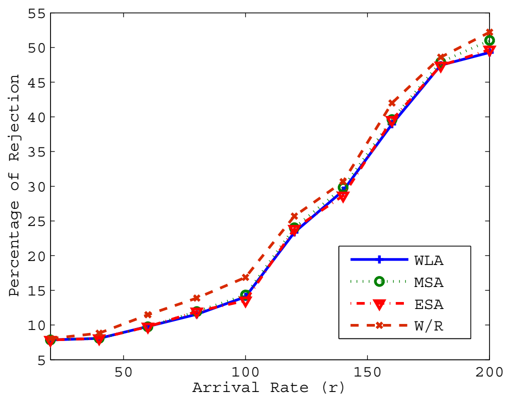

Figure 10: Rejection rates of MSA, ESA, WLA and without rerouting for different arrival rates for the random network. ($N = 50$, $\alpha = 0.2$, $W = 10$ and 5000 requests)

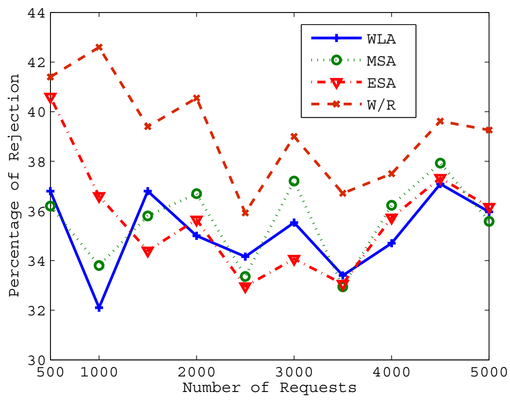

Figure 11: Rejection rates of MSA, ESA, WLA and without rerouting for different number of requests for the lattice network. ($N = 49$, $W = 10$, $r = 60$)

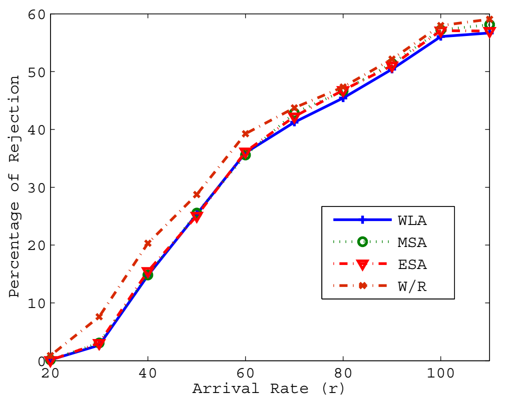

Figure 12: Rejection rates of MSA, ESA, WLA and without rerouting for different arrival rates in the lattice network. ($N = 49$, $W = 10$ and 5000 requests)

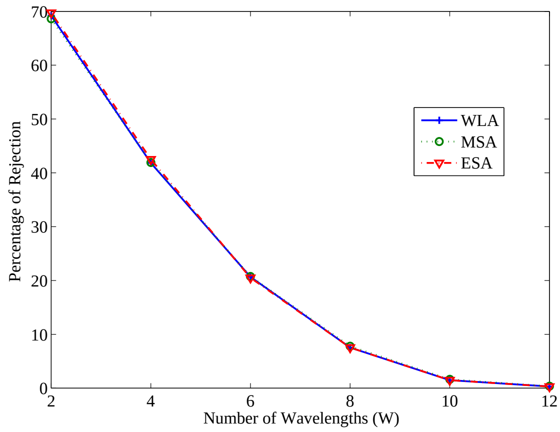

Figure 13: The percentage of rejection of MSA, ESA and WLA for different wavelengths on the ARPANET network. ($r = 20$ and 1000 requests)

studied on-line SRWA and have provided constant and logarithmic competitive ratios for special networks. For general networks, it is not possible to have algorithms with good competitive ratios. Since the competitive ratio reflects a worst-case performance, we considered lightpath rerouting, which generally improves the acceptance rate, but not the competitive ratio. To serve this purpose, we studied the Minimum Disruption Link-Disjoint Paths (MDLDP) problem, for which we provided two 2-approximation algorithms: MSA and ESA. We have shown through simulations that these algorithms perform close to the best-known exact algorithm for MDLDP, which incurs a very high time-complexity. We subsequently applied all considered MDLDP algorithms as heuristics for infinite and finite duration on-line SRWA. For infinite duration SRWA, these algorithms performed close to the optimal off-line solution (for which we provided an ILP formulation). For finite duration SRWA, we considered Poissonian distributed input sequences with exponential holding times. In these scenarios, our algorithms performed as good as (and at times even better than) the exact algorithm of the MDLDP problem, but in a much faster time. These findings suggest that our algorithms are more suitable for dealing with the (on-line) SRWA problem.

Acknowledgements

This work has partly been supported by the GigaPort project, which is led by SURFnet and funded by the Dutch Ministry of Economic Affairs under grant number BSIK03020 and partly by the Next Generation Infrastructures foundation (http://www.nginfra.nl).

A Appendix

A.1 Modified AAP Algorithm

Proposition 1 *For a graph $G = (\mathcal{N}, \mathcal{L})$, if all link capacities are at least $(\varepsilon \log n + 1 + \varepsilon)$, for some $\varepsilon > 0$, then there is a deterministic $O(2^{\frac{1}{\varepsilon}} \log n)-$competitive 2-MDP algorithm.*

Proof. We confine to identifying our modifications to the proof in [11].

Let $\mu = 2^{1+1/\varepsilon} n$. Hence, $\varepsilon \log \mu = \varepsilon \log n + 1 + \varepsilon$ and let $u_e \geq \varepsilon \log \mu$ denote the capacity of link e.

The modified AAP algorithm is as follows:

Define λ_e^j to be the fraction of u_e that already routed requests have used and define $c_e^j = u_e(\mu^{\lambda_e^j} - 1)$.

For request (s_j, t_j), route it on link-disjoint paths P_{j1} and P_{j2} that satisfy $\sum_{e \in P_{j1}} \frac{1}{u_e} c_j^e + \sum_{e \in P_{j2}} \frac{1}{u_e} c_j^e \leq n$.

We need to show that the relative load never exceeds 1. By contradiction, the relative load exceeds 1 when request j is accepted, i.e., $\lambda_e^j > 1 - \frac{1}{u_e} \geq 1 - \frac{1}{\varepsilon \log \mu}$. Thus, $\frac{c_e^j}{u_e} = \mu^{\lambda_e^j} - 1 > \mu^{1 - \frac{1}{\varepsilon \log \mu}} - 1 = \frac{\mu}{2^{1/\varepsilon}} - 1 > n$, which is a contradiction.

Given a total of k requests. Let A be the set of requests routed by the AAP algorithm. Then, we show that $3 \cdot 2^{1/\varepsilon} \log \mu \sum_{j \in A} n \geq \sum_{e \in \mathcal{L}} c_e^{k+1}$. The proof is by induction on k. For $k = 0$, the inequality is true. For $k > 0$, it is enough to show that, for any j, if we admit request j, we get $\sum_{e \in \mathcal{L}} \left(c_e^{j+1} - c_e^j\right) \leq 3 \cdot 2^{1/\varepsilon} n \log \mu$. Consider link e on either of the link-disjoint paths of the j^{th} request used by the AAP algorithm. We have, $c_e^{j+1} - c_e^j = u_e(\mu^{\lambda_e^j}(2^{(\log \mu)/u_e} - 1))$ and the exponent on 2 is at most $1/\varepsilon$. For $x \in [0, 1/\varepsilon]$, we have $2^x - 1 \leq 2^{1/\varepsilon} \cdot x$. Therefore,

$$\begin{aligned} c_e^{j+1} - c_e^j &\leq u_e \cdot \mu^{\lambda_e^j} \cdot 2^{1/\varepsilon} \cdot (\log \mu)/u_e \\ &= \mu^{\lambda_e^j} \cdot 2^{1/\varepsilon} \cdot \log \mu = 2^{1/\varepsilon} \cdot \log \mu \cdot \left[\frac{c_e^j}{u_e} + 1\right] \end{aligned}$$

Summing over all links, we get

$$\sum_{e \in \mathcal{L}} c_e^{j+1} - c_e^j \leq 2^{1/\varepsilon} \cdot \log \mu \cdot \sum_{e \in \mathcal{L}} \left[\frac{c_e^j}{u_e} + 1\right]$$

Only links on the link-disjoint paths of the j^{th} request are affected i.e.,

$$\sum_{e \in \mathcal{L}} c_e^{j+1} - c_e^j = \sum_{e \in P_{j1}} c_e^{j+1} - c_e^j + \sum_{e \in P_{j2}} c_e^{j+1} - c_e^j$$

We also know that $\sum_{e \in P_{j1}; P_{j2}} \frac{c_e^j}{u_e} \leq n$ and $\sum_{e \in P_{j1}; P_{j2}} 1 \leq 2n$. Hence, the desired bound is obtained. Finally, $\sum\limits_{e \in \mathcal{L}} c_e^{k+1}$ is an upper bound on the profit of the fractional optimum minus the profit of the on-line algorithm as shown in [11]. ■

A.2 Algorithm for short requests

Proposition 2 *Let $G = (\mathcal{N}, \mathcal{L})$ be an arbitrary graph of diameter d. Then, there is a deterministic on-line 2-MDP algorithm that is $\left(4 \cdot \max\left(d, \sqrt{L}\right)\right)$-competitive.*

Proof. Here also the proof is from [11] with a slight modification to incorporate link-disjoint paths. At the arrival of request i, the algorithm maintains a graph G_i, with $G_1 = G(\mathcal{N}, \mathcal{L})$. A request (s_i, t_i) is accepted and routed on shortest link-disjoint paths $\{P_{i1}, P_{i2}\}$ in G_i if $|P_{i1}| + |P_{i2}| \leq 2\sqrt{L}$. It then sets $G_{i+1} = G_i \backslash \{P_{i1}, P_{i2}\}$. Let k be the total number of requests accepted by the algorithm. Let $d' = \max(d, \sqrt{L})$. Consider any routing of requests consisting of q link-disjoint path pairs $\{Q_{11}, Q_{12}\}, \ldots, \{Q_{q1}, Q_{q2}\}$. Since each pair $\{Q_{j1}, Q_{j2}\}$ is link disjoint with the others and $|P_{i1}| + |P_{i2}| \leq 2d'$, at most $2d'$ of the path pairs $\{Q_{j1}, Q_{j2}\}$ intersect each pair $\{P_{i1}, P_{i2}\}$. Additionally, at most $\sqrt{L}$ of the link-disjoint path pairs $\{Q_{j1}, Q_{j2}\}$ fail to intersect any of the $\{P_{i1}, P_{i2}\}$ (since each pair (s_j, t_j) associated with $\{Q_{j1}, Q_{j2}\}$ that is rejected by the on-line algorithm and does not intersect with any $\{P_{i1}, P_{i2}\}$ has $|Q_{j1}| + |Q_{j2}| > 2\sqrt{L}$). Thus, we have $q \leq 2d'k + \sqrt{L} \leq 4d'k$. ■

References

[1] R. Adler and Y. Azar, "Beating the logarithmic lower bound: Randomized preemptive disjoint paths and call control algorithms," *Proc. of the 10th ACM-SIAM Symp. on Discrete Algorithms,* pp. 1-10, 1999.

[2] R.S. Anand and T. Erlebach, "On-line algorithms for edge-disjoint paths in trees of rings," *Proc. of the 5th Latin American Symp. on Theoretical Informatics*, pp. 584 - 597, 2002.

[3] R. Andersen, F. Chung, A. Sen, and G. Xue, "On disjoint path pairs with wavelength continuity constraint in WDM networks," *Proc. IEEE INFOCOM'04, pp. 524–535, 2004.*

[4] B. Awerbuch, Y. Azar, and S. Plotkin, "Throughput-competitive on-line Routing," *Proc. of the 34th IEEE FOCS*, pp. 32-40, 1993.

[5] B. Awerbuch, Y. Azar, A. Fiat, and A. Rosen, "Competitive non-preemptive call control," *Proc. of the 5th ACM-SIAM Symp. on Discrete Algorithms*, pp. 312-320, 1994.

[6] B. Awerbuch, R. Gawlick, F.T. Leighton, and Y. Rabani, "On-line admission control and circuit routing for high performance computing and communication," *Proc. of the Symp. on Foundations of Computer Science*, pp. 412–423, 1994.

[7] B. Awerbuch, Y. Azar, A. Fiat, S. Leonardi, and A. Rosen, "On-line competitive algorithms for call admission in optical networks," *Proc. of the 4th Annual European Symp. on Algorithms*, vol. 1136, pp. 431-444, 1996.

[8] Y. Bartal, A. Fiat, and S. Leonardi, "Lower bounds for on-line graph problems with application to on-line circuit and optical routing," *Proc. of the 28th Annual ACM Symp. on the Theory of Computing*, pp. 531-540, 1996.

[9] J.A. Garay, I.S. Gopal, S. Kutten, Y. Mansour, and M. Yung, "Efficient on-line call control algorithms," *Proc. of the Symp. on Theory of Computing Systems*, pp. 285-293, 1993.

[10] R.M. Karp, "On the computational complexity of combinatorial problems," *Networks*, vol. 5, no. 9, pp. 45-68, 1975.

[11] J. Kleinberg and É. Tardos, "Disjoint paths in densely embedded graphs," *Proc. of the 36th Annual Symp. on Foundations of Computer Science*, pp. 52-61, Oct. 1995.

[12] K.C. Lee and V.O.K. Li, "A wavelength rerouting algorithm in wide-area all-optical networks," *J. of Lightwave Tech.*, vol. 14, pp. 1218-1229, 1996.

[13] G. Mohan and C. Murthy, "A time optimal wavelength rerouting algorithm for dynamic traffic in WDM networks," *J. of Lightwave Tech.*, vol. 17, pp. 406-417, 1999.

[14] J. Suurballe, "Disjoint paths in a network," *Networks*, vol. 4, no. 2, pp. 125–145, 1974.

[15] Y. Wan and W. Liang, "Wavelength rerouting in survivable WDM networks," *Proc. of Networking*, Lecture Notes in Computer Science, vol. 3462, pp. 431-442, 2005.

[16] G. Xue, "Optimal lightpath routing and rerouting in WDM networks," *Proc. of GLOBECOM' 01*, vol. 4, pp. 2124 -2128, 2001.