

# Heuristic Algorithms for the Routing and Wavelength Assignment of Scheduled Lightpath Demands in Optical Networks

Nina Skorin-Kapov

**Abstract**—This paper addresses the problem of routing and wavelength assignment (RWA) of scheduled lightpath demands (SLDs) in wavelength routed optical networks with no wavelength converters. The objective is to minimize the number of wavelengths used. This problem has been shown to be NP-complete so heuristic algorithms have been developed to solve it suboptimally. Suggested is a tabu search algorithm along with two simple and fast greedy algorithms for the RWA SLD problem. We compare the proposed algorithms with an existing tabu search algorithm for the same problem and with lower bounds derived in this paper. Results indicate that the suggested algorithms not only yield solutions superior in quality to those obtained by the existing algorithm, but have drastically shorter execution times.

**Index Terms**—Routing and wavelength assignment, scheduled lightpath demands, optical networks, tabu search

## I. INTRODUCTION

WAVELENGTH routed WDM (*wavelength division multiplex*) optical networks can exploit the large bandwidth of optical fibers by dividing it among different wavelengths. These networks can also connect a large number of nodes using a small number of wavelengths by taking advantage of spatial wavelength reusability. Wavelength routed WDM networks are equipped with configurable WDM nodes which enable us to set up and tear down all-optical connections, called *lightpaths*, between pairs of nodes. These all-optical connections can traverse multiple physical links in the network. Information sent via a lightpath does not require any opto-electronic conversion at intermediate nodes.

Establishing a set of lightpaths creates a virtual topology on top of the physical topology. The physical topology represents the physical interconnection of WDM nodes by actual fiber links in the WDM optical network. The links in the virtual topology represent all-optical connections or lightpaths established between pairs of nodes. Demands to set up lightpaths between certain nodes can be static, scheduled or dynamic. In the case of *static lightpath demands*, a desired static virtual topology (i.e. the set of lightpaths we wish to establish) is known *a priori*. Such a virtual topology is set up ‘permanently’ which implies that even when a certain connection is not being used its resources remain reserved. When we refer to *scheduled lightpath demands* we refer to connection requests for which we know the set-up and tear-down times *a priori*. In other words, we know in advance when a connection

will be needed. For *dynamic lightpath demands*, lightpath requests arrive unexpectedly with random holding times. These lightpaths are set up dynamically at request arrival time and are released when the connection is terminated.

To set up a lightpath, nodes on its corresponding physical path must be configured to do so. If the network lacks wavelength converters, the lightpath must use the same wavelength along its entire physical path. This is called the *wavelength continuity constraint*. Two lightpaths that share a common physical link cannot be assigned the same wavelength. This is called the *wavelength clash constraint*. The source and destination nodes of the lightpath must also have available transmitters and receivers respectively in order for the lightpath to be successfully set up.

Determining routes for a set of lightpath demands and assigning wavelengths to these routes subject to a subset of the above mentioned constraints is known as the Routing and Wavelength Assignment (RWA) problem. This problem has been proven to be NP-complete [1] and several heuristic algorithms have been developed to solve it suboptimally [2] [3] [4] [5]. Several variations of the RWA problem have been studied [6] [7] [8] such as the routing and wavelength assignment of static, scheduled or dynamic lightpath demands with a limited or unlimited number of wavelengths in networks with wavelength converters at each node, at a subset of nodes, or in networks with no wavelength converters. The objective is often to minimize the number of wavelengths used or to maximize the number of lightpaths set up subject to a limited number of wavelengths.

In this paper we consider the Routing and Wavelength Assignment of Scheduled Lightpath Demands (RWA SLD) in networks with no wavelength converters. We assume no limit on the number of transmitters and receivers at each node and the number of available wavelengths on each link. The objective is to minimize the number of wavelengths needed to successfully establish a desired set of scheduled lightpath demands. Considering scheduled lightpath demands seems relevant due to the periodic nature of traffic [9]. We know that traffic between some nodes (e.g. office headquarters) is heavier during office hours than in the middle of the night, and *vice versa* for the other nodes (e.g. networked data bases). We could utilize this information by setting up multiple lightpaths between nodes at times when their traffic is heavy and tearing down some or all of these lightpaths at times when their traffic is low. By tearing down lightpaths between nodes at times of low traffic, their resources are freed and can thus be used to

N. Skorin-Kapov is with the Department of Telecommunications, Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia. (e-mail: nina.skorin-kapov@fer.hr).

establish alternative connections. If we have lightpath demands which do not overlap in time and if we take this information into consideration when performing routing and wavelength assignment, we can route both demands on the same path using the same wavelength without them clashing. This can significantly reduce the amount of network resources required to successfully route a set of lightpath demands.

Since the Routing and Wavelength Assignment of Scheduled Lightpath Demands is solved *a priori* using given scheduling information, the lightpaths can be set up and torn down quickly at the specified times. This is an advantage over the routing and wavelength assignment of dynamic lightpath demands where RWA is performed dynamically as lightpath requests arrive, resulting in longer set-up delays. The RWA SLD problem has not been studied as widely as the static and dynamic cases. A branch and bound algorithm along with a tabu search heuristic algorithm are given in [9]. In this work we suggest a faster and more efficient tabu search algorithm for the RWA SLD problem along with two very fast and simple greedy algorithms based on edge and time disjoint paths. Furthermore, we derive an efficient lower bound on the number of wavelengths needed for the RWA SLD problem.

The rest of the paper is organized as follows. In Section II, we formally define the RWA SLD problem. Related work is briefly described in Section III. In Section IV we suggest a tabu search algorithm for the RWA problem, followed by two simple yet very efficient greedy algorithms in Section V. In VI we discuss lower bounds. Numerical results and concluding remarks are given in Sections VII and VIII, respectively.

## II. PROBLEM DEFINITION

The physical optical network is modelled as a graph  $G = (V, E)$ , where  $V$  is the set of nodes and  $E$  is the set of edges. Edges are assumed to be bidirectional (each representing a pair of optical fibers, i.e. one fiber per direction) and have assigned weights representing their length or cost. Given is a set of scheduled lightpath demands  $\tau = \{SLD_1, \dots, SLD_M\}$ . Each scheduled lightpath demand  $SLD_i$ , where  $i = 1, \dots, M$ , is represented by a tuple  $(s_i, d_i, n_i, \alpha_i, \omega_i)$  as suggested in [9]. Here  $s_i, d_i \in V$ , are the source and destination nodes of  $SLD_i$ ,  $n_i$  is the the number of requested lightpaths between these nodes, and  $\alpha_i$  and  $\omega_i$  are the set-up and tear-down times respectively. The Routing and Wavelength Assignment problem consists of finding a set of paths  $P = \{P(SLD_1), \dots, P(SLD_M)\}$  in  $G$ , each corresponding to one scheduled lightpath demand, and assigning a set of wavelengths to each of these paths. As in [9], we assume that all the lightpaths of a particular SLD must be routed on the same path and must therefore be assigned different wavelengths. We will refer to this as the *group lightpath constraint*. As a result, each path  $P(SLD_i)$ , where  $i = 1, \dots, M$ , must be assigned a set of  $n_i$  wavelengths, one for each individual lightpath of  $SLD_i$ . The set of wavelengths assigned to paths  $P(SLD_i)$  and  $P(SLD_j)$ , where  $i \neq j$  and  $i, j = 1, \dots, M$ , must be disjoint if these paths share a common edge *and* if  $SLD_i$  and  $SLD_j$  overlap in time. The objective is to minimize the number of wavelengths assigned and required to successfully route and

assign wavelengths to all the scheduled lightpath demands in  $\tau$ .

## III. RELATED WORK

In [9], the authors solve the Routing and Wavelength Assignment problem of Scheduled Lightpath Demands by decoupling it into two separate subproblems: routing and wavelength assignment. They suggest a branch and bound algorithm for the routing subproblem which provides optimal solutions but has an exponential complexity. To solve the routing subproblem for larger problems, the authors propose a tabu search algorithm which obtains suboptimal solutions. Two different optimization criteria are considered giving rise to two versions of the tabu search algorithm:  $TS_{ch}$  and  $TS_{cg}$ . The  $TS_{ch}$  algorithm minimizes the number of WDM channels<sup>1</sup> which is particularly important in opaque WDM networks and will not be discussed here. The  $TS_{cg}$  algorithm minimizes congestion, i.e. the number of lightpaths on the most heavily loaded link. This optimization criterion is important in networks with a limited number of wavelengths since congestion is essentially a lower bound on the number of wavelengths required. Wavelength assignment is performed subsequently using a greedy graph coloring algorithm (referred to as *GGC*) suggested in [10]. The objective of the *GGC* algorithm is to minimize the number of wavelengths used. The quality of the solutions for the RWA SLD problem obtained by  $TS_{cg}/GC$  are measured in terms of the number of wavelengths needed. A complete description is provided in [9].

Fault tolerant routing and wavelength assignment of scheduled lightpath demands was studied in [11] and [12]. In [11], the authors formulate the problem of fault tolerant RWA SLD with the objective being to minimize the number of WDM channels. They propose a Simulated Annealing algorithm using channel reuse and back-up multiplexing. In [12], fault tolerant RWA SLD under single component failure is considered. The authors develop ILP formulations for the problem with dedicated and shared protection. Two objectives are considered: minimizing the capacity needed to guarantee protection for all connection requests and maximizing the number of requests accepted subject to a limited capacity.

## IV. AN ALTERNATIVE TABU SEARCH ALGORITHM FOR THE ROUTING SUBPROBLEM

### A. Tabu Search

Tabu search is an iterative meta-heuristic which guides simpler heuristics in such a way that they explore various areas of the solution space and prevents them from remaining in local optima. In every iterative step of the tabu search method, we begin with some current solution and explore its neighboring solutions. Neighboring solutions with respect to the current one are all those obtained by applying some elementary transformation to the current solution. The best neighboring solution according to some evaluation function is selected as the new current solution in the next iteration.

<sup>1</sup>WDM channels refer to the use of a particular wavelength on a directed physical link and are specified by the wavelength used and the head and tail nodes of the directed link.

After executing a desired number of iterations, the best found solution overall, called the incumbent solution, is deemed the final result.

To prevent the search technique from getting stuck in a local optimum or cycling between already seen solutions, a memory structure called a tabu list is introduced. The tabu list ‘memorizes’ a certain number of previously visited solutions which are then forbidden for as long as they remain in the list. The tabu list is updated circularly after every iteration by adding the current solution to the list and removing the oldest element if the list is full. The length of the tabu list can vary depending on different problems and is often determined experimentally. The key to developing a good tabu search algorithm is to define a good initial solution, neighborhood structure and evaluation function. Sometimes the neighborhood of a solution can be very large so various neighborhood reduction techniques are applied. As a result, only a subset of the neighboring solutions are evaluated. A detailed description of the tabu search method can be found in [13].

### B. The proposed tabu search algorithm: $TS_{cn}$

The tabu search algorithm  $TS_{cg}$  for the routing subproblem suggested by the authors of [9], although faster than their branch and bound algorithm, still has a fairly long execution time if run for a large enough neighborhood size and the number of iterations needed to obtain solutions of good quality. This is due to their randomized neighborhood search technique. Namely, the  $TS_{cg}$  algorithm begins by computing the  $K$ -shortest paths between the source and destination nodes of each of the  $M$  SLDs in  $\tau$ . Potential routing solutions are represented by a vector of  $M$  integers (initially all set to 1) ranging from 1 to  $K$ , each representing the path used by a particular SLD. Neighboring solutions are all those in which one and only one SLD is routed on a different route. A neighborhood defined as such can be very large so a desired number of SLDs are chosen at random, randomly rerouted and evaluated according to their corresponding congestion. To obtain good solutions, a large number of iterations and a fairly large number of neighbors need to be evaluated.

Since our final objective for the RWA SLD problem is to minimize the number of wavelengths used, we will compare our results with that of algorithm  $TS_{cg}/GGC$ . The evaluation function used by  $TS_{cg}$  (i.e. minimization of congestion) does not necessarily lower the number of wavelengths needed even though this is the final goal of the  $TS_{cg}/GGC$  algorithm. Recall that by minimizing congestion, which is the number of lightpaths on the most loaded link, we are essentially trying to minimize the lower bound on the number of wavelengths needed. Minimizing the *lower* bound does not necessarily mean that the number of wavelengths needed will be lower. On the other hand, if we tried to minimize the *upper* bound on the number of wavelengths needed, this guarantees that there exists a wavelength assignment with at most the upper bound number of wavelengths.

The tabu search algorithm for the routing subproblem suggested in this work attempts to improve the drawbacks

mentioned above. A new evaluation function is suggested along with a directed neighborhood search technique which drastically reduces the neighborhood size. As a result, this algorithm, in combination with the same graph coloring algorithm ( $GGC$ ) used in [9] for wavelength assignment, performs faster and obtains solutions of better or equal quality than those obtained by  $TS_{cg}/GGC$ . We will refer to the proposed algorithm as  $TS_{cn}$  where  $cn$  stands for *chromatic number*. The relevance of this name will be described below.

1) *Preliminaries*: Recall that the objective of the wavelength assignment algorithm as well as our final objective for the RWA SLD problem is to minimize the total number of wavelengths used. Since we perform wavelength assignment using the  $GGC$  algorithm *after* solving the routing subproblem, it seems that having a routing algorithm aware of the objective and behavior of the wavelength assignment algorithm could help to obtain better solutions for the RWA SLD problem. In other words, the optimization criteria of the routing algorithm should be such that it gives routing schemes on which wavelength assignment can be performed using a smaller number of wavelengths. To formulate such an optimization criteria we must first analyze the behavior of the wavelength assignment algorithm which is here essentially a graph coloring algorithm.

Namely, the problem of wavelength assignment can be reduced to the *graph coloring problem* which consists of assigning colors to the nodes of a graph such that no two neighboring nodes are assigned the same color. The objective is to minimize the total number of colors used. This classical graph theory problem has been proven to be NP-complete so several heuristic algorithms have been developed [14]. One such algorithm is the  $GGC$  algorithm proposed in [10]. The routing solution obtained by solving the routing subproblem is used as input for the  $GGC$  algorithm in the following manner. A conflict graph corresponding to the routing solution is created where each established lightpath is represented by one node in the conflict graph<sup>2</sup> and there is an edge between two nodes if their respective paths share a common physical link in  $G$  and overlap in time. This means that the lightpaths corresponding to neighboring nodes in the conflict graph cannot be assigned the same wavelength. The graph coloring algorithm  $GGC$  is executed on this conflict graph where each color represents a different wavelength.

The minimum number of colors needed to color a graph is called the *chromatic number*. In 1941, Brooks [15] showed the upper bound on the chromatic number to be  $\Delta(G) + 1$ , where  $\Delta(G)$  is the maximum degree in  $G$ . This bound was used for a long time. A more recent result by L. Stacho in 2001 [16] gives a tighter upper bound. The author showed that the chromatic number is always less than or equal to  $\Delta_2(G) + 1$  where  $\Delta_2(G)$  is the largest degree of any node  $v$  in  $G$ , such that  $v$  is adjacent to a node whose degree is at least as big as its own.

Let us consider an example. In Fig. 1, two simple 4 node

<sup>2</sup>Note that one node in the conflict graph represents one particular *lightpath* of an SLD and not the SLD itself. In other words, each scheduled lightpath demand  $SLD_i$  is represented by  $n_i$  nodes in the conflict graph which are all adjacent to each other.

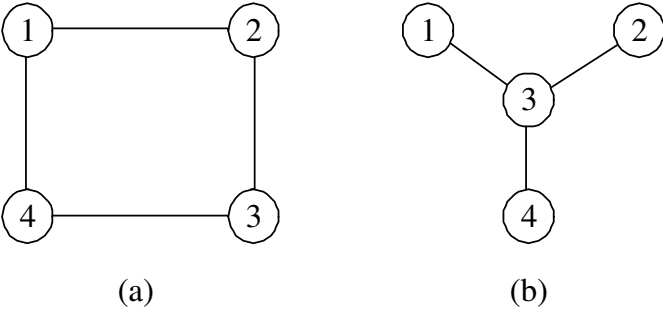


Fig. 1. Two simple 4 node networks.

networks are shown. For the network shown in Fig 1.(a), both Brooks' and Stacho's upper bounds give a value of 3. However, for the network shown in Fig. 1.(b), using Brooks' upper bound on the chromatic number, we get a value of  $\Delta(G) + 1 = 3 + 1 = 4$ , while using Stacho's we get a value of  $\Delta_2(G) + 1 = 1 + 1 = 2$ . We can easily see that nodes 1, 2, and 4 can be colored with one color and node 3 with a second color.

According to Stacho's upper bound, it is evident that graphs with smaller values of  $\Delta_2(G)$  give smaller upper bounds for the chromatic number. Note that a routing solution  $X$  obtained by solving the routing subproblem corresponds to exactly one conflict graph  $CG(X)$  on which we solve the graph coloring problem. If we take into consideration upon constructing routing solution  $X$  that we wish to minimize its corresponding value for  $\Delta_2(CG(X))$ , we may attain a routing scheme whose corresponding conflict graph will need fewer colors to perform graph coloring successfully. This also means that we need fewer wavelengths to perform a successful wavelength assignment. Accordingly, the optimization criteria or evaluation function used by the  $TS_{cn}$  algorithm to evaluate a routing solution  $X$  is the minimization of the upper bound on the chromatic number of its corresponding conflict graph (i.e.  $\min (\Delta_2(CG(X)) + 1)$ ).

2) *The  $TS_{cn}$  algorithm:* A description of the tabu search algorithm  $TS_{cn}$  proposed for the routing subproblem of scheduled lightpath demands follows. As in [9], we first compute the  $K$ -shortest paths between each source-destination pair of each SLD using Eppstein's algorithm [17].  $K$  can be set to various values. If we set  $K$  to a larger value, the solution obtained will probably need less wavelengths but the physical paths used to route the SLDs will probably be longer. This may present a problem if delay is an issue. On the other hand, if  $K$  is set to a smaller value, the physical paths will be restricted to only a few of the shortest paths. As a result, the number of wavelengths needed to successfully route the SLDs will most likely be larger.

Recall that we have given a graph  $G = (V, E)$  and a set of  $M$  Scheduled Lightpath Demands (SLDs) each represented by a tuple  $(s_i, d_i, n_i, \alpha_i, \omega_i)$ , where  $s_i$  is the source,  $d_i$  is the destination,  $n_i$  is the number of requested lightpaths,  $\alpha_i$  is the set-up time, and  $\omega_i$  is the tear-down time of the SLD. For simplification purposes, the authors of [9] assume that the *group lightpath constraint* applies, i.e. all the lightpaths

of a particular SLD are routed on the same path. The same will be assumed here for easier comparison of the mentioned algorithms. A potential routing solution  $X$  is represented by a vector of  $M$  integers,  $X = (x_1, \dots, x_M)$ , where  $x_i \in \{1, \dots, K\}$ ,  $i = 1, \dots, M$ , represents the path used by  $SLD_i$ . If the integer representing the path of a specific SLD is set to 1, that means that that particular SLD is routed on the shortest path from its source to destination. If it is set to 2, then that SLD is routed on the second shortest path from source to destination, and so on up to the  $K^{th}$  shortest path. The  $TS_{cn}$  algorithm initially routes all the SLDs in  $\tau$  on their shortest paths in  $G$ .

Neighboring solutions with respect to the current one are all those where one and only one SLD is routed on a different route. Instead of selecting a large number of neighbors at random as in [9] and evaluating them, we suggest a more directed neighborhood reduction technique. This technique drastically reduces the size of the neighborhood and yet helps obtain solutions of good quality. First we construct the conflict graph  $CG(X)$  of the current solution  $X$  and then find the set of nodes  $L(X)$  which determine  $\Delta_2(CG(X))$ . That is, we find the one or more nodes which have the largest degree in  $CG(X)$ , subject to the fact that they are adjacent to a node whose degree is at least as big as their own. Recall that the nodes in the conflict graph represent individual lightpaths and not SLDs. Since we are routing all the lightpaths of a particular SLD on the same path (i.e. they all have the same degree<sup>3</sup>), either all or none of the lightpaths of a particular SLD are in  $L(X)$ . As a result, we can easily reduce the set of lightpaths  $L(X)$  to their corresponding set of SLDs  $L_{SLD}(X)$  where  $|L_{SLD}(X)| \leq |L(X)|$ . The number of SLDs in  $L_{SLD}(X)$  is usually fairly small. Instead of evaluating a huge number of neighboring solutions, we evaluate only  $|L_{SLD}(X)|$  neighbors. The  $|L_{SLD}(X)|$  neighbors are obtained by randomly rerouting each SLD in  $L_{SLD}(X)$ .

To determine the best neighboring solution which will pass into the next iteration, we create a conflict graph for each neighboring solution and find its corresponding upper bound on the chromatic number. In other words, we find  $\Delta_2(CG(X)) + 1$  for each neighbor  $X$ . The neighboring solution with the lowest upper bound is passed into the next iteration and becomes the new current solution. If this solution is better than the incumbent solution, the incumbent solution is updated. Such an evaluation function is the motivation for the neighborhood reduction technique. Namely, if we reroute the SLDs which determine  $\Delta_2(CG(X))$  (i.e. the SLDs in  $L_{SLD}(X)$ ) instead of rerouting SLDs at random, there is a greater chance that we might improve the upper bound and pass a better solution into the next iteration. Of course, this does not guarantee that a better solution cannot be found by rerouting a series of SLDs not included in set  $L_{SLD}(X)$ . However, this is an approximation algorithm in which a trade

<sup>3</sup>For example, if  $SLD_1$  with  $n_1 = 3$  lightpaths is adjacent to  $SLD_2$  and  $SLD_3$  with  $n_2 = 7$  and  $n_3 = 5$  lightpaths respectively, all three nodes representing lightpaths of  $SLD_1$  have a degree of  $n_2 + n_3 + (n_1 - 1) = 7 + 5 + (3 - 1) = 14$  in the conflict graph.  $n_1 - 1$  is added because each lightpath of  $SLD_1$  is adjacent to all the other lightpaths of  $SLD_1$  except itself.

off between execution time and potential solution quality must be made.

A few extra features of the algorithm are as follows. For diversification purposes, if there is no improvement after a certain number of iterations, we take a random number of SLDs and randomly reroute them. If at some point no neighbor can be rerouted (basically, they have all been rerouted and are on the tabu list), we reroute all the SLDs with the maximum degree in the conflict graph, (i.e.  $\Delta(CG(X))$ , not  $\Delta_2(CG(X))$ ). If this solution is not on the tabu list, it becomes the new current solution. If it is on the tabu list, we take a random number of SLDs and randomly reroute them. In addition to the tabu list which records the last change made in the form of  $(SLD_i, P(SLD_i))$ , where  $SLD_i$  is a number ranging from 1 to  $M$  and  $P(SLD_i)$  is a number ranging from 1 to  $K$ , we separately record the SLD which was last changed. Rerouting this SLD on *any* path is forbidden in the following iteration.

The pseudocode of  $TS_{cn}$  follows.

**Input and initialization:**

$G = (V, E)$ ;  
 $\tau = \{SLD_1, \dots, SLD_M\}$ , where  $SLD_i = (s_i, d_i, n_i, \alpha_i, \omega_i), i = 1, \dots, M$ ; //the set of SLDs  
 $K$ ; //the number of  $K$ -shortest paths  
 //initial routing solution with all paths set to 1  
 $X_0 = (x_1^0, \dots, x_M^0), x_i^0 := 1, i = 1, \dots, M$ ;  
 Find  $\Delta_2(CG(X_0))$  and the corresponding SLDs  $L_{SLD}(X_0) = \{SLD_{r_1}, \dots, SLD_{r_s}\}, r_i \in \{1, \dots, M\}, i = 1, \dots, s$ ;  
 $X := X_0$ ; //incumbent solution  
 $\Delta_2 := \Delta_2(CG(X_0))$ ; //fitness of incumbent solution  
 $Tabulist := \{\}, i := 0, itWOImprovement := 0$ ;

**Begin:**

//iterations  
**while**  $i <$  desired number of iterations **do**  
 $X_{it} := \{\}, \Delta_2(CG(X_{it})) := \infty, L_{SLD}(X_{it}) := \{\}$ ;  
**for**  $j$  in  $1, \dots, |L_{SLD}(X_i)|$  **do**  
 $x_{r_j}^i :=$  random number in  $\{1, \dots, K\} \setminus x_{r_j}^i$  except for that forbidden by tabu list;  
 $X_i' := (x_1^i, \dots, x_{r_j-1}^i, x_{r_j}^i, x_{r_j+1}^i, \dots, x_M^i)$ ;  
 Find  $\Delta_2(CG(X_i'))$  and  $L_{SLD}(X_i')$ ;  
**if**  $\Delta_2(CG(X_i')) < \Delta_2(CG(X_{it}))$  **then**  
 $X_{it} := X_i', \Delta_2(CG(X_{it})) := \Delta_2(CG(X_i')), L_{SLD}(X_{it}) := L_{SLD}(X_i')$ ;  
**end if**  
**end for**  
**if**  $\Delta_2(CG(X_{it})) == \infty$  **then**  
 //all neighbors are on the tabu list  
 Find all nodes with max degree in conflict graph of solution  $X_i$  (i.e.  $\Delta(CG(X_i))$ ) and randomly reroute them. If this is on tabu list, choose a random number of SLDs and randomly reroute them;  
**else**  
 $X_i := X_{it}, \Delta_2(CG(X_i)) := \Delta_2(CG(X_{it})), L_{SLD}(X_i) := L_{SLD}(X_{it})$ ;  
**end if**  
 Update tabu list;  
**if**  $\Delta_2(CG(X_i)) < \Delta_2$  **then**  
 $X := X_i, \Delta_2 := \Delta_2(CG(X_i))$ ;  
**else**  
 $itWOImprovement := itWOImprovement + 1$ ;  
**end if**  
**if**  $itWOImprovement \geq$  allowed no. of iterations without improvement **then**  
 Select a random number of SLDs and randomly reroute them;  
**end if**  
 $i := i + 1$ ;  
**end while**  
**End**

After solving the routing subproblem with the  $TS_{cn}$  algorithm, we use the  $GGC$  graph coloring algorithm [10] for wavelength assignment. The computational results are presented in Section VII.

### C. Complexity Analysis

For better insight, we examine the computational complexity of the  $TS_{cg}/GC$  and  $TS_{cn}/GC$  algorithms. Both tabu search algorithms use Eppstein's algorithm for computing the  $k$ -shortest paths, run the desired number of iterations of their respective tabu search algorithms, and then use the  $GGC$  algorithm for wavelength assignment. As a result, the computational complexity of the  $TS_{cg}/GC$  and  $TS_{cn}/GC$  algorithms differ only with respect to the operations performed in each iteration of the tabu search algorithms. Eppstein's algorithm for the  $K$ -shortest paths with time complexity  $O(|E| + |V| \log |V| + K)$  is run for each of the  $M$  SLDs. The  $GGC$  algorithm is an improvement algorithm which is run for a desired number of iterations where each iteration has a worst case time complexity of  $O(|V|^2)$ . The complexity analysis of the iterations of the respective tabu search algorithms follows.

In each iteration of the  $TS_{cg}/GC$  algorithm, each neighboring solution is evaluated by finding the highest congestion on any of the  $|E|$  links. The congestion on edge  $e \in E$  is computed by sorting the set-up and tear-down times of the SLDs routed over  $e$  and then finding the time interval in which the maximum number of lightpaths are active. Sorting takes  $O(M \log M)$  time. Finding the highest congestion takes  $O(M)$  time since the number of time intervals must be  $\leq 2M$ . It follows that finding the highest congestion over all edges takes  $O(|E|M(\log M + 1))$  time. Time complexity analysis for some of these steps was developed in [18] for their Simulated Annealing algorithm for fault-tolerant RWA SLD. If  $Nbr$  is the neighborhood size, each of the  $Nbr$  neighbors is evaluated in  $O(|E|M(\log M + 1))$  time in each iteration.

The  $TS_{cn}/GC$  algorithm, on the other hand, evaluates each neighboring solution  $X$  by constructing a conflict graph  $CG(X)$  and then finding the upper bound on the chromatic number,  $\Delta_2(CG(X)) + 1$ , of the conflict graph. The conflict graph can be constructed in  $O(M^2)$  time.  $\Delta_2(CG(X))$  and the corresponding neighborhood  $L_{SLD}(X)$ , can be found in  $O(M^2)$ . It follows that the complexity of evaluating a neighboring solution is  $O(M^2)$ . Since the neighborhood is adaptive, the size of the neighborhood is not constant. The upper bound on the number of neighbors is  $M$ . This occurs only if the conflict graph  $CG(X)$  is a complete graph. However, empirical testing indicates that the neighborhood size is often drastically smaller than  $M$  (see Section VII, Table V), even when  $M$  is large. The neighborhood size could also be additionally upper bounded by a constant, say value  $Nbr$  used by  $TS_{cg}/GC$ , so that the number of neighbors evaluated in each iteration is  $\min\{L_{SLD}(X), Nbr\}$ , where each evaluation is performed in  $O(M^2)$  time. The numerical results in Section VII indicate that  $TS_{cg}/GC$  is significantly slower than  $TS_{cn}/GC$  for the cases tested.

V. EDGE AND TIME DISJOINT PATH ALGORITHMS:  
 $DP\_RWA\_SLD$  AND  $DP\_RWA\_SLD^*$

A.  $DP\_RWA\_SLD$

In order to solve the routing and wavelength assignment problem of a set of scheduled lightpath demands, we propose an algorithm motivated by a routing and wavelength assignment algorithm for *static* lightpath demands suggested in [4]. This algorithm, called *Greedy\_EDP\_RWA*, creates a partition  $\tau_1, \dots, \tau_k$  of a set of static lightpath demands<sup>4</sup>  $\tau = \{(s_i, d_i), \dots, (s_M, d_M)\}$ , where  $s_i, d_i \in V, i = 1, \dots, M$ . Each element of the partition is composed of a subset of lightpath demands which can be routed on mutually edge disjoint paths in  $G$  and hence can be assigned the same wavelength. The length of each path is upper bounded by a value  $h$  set in [4] to  $\max(\text{diam}(G), \sqrt{|E|})$ . The justification for setting  $h$  to this value is given in [19]. The number of distinct wavelengths needed to successfully perform RWA corresponds to the number of elements in the partition.

To solve the RWA SLD problem, we propose a fast algorithm using some of the ideas introduced above. Routing and wavelength assignment are solved simultaneously based on the idea of finding a partition  $\tau_1, \dots, \tau_k$  of the set of *scheduled* lightpath demands  $\tau$  where each element  $\tau_i, i \in 1, \dots, k$ , of the partition is composed of SLDs routed over ‘disjoint’ paths. Here, ‘disjoint’ paths include not only *edge* disjoint paths as in *Greedy\_EDP\_RWA*, but *time* disjoint paths as well. Two paths that are disjoint in time may be routed using the same physical edges. The lengths of the paths are upper bounded by a value  $h$ . We will refer to this algorithm as  $DP\_RWA\_SLD$ , where  $DP$  stands for *Disjoint Paths*.

The  $DP\_RWA\_SLD$  algorithm first sorts the SLDs in  $\tau$  in decreasing order of the number of lightpaths each SLD requests. The reason for this will be discussed later. A partition of  $\tau$  is then created in the following manner. The first SLD from the sorted set of demands is routed on its shortest path in  $G$ . This SLD and its corresponding path are placed in  $\tau_1$  and removed from  $\tau$ . Subsequent attempts are made to route the remaining requests in  $\tau$  as follows. For each new SLD considered, the edges of the paths of those SLDs already in  $\tau_1$  with which the new SLD overlaps in time are deleted from  $G$ . The resulting graph is referred to as  $G'$ . The new SLD is now routed on its shortest path in  $G'$ . If this routing is successful (i.e. there exists such a path in  $G'$  whose length is  $\leq h$ ), the new SLD is added to  $\tau_1$  and removed from  $\tau$ . Otherwise, it remains in  $\tau$ . After attempting to route all the SLDs in  $\tau$ , we are left with a set of demands routed on mutually disjoint paths in  $\tau_1$  and a set of unrouted demands in  $\tau$ . This entire procedure is iteratively repeated on the SLDs remaining in  $\tau$  to create the other elements of the partition,  $\tau_2, \dots, \tau_k$ , until all the demands in  $\tau$  are successfully routed.

Since we are creating a partition of SLDs (not individual lightpaths) we cannot assume that only one wavelength is needed for each element of the partition. Since all the SLDs in  $\tau_i$  are mutually disjoint, their respective lightpaths can be

<sup>4</sup>Here, each static lightpath demand represents a single lightpath which is to be set up permanently. As a result, each demand is defined only by its source and destination nodes.

TABLE I  
 EXAMPLE

$SLD_i$	$s_i$	$d_i$	$n_i$	$\alpha_i$	$\omega_i$
$SLD_1$	4	3	5	1:00	6:00
$SLD_2$	4	2	10	2:00	6:00
$SLD_3$	4	1	9	2:00	7:00
$SLD_4$	1	3	7	1:00	2:00

assigned the same set of wavelengths. On the other hand, each individual lightpath of a particular SLD must be assigned a different wavelength since they are all routed on the same path. It follows that the number of wavelengths  $W_i$  which must be assigned to  $\tau_i$  is the maximum number of lightpaths any SLD included in  $\tau_i$  requests. Wavelength assignment is performed in the following manner. For each SLD in  $\tau_1$ , its corresponding lightpaths are assigned wavelengths 1, 2,  $\dots$  up to  $W_1$  if necessary. The lightpaths routed in  $\tau_2$  are assigned wavelengths  $\{(W_1 + 1), \dots, (W_1 + W_2)\}$ , the lightpaths in  $\tau_3$  are assigned wavelengths  $\{((W_1 + W_2) + 1), \dots, ((W_1 + W_2) + W_3)\}$ , and so on. Generally speaking, each element  $\tau_i, i = 1, \dots, k$  is assigned wavelengths  $\{(\sum_{t=0}^{i-1} W_t + 1), \dots, (\sum_{t=0}^{i-1} W_t + W_i)\}$ , where  $W_0 = 0$ .

This method of wavelength assignment is the motivation for sorting the SLDs in  $\tau$  in decreasing order of the number of lightpaths each SLD requests. Recall that the number of wavelengths  $W_i$  which must be assigned to  $\tau_i$  is the maximum number of lightpaths any SLD included in  $\tau_i$  requests. If such is the case, it is evident that it is more desirable to route SLDs which request a large number of lightpaths (i.e. the requests with high traffic demands) in the same element of the partition. In most cases, this will lead to a smaller number of total wavelengths assigned, as will be demonstrated on an example. This also means that high traffic demands are routed on mutually edge/time disjoint paths. We can intuitively see that this will reduce congestion as opposed to routing high traffic on the same path at the same time.

Furthermore, the SLDs with the same number of lightpaths are sorted in decreasing order of the lengths of their corresponding shortest paths in  $G$ . This is done since SLDs which have longer shortest paths are generally harder to route and should therefore be routed when more edges are available. Related work is given in [20]. If there are multiple SLDs with the same number of lightpaths and the same shortest path length, they are placed in random order.

To demonstrate the benefit of sorting the SLDs before creating a partition of  $\tau$ , a short example is considered. Suppose the set of SLDs in Table I and the physical network shown in Fig. 1.(a). Let the upper bound  $h$  on the physical length of a lightpath to be set to 2. The lightpaths of  $SLD_1$ ,  $SLD_2$ , and  $SLD_3$  all overlap in time, while the lightpaths of  $SLD_4$  are only in time conflict with those of  $SLD_1$ . Suppose we create a partition of  $\tau$  in the order in which the SLDs are shown in Table I. In that case,  $SLD_1$ ,  $SLD_2$  and  $SLD_4$  could be routed in the first element of the partition  $\tau_1$ , while  $SLD_3$  would require a second element  $\tau_2$ , as shown in Fig. 2.(a). Such a partition would require  $W_1 + W_2 = \max(n_1, n_2, n_4) + \max(n_3) = 10 + 9 = 19$  wavelengths to perform wavelength assignment. Now consider

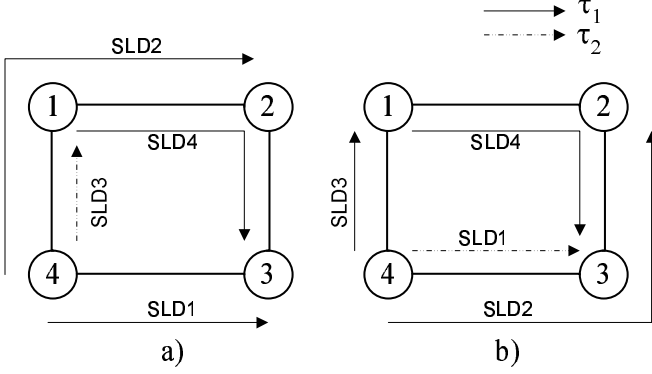


Fig. 2. An example of a partition of a set of SLDs  $\tau$  obtained using the  $DP\_RWA\_SLD$  algorithm (a) without sorting the SLDs and (b) with sorting the SLDs.

routing the SLDs in descending order of their requested lightpaths, i.e.  $\{SLD_2, SLD_3, SLD_4, SLD_1\}$ . This could result in a partition as follows:  $\tau_1 = \{SLD_2, SLD_3, SLD_4\}$  and  $\tau_2 = \{SLD_1\}$  shown in Fig. 2.(b). Such a partition would require  $W_1 + W_2 = \max(n_2, n_3, n_4) + \max(n_1) = 10 + 5 = 15$  wavelengths.

The pseudocode of  $DP\_RWA\_SLD$  follows.

**Input and initialization:**

$G = (V, E)$ ;  
 $\tau = \{SLD_1, \dots, SLD_M\}$ , where  $SLD_i = (s_i, d_i, n_i, \alpha_i, \omega_i)$ ,  $i = 1, \dots, M$ ; //the set of SLDs  
 $h = \max(\text{diam}(G), \sqrt{|E|})$ ;  
 $\lambda = 0$ ; //the number of wavelengths  
 $i := 0$ ; //the number of elements in the partition

**Begin:**

Sort the SLDs in  $\tau$  in decreasing order of their corresponding values of  $n_i$ . Sort requests with the equal values of  $n_i$  in decreasing order of the lengths of their shortest paths in  $G$  (if more than one request has the same length place them in random order);

**while**  $\tau$  is not empty **do**

$i := i + 1$ ;

$\tau_i = \{\}$ ;

$P_{\tau_i} = \{\}$ ; //paths of SLDs in  $\tau_i$

**for each**  $SLD_j \in \tau$  in the sorted order **do**

$G' = G$ ;

**for each**  $SLD_k \in \tau_i$  **do**

**if**  $(\alpha_j \leq \alpha_k \leq \omega_j)$  or  $(\alpha_k \leq \alpha_j \leq \omega_k)$  **then**

// $SLD_j \in \tau$  and  $SLD_k \in \tau_i$  overlap in time

Remove from  $G'$  all edges in  $P(SLD_k)$ ;

**end if**

**end for**

Find shortest path  $P(SLD_j)$  for  $SLD_j$  in  $G'$ ;

**if** the length of  $P(SLD_j)$  is  $\leq h$  **then**

Add  $P(SLD_j)$  to  $P_{\tau_i}$  and  $SLD_j$  to  $\tau_i$ ;

**end if**

**end for**

$W_i = \max$  value of  $n_j$  of any  $SLD_j \in \tau_i$ ;

For each SLD in  $\tau_i$ , assign to their corresponding lightpaths wavelengths  $(\lambda + 1), \dots$  up to  $(\lambda + W_i)$  if needed;

$\lambda := \lambda + W_i$ ;

$\tau := \tau \setminus \tau_i$ ;

**end while**

**End**

**B.  $DP\_RWA\_SLD^*$**

A related version of the  $DP\_RWA\_SLD$  algorithm is also proposed, referred to as  $DP\_RWA\_SLD^*$ . After creating an element of the partition  $\tau_i$ , a second attempt at routing into  $\tau_i$  the SLDs remaining in  $\tau$  is executed. The basic idea is the following. After creating each element of the partition  $\tau_i$  and assigning up to  $W_i$  wavelengths to each of the lightpaths of the SLDs included in  $\tau_i$ , we can see that there may be several SLDs that require less than  $W_i$  wavelengths. The edges on paths used by these SLDs could be utilized by routing other SLDs using the wavelengths assigned to  $\tau_i$  but not used on these particular edges. In other words, we want to “fill up”  $\tau_i$  by fully utilizing the set of wavelengths already assigned to it.

This is best shown on an example. Suppose we created an element  $\tau_i$  which is assigned  $W_i = 10$  wavelengths. Now suppose demand  $SLD_j$  routed in  $\tau_i$  requests 4 lightpaths (i.e.  $n_j = 4$ ). These lightpaths are assigned wavelengths  $(\sum_{t=0}^{i-1} W_t + 1)$ ,  $(\sum_{t=0}^{i-1} W_t + 2)$ ,  $(\sum_{t=0}^{i-1} W_t + 3)$  and  $(\sum_{t=0}^{i-1} W_t + 4)$ . Each edge on path  $P(SLD_j)$  could be used to route any SLD which demands  $(W_i - n_j) = 10 - 4 = 6$  lightpaths or less even if it overlaps in time with  $SLD_j$ . These lightpaths would simply be assigned wavelengths  $(\sum_{t=0}^{i-1} W_t + 5)$ ,  $(\sum_{t=0}^{i-1} W_t + 6), \dots$  up to  $(\sum_{t=0}^{i-1} W_t + 10)$  if necessary.

To successfully execute this modification, the following steps are added to algorithm  $DP\_RWA\_SLD$  giving rise to  $DP\_RWA\_SLD^*$ . After creating an element  $\tau_i$  and assigning  $W_i$  wavelengths in the same way as  $DP\_RWA\_SLD$  (i.e. one run of the **while** loop), we try and route the SLDs remaining in  $\tau$  a second time. As before, to route  $SLD_j \in \tau$  in  $\tau_i$  we start with graph  $G$  and check to see if it is in time conflict with any of the SLDs already routed in  $\tau_i$ . For the SLDs which are in time conflict with  $SLD_j$  and request more than  $(W_i - n_j)$  lightpaths, we delete the edges of their corresponding paths from  $G$  creating  $G'$ . The edges of those paths whose SLDs request  $(W_i - n_j)$  or less lightpaths remain in  $G'$  even though they are in time conflict with  $SLD_j$ .

$SLD_j$  is then routed on its shortest path  $P(SLD_j)$  in  $G'$ . If the routing is successful (i.e. there exists such a path and its length is  $\leq h$ ),  $SLD_j$  is added to  $\tau_i$  and removed from  $\tau$ . In order to assign wavelengths to the lightpaths of  $SLD_j$ , we do the following. We check all the edges in path  $P(SLD_j)$  and determine the highest wavelength  $W_{max}(P(SLD_j))$  used on any of these edges by an SLD in  $\tau_i$  which overlaps in time with  $SLD_j$ . We then assign wavelengths  $(W_{max}(P(SLD_j)) + 1), \dots, (W_{max}(P(SLD_j)) + n_j)$  to the  $n_j$  lightpaths of  $SLD_j$ . Note that  $W_{max}(P(SLD_j))$  is the highest wavelength assigned to some demand  $SLD_k \in \tau_i$  whose path overlaps with  $P(SLD_j)$  and can therefore be written as  $(\sum_{t=0}^{i-1} W_t + n_k)$ . Since prior to routing  $SLD_j$ , we deleted from  $G$  all those edges used by SLDs in time conflict with  $SLD_j$  requesting more than  $(W_i - n_j)$  lightpaths, we can be certain that  $n_k \leq W_i - n_j$ . It follows that  $W_{max}(P(SLD_j)) + n_j = \sum_{t=0}^{i-1} W_t + n_k + n_j \leq \sum_{t=0}^{i-1} W_t + W_i$ . This proves that we have not assigned to  $SLD_j$  any wavelength aside from the  $W_i$  wavelengths already assigned to  $\tau_i$ .

The pseudocode of  $DP\_RWA\_SLD^*$  follows:

**Input and initialization:**  
 $G = (V, E)$ ;  
 $\tau = \{SLD_1, \dots, SLD_M\}$ , where  
 $SLD_i = (s_i, d_i, n_i, \alpha_i, \omega_i), i = 1, \dots, M$ ; //the set of SLDs  
 $h = \max(\text{diam}(G), \sqrt{|E|})$ ;  
 $\lambda = 0$ ; //the number of wavelengths  
 $i := 0$ ; //the number of elements in the partition  
 $fillingUp := \text{false}$ ; //this indicates if we are starting to create a partition or "filling it up"  
**Begin:**  
Sort the SLDs in  $\tau$  in decreasing order of their corresponding values of  $n_i$ . Sort requests with the equal values of  $n_i$  in decreasing order of the lengths of their shortest paths in  $G$  (if more than one request has the same length place them in random order);  
**while**  $\tau$  is not empty **do**  
  **if**  $fillingUp == \text{false}$  **then**  
    Run one while loop of the  $DP\_RWA\_SLD$  algorithm;  
     $fillingUp := \text{true}$ ;  
  **else**  
    **for** each  $SLD_j \in \tau$  in the sorted order **do**  
       $G' = G$   
      **for** each  $SLD_k \in \tau_i$  **do**  
        **if**  $(\alpha_j \leq \alpha_k \leq \omega_j)$  or  $(\alpha_k \leq \alpha_j \leq \omega_k)$  **then**  
          // $SLD_j \in \tau$  and  $SLD_k \in \tau_i$  overlap in time  
          **if**  $n_k > W_i - n_j$  **then**  
            Remove from  $G'$  all edges in  $P(SLD_k)$ ;  
          **end if**  
        **end if**  
      **end for**  
      Find shortest path  $P(SLD_j)$  for  $SLD_j$  in  $G'$ ;  
      **if** the length of  $P(SLD_j)$  is  $\leq h$  **then**  
        Add  $P(SLD_j)$  to  $P_{\tau_i}$  and  $SLD_j$  to  $\tau_i$ ;  
        Find the max wavelength  $W_{max}(P(SLD_j))$  used by any SLD in  $\tau_i$  which uses any of the edges in  $P(SLD_j)$  and is in time conflict with  $SLD_j$ ;  
        Assign to  $SLD_j$  the wavelengths  $(W_{max}(P(SLD_j)) + 1), \dots, (W_{max}(P(SLD_j)) + n_j)$ ;  
      **end if**  
      **end for**  
       $fillingUp := \text{false}$ ;  
       $\tau := \tau \setminus \tau_i$ ;  
    **end if**  
  **end while**  
**End**

### C. Complexity Analysis

The computational complexity of the  $DP\_RWA\_SLD$  and  $DP\_RWA\_SLD^*$  algorithms follows. The  $DP\_RWA\_SLD$  algorithm first finds the all-pairs shortest paths between nodes in the physical network using Floyd's algorithm [21] in  $O(|V|^3)$  time. The  $M$  SLDs are then sorted in  $O(M \log M)$  time. The while loop runs  $O(M^2|V|^2)$  time giving us a final complexity of  $O(|V|^3 + M \log M + M^2|V|^2)$ . In the  $DP\_RWA\_SLD^*$  algorithm, the while loop is run twice as many times as in  $DP\_RWA\_SLD$  which still yields the same complexity. The complexity of the  $DP\_RWA\_SLD$  and  $DP\_RWA\_SLD^*$  algorithms is not comparable to that of the  $TS_{cg}/GGC$  and  $TS_{cn}/GGC$  algorithms since the former are constructive heuristics which end deterministically, while the later are improvement heuristics which can be terminated at any time and still obtain a feasible solution. However, numerical results (see Section VII) indicate that in order to obtain good solutions using the tabu search algorithms, a fair number of iterations need to be run resulting in execution times drastically longer than those of the greedy algorithms.

## VI. LOWER BOUNDS

Since the algorithms considered in this paper are heuristics which obtain upper bounds on the minimal objective function values, it is useful to have good lower bounds in order to assess the quality of the sub-optimal solutions. A simplistic lower bound on the number of wavelengths needed to perform successful routing and wavelength assignment on a set of scheduled lightpath demands such that the *group lightpath constraint* is satisfied is

$$W_{n_{max}}^{LB} = \max_{i=1, \dots, M} \{n_i\}. \quad (1)$$

This represents the maximum number of lightpaths requested by any SLD in  $\tau$ . However, this lower bound is not necessarily efficient for a set of lightpath requests highly correlated in time. In [22], a simple lower bound on the number of wavelengths required to set up a regular virtual topology in wavelength routed optical networks is obtained by comparing the fixed logical degree to the maximum physical degree in the network. We further develop this idea of the logical to physical degree ratio to derive a tighter lower bound for the RWA SLD problem as follows.

Let

$$S_s = \{SLD_i | s_i = s, i = 1, \dots, M\}, \quad \forall s \in V \quad (2)$$

be the set of SLDs whose source node is node  $s$ .

Let

$$T_s^S = \{\alpha_i \cup \omega_i | SLD_i \in S_s, i = 1, \dots, M\}, \quad \forall s \in V \quad (3)$$

be an ordered set of moments in time when some SLD in  $S_s$  is either set up and/or some SLD in  $S_s$  is torn down. If  $T_s^S = \{t_{s_1}, \dots, t_{s_{|T_s^S|}}\}$ , then  $t_{s_1} < t_{s_1} \dots < t_{s_{|T_s^S|}}$  and  $|T_s^S| \leq 2|S_s|$ .

Let

$$TO_{s_j}^S = \{SLD_k | [t_{s_j}, t_{s_{j+1}}] \subseteq [\alpha_k, \omega_k], SLD_k \in S_s\}, \quad (4)$$

$$\forall s \in V, \forall j = 1, \dots, |T_s^S| - 1,$$

be the set of SLDs whose source node is  $s$  and are active in time interval  $[t_{s_j}, t_{s_{j+1}}]$ . This means that all the SLDs in  $TO_{s_j}^S$  overlap in time. Furthermore, let  $TO_{s_j}^S$  be an *ordered* set with respect to the number of lightpaths requested by each SLD. In other words, if  $TO_{s_j}^S = \{SLD_{to_1^{s_j}}, \dots, SLD_{to_{|TO_{s_j}^S|}^{s_j}}\}$ , then  $n_{to_1^{s_j}} < n_{to_2^{s_j}} < \dots < n_{to_{|TO_{s_j}^S|}^{s_j}}$ .

Lastly, let  $\Delta_{phy_s}$  be the out-degree<sup>5</sup> of node  $s$  in the physical topology. All the lightpaths of the SLDs in  $S_s$  will surely be routed over one of the  $\Delta_{phy_s}$  outgoing edges adjacent to node  $s$ . If the individual lightpaths of a single SLD do not necessarily need to be routed on the same path (i.e. if we relax the *group lightpath constraint*), each individual lightpath can be routed over any one of the  $\Delta_{phy_s}$  outgoing edges. Lightpaths in  $S_s$  which overlap in time, i.e. their respective SLDs are both in at least one set  $TO_{s_j}^S, j = 1, \dots, |T_s^S| - 1$ , and which are routed over the same physical edge must be assigned different

<sup>5</sup>According to our problem definition, the physical out-degree is equal to the physical in-degree for each node in  $V$  since we assume that each link in the physical topology represents two fibers - one in each direction.



wavelengths. To route and assign wavelengths to the lightpaths of the SLDs in some set  $TO_{s_j}^S$ , at least one physical link will have

$$W_{TO_{s_j}^S}^{LB} = \left\lceil \frac{\sum_{i|SLD_i \in TO_{s_j}^S} n_i}{\Delta_{phy_s}} \right\rceil, \quad (5)$$

$$\forall s \in V, \forall j \in \{1, \dots, |T_s^S| - 1\},$$

lightpaths routed over it and therefore require at least as many wavelengths.

If we consider the lightpaths of the SLDs in set  $TO_{s_j}^S$  to represent a logical topology over the physical topology which is constant in the corresponding time interval,  $W_{TO_{s_j}^S}^{LB}$  represents the ratio of logical to physical degree of node  $s$  in time interval  $[t_{s_j}, t_{s_{j+1}}]$ . The highest such ratio

$$W_S^{LB} = \max_{s \in V} \max_{1 \leq j \leq |T_s^S| - 1} W_{TO_{s_j}^S}^{LB} \quad (6)$$

for any source node in the network over all time intervals is a lower bound on the number of wavelengths needed to perform routing and wavelength assignment for a set of scheduled lightpath demands  $\tau$ . Note that  $W_S^{LB}$  is a lower bound for the RWA SLD problem where the *group lightpath constraint* is relaxed. Since imposing such a constraint makes the problem harder,  $W_S^{LB}$  is also a lower bound for the constrained problem.

Furthermore, assuming the *group lightpath constraint* does apply, we suggest an alternative lower bound, referred to as  $W_S^{LB'}$ . Let the load of  $SLD_i$  be its corresponding number of lightpaths  $n_i$ . A lower bound  $W_{TO_{s_j}^S}^{LB'}$  on the number of wavelengths needed to perform routing and wavelength assignment of the SLDs in set  $TO_{s_j}^S$  is the maximum load on any outgoing physical link adjacent to  $s$  after performing optimal load balancing of the  $|TO_{s_j}^S|$  SLDs over the  $\Delta_{phy_s}$  links. If  $n_i = 1$  for all SLDs in  $TO_{s_j}^S$ , load balancing is trivial and gives the same lower bound as (6). Otherwise, this problem is NP-Complete. For very small cases, exhaustive search could be applied. However, for larger cases this is not practical. Since we do not actually need to perform load balancing but are solely interested in the maximum load of the optimal solution, we can use a lower bound on the maximum load, which, in turn, is a lower bound on the number of wavelengths needed. We know that at least

$$N_{s_j}^S = \left\lceil \frac{|TO_{s_j}^S|}{\Delta_{p_s}} \right\rceil, \quad \forall s \in V, \forall j \in \{1, \dots, |T_s^S| - 1\} \quad (7)$$

SLDs (not individual lightpaths) will surely be routed on at least one physical outgoing link adjacent to  $s$  in time period  $[t_{s_j}, t_{s_{j+1}}]$ . Defined as such,  $N_{s_j}^S \leq |TO_{s_j}^S|$ . By summing up the load of the  $N_{s_j}^S$  SLDs in  $TO_{s_j}^S$  with the lightest load, i.e. the lowest number of lightpaths  $n_i$ , we obtain a lower bound on the maximum load. Since  $TO_{s_j}^S$  is a set of SLDs sorted in nondecreasing order of their corresponding number of SLDs, the lower bound on the number of lightpaths routed over at least one of the outgoing edges of  $s$  in time interval  $[t_{s_j}, t_{s_{j+1}}]$  is the sum of the number of lightpaths of the first  $N_{s_j}^S$  SLDs in  $TO_{s_j}^S$ . In other words, if  $TO_{s_j}^S =$

$\{SLD_{to_1^{s_j}}, \dots, SLD_{to_{|TO_{s_j}^S|}^{s_j}}\}$ , then

$$W_{TO_{s_j}^S}^{LB'} = \sum_{i=1}^{N_{s_j}^S} n_{to_i^{s_j}}, \quad \forall s \in V, \forall j \in \{1, \dots, |T_s^S| - 1\}. \quad (8)$$

It follows that the lower bound on the number of wavelengths needed to perform RWA of a set of scheduled lightpath demands in the case that the *group lightpath constraint* applies is

$$W_S^{LB'} = \max_{s \in V} \max_{1 \leq j \leq |T_s^S| - 1} W_{TO_{s_j}^S}^{LB'}. \quad (9)$$

Note that for some cases, e.g. when one or a few SLDs request a very large number of lightpaths, bounds (1) and/or (6) may be tighter. As a result, we consider all the mentioned bounds.

The above discussion regarding lower bounds derived by considering SLDs with common source nodes can also be applied to SLDs with common destination nodes. Namely, if SLDs terminate at the same node  $d \in V$ , they will surely be routed over one of the  $\Delta_{phy_d}$  in-degree edges adjacent to node  $d$ . Let

$$D_d = \{SLD_i | d_i = d, i = 1, \dots, M\}, \quad \forall d \in V \quad (10)$$

be the set of SLDs whose destination node is  $d$ . This is analogous to (2) for SLDs with source node  $s$ . Sets  $T_d^D$  and  $TO_d^D$  representing the time intervals and time overlapping SLDs in  $D_d$  can be obtained from (3) and (4), respectively, by replacing  $S$  with  $D$  and  $s$  with  $d$ .  $N_d^D$  can be obtained in the same manner from (7). This leads to two additional lower bounds,

$$W_D^{LB} = \max_{d \in V} \max_{1 \leq j \leq |T_d^D| - 1} W_{TO_{d_j}^D}^{LB} = \max_{d \in V} \max_{1 \leq j \leq |T_d^D| - 1} \left\lceil \frac{\sum_{i|SLD_i \in TO_{d_j}^D} n_i}{\Delta_{phy_d}} \right\rceil \quad (11)$$

and

$$W_D^{LB'} = \max_{d \in V} \max_{1 \leq j \leq |T_d^D| - 1} W_{TO_{d_j}^D}^{LB'} = \max_{d \in V} \max_{1 \leq j \leq |T_d^D| - 1} \left\{ \sum_{i=1}^{N_{d_j}^D} n_{to_i^{d_j}} \right\} \quad (12)$$

analogous to (6) and (9).

The preceding discussion shows a lower bound on the number of wavelengths needed to solve the RWA SLD problem without the *group lightpath constraint* to be

$$W_{LB} = \max\{W_S^{LB}, W_D^{LB}\}. \quad (13)$$

For the problem augmented with the *group lightpath constraint*, a tighter lower bound is

$$W'_{LB} = \max\{W_{n_{max}}^{LB}, W_S^{LB}, W_S^{LB'}, W_D^{LB}, W_D^{LB'}\}. \quad (14)$$

In the example given in Table I, supposing the physical topology shown in Fig. 1.(a), the lower bound  $W'_{LB}$  would be calculated as follows. In this example,  $\tau =$

$\{SLD_1, SLD_2, SLD_3, SLD_4\}$ ,  $M = 4$ , and  $V = \{1, 2, 3, 4\}$ , while the physical in and out-degree of each node is  $\Delta_{phy_i} = 2$ ,  $\forall i \in V$ . Lower bound  $W_{n_{max}}^{LB} = 10$  represents the maximum number of lightpaths requested by any SLD in  $\tau$ . To calculate  $W_S^{LB}$  we must find  $W_{TO_{s_j}^S}^{LB}$  for each  $s$  and  $j$ . For  $s = 1$ ,  $S_1 = \{SLD_4\}$ , while for  $s = 4$ ,  $S_4 = \{SLD_1, SLD_2, SLD_3\}$ . For  $s = 2$  or  $s = 3$ , these sets are empty since nodes 2 and 3 are not source nodes for any requested SLD.  $T_1^S = \{1:00, 2:00\}$  and  $T_4^S = \{1:00, 2:00, 6:00, 7:00\}$ , while  $TO_{11}^S = \{SLD_4\}$ ,  $TO_{41}^S = \{SLD_1\}$ ,  $TO_{42}^S = \{SLD_1, SLD_3, SLD_2\}$ , and  $TO_{43}^S = \{SLD_3\}$ . Note that these sets are ordered in nondecreasing order of the number of lightpaths requested by the SLDs in the set. Lower bounds over the source nodes and time intervals are  $W_{TO_{11}^S}^{LB} = \lceil 7/2 \rceil = 4$ ,  $W_{TO_{41}^S}^{LB} = \lceil 5/2 \rceil = 3$ ,  $W_{TO_{42}^S}^{LB} = \lceil (5+10+9)/2 \rceil = 12$  and  $W_{TO_{43}^S}^{LB} = \lceil 9/2 \rceil = 5$ . It follows that  $W_S^{LB} = 12$ . Furthermore,  $N_{11}^S = 1$ ,  $N_{41}^S = 1$ ,  $N_{42}^S = 2$ , and  $N_{43}^S = 1$ . It follows that  $W_{TO_{11}^S}^{LB'} = n_4 = 7$ ,  $W_{TO_{41}^S}^{LB'} = n_1 = 5$ ,  $W_{TO_{42}^S}^{LB'} = n_1 + n_3 = 5 + 9 = 14$ , and  $W_{TO_{43}^S}^{LB'} = n_2 = 10$ . This leads to lower bound  $W_S^{LB'} = 14$ .  $W_D^{LB}$  and  $W_D^{LB'}$  are analogously found to be 6 and 10 respectively. It follows that a lower bound for the RWA SLD without the *group lightpath constraint* is  $W_{LB} = \max\{12, 6\} = 12$ , while  $W_{LB}' = \max\{10, 12, 14, 6, 10\} = 14$  gives a lower bound for the constrained version of the problem. In the example in Fig. 2.(b), we can see that a routing and wavelength assignment was found with 15 wavelengths, demonstrating the efficiency of the bound for this case.

## VII. ANALYSIS OF COMPUTATIONAL RESULTS

### A. Experimental method and numerical results

The  $TS_{cg}/GGC$  [9],  $TS_{cn}/GGC$ ,  $DP\_RWA\_SLD$ , and  $DP\_RWA\_SLD^*$  algorithms for the Routing and Wavelengths Assignment problem of Scheduled Lightpath Demands were all implemented in C++ and run on a PC powered by a P4 2.8GHz processor. The  $TS_{cg}$  [9] and the suggested  $TS_{cn}$  tabu search algorithms for the routing subproblem were run in combination with the  $GGC$  graph coloring algorithm from [10] for wavelength assignment. The source code for the  $GGC$  algorithm was provided by the authors. Random numbers were generated using the R250 random number generator [23].

We tested the algorithms using the hypothetical U.S. backbone given in [9]. The network consists of 29 nodes and 44 edges which are assumed to be bidirectional. The weight of an edge represents its physical length. Using a Perl script provided by the authors of [9], 60 sets of  $M=30$  SLDs were generated with time correlation 0.01, and 60 sets with time correlation 0.8. Each SLD could request at most 10 lightpaths. Time correlation closer to 0 means that the SLDs are weakly time correlated while time correlation closer to 1 means that the SLDs generated are strongly time correlated. For exact definition of the time correlation parameter used, refer to [9]. In this paper, we will refer to this parameter as  $\delta$ .

As in [9], the  $TS_{cg}/GGC$  algorithm was run with a neighborhood size of 200, the length of the tabu list was set to 2 times the neighborhood size and the number of allowed

iterations without improvement was set to 150. Regarding the  $TS_{cn}/GGC$  algorithm, the size of the neighborhood is not an input parameter since  $TS_{cn}$  uses an adaptive neighborhood. The remaining parameters for the  $TS_{cn}/GGC$  algorithm were determined experimentally. Since effective tabu tenures, i.e. the length of the tabu list, have been shown to depend on the size of the problem [13], we tested the algorithm with tabu tenures proportional to the number of possible neighboring solutions. Since a neighboring solution with respect to a current one is defined such that one of the  $M$  SLDs is routed on a different path, there are  $M(K - 1)$  possible neighbors. Experimental results indicated that a tabu list of size  $M(K - 1)/10$  was long enough to disable cycling and short enough so as not to restrict the search. Setting the number of iterations without improvement to a value dependant on the size of the problem also proved effective. Empirical testing also showed that applying diversification every  $M(K - 1)/3$  iterations helped obtain good results.

Both the  $TS_{cg}/GGC$  and  $TS_{cn}/GGC$  algorithms were run for 3000 iterations, as in [9], and  $K$  ranged from 2 to 5. Since a tabu search algorithm can reach its best incumbent solution in any iteration and then continue running without any improvement (even with diversification), we recorded the iteration in which the best solution was first found for each test case for both tabu search algorithms. We also measured the average execution time per iteration and the time it took each tabu search algorithm to reach its best solution. These results, averaged over the 60 test cases, and the average number of wavelengths of the solutions obtained by each of the tabu search algorithms for time correlations 0.01 and 0.8 are shown in Tables II and III, respectively. The number of wavelengths and execution times for the  $DP\_RWA\_SLD$  and  $DP\_RWA\_SLD^*$  algorithms and lower bound  $W_{LB}'$  are also shown in Tables II and III. For further insight regarding execution time, in Table IV, the number of iterations and the time it took to reach the best solution by each of the tabu search algorithms for the test case for which they performed *worst* are shown. Note that the results shown regarding the execution times of the the tabu search algorithms do not include the time it takes to subsequently run the  $GGC$  algorithm. The average execution times of the  $GGC$  algorithm were around 12 and 18 seconds for time correlations 0.01 and 0.8, respectively.

We can see that  $TS_{cn}/GGC$  performs better than (or equal to)  $TS_{cg}/GGC$  in all cases with respect to solution quality and execution time. For test data with time correlation 0.01, the initial solution is often optimal since most of the SLDs do not overlap in time. These test cases, although helpful in showing the benefit of performing RWA considering scheduled lightpath demands as opposed to static lightpath demands, are less effective in comparing the results of RWA SLD algorithms. The results for time correlation 0.8 are much more interesting. The specific test cases where the number of wavelengths differed in the solutions obtained by each of the tabu search algorithms are shown in Fig. 3. We can see that  $TS_{cn}/GGC$  used less wavelengths in all cases.

According to Table II, the  $DP\_RWA\_SLD$  algorithm outperforms both tabu search algorithms in combination with the  $GGC$  algorithm for time correlation 0.01. For time cor-

TABLE II

HYPOTHETICAL U.S. NETWORK [9],  $\delta = 0.01$ ,  $M = 30$ : AVG. NO. OF WAVELENGTHS, AVG. ITER. IN WHICH THE BEST SOLUTION WAS OBTAINED, AVG. EXEC. TIME PER ITERATION AND AVG. EXEC. TIME TO BEST SOLUTION FOR ALGORITHMS  $TS_{cg}/GGC$  [9] AND  $TS_{cn}/GGC$ ; AVG. NO. OF WAVELENGTHS AND AVG. EXEC. TIME FOR ALGORITHMS  $DP\_RWA\_SLD$  AND  $DP\_RWA\_SLD^*$ , AND LOWER BOUND  $W'_{LB}$ .

$K$	$TS_{cg}/GGC$ [9]				$TS_{cn}/GGC$				Lower bound Avg. $W'_{LB}$
	Avg. wave-lengths	Avg. iter. found best	Avg. time/iter (ms)	Avg. time to best sol. (ms)	Avg. wave-lengths	Avg. iter. found best	Avg. time/iter (ms)	Avg. time to best sol. (ms)	
2	11.12	0.81	395.37	85.74	11.12	3.53	4.48	27.57	9.90
3	10.50	12.25	402.11	4918.09	10.50	2.72	2.73	10.44	
4	10.28	35.38	402.12	14204.34	10.28	9.92	2.69	33.71	
5	10.22	26.63	495.22	10901.78	10.22	10.52	2.59	38.65	
$DP\_RWA\_SLD$				$DP\_RWA\_SLD^*$					
Avg. wavelengths		Avg. execution time (ms)		Avg. wavelengths		Avg. execution time (ms)			
10.00		0.76		9.90		0.88			

TABLE III

HYPOTHETICAL U.S. NETWORK [9],  $\delta = 0.8$ ,  $M = 30$ : AVG. NO. OF WAVELENGTHS, AVG. ITER. IN WHICH THE BEST SOLUTION WAS OBTAINED, AVG. EXEC. TIME PER ITERATION AND AVG. EXEC. TIME TO BEST SOLUTION FOR ALGORITHMS  $TS_{cg}/GGC$  [9] AND  $TS_{cn}/GGC$ ; AVG. NO. OF WAVELENGTHS AND AVG. EXEC. TIME FOR ALGORITHMS  $DP\_RWA\_SLD$  AND  $DP\_RWA\_SLD^*$ , AND LOWER BOUND  $W'_{LB}$ .

$K$	$TS_{cg}/GGC$ [9]				$TS_{cn}/GGC$				Lower bound Avg. $W'_{LB}$
	Avg. wave-lengths	Avg. iter. found best	Avg. time/iter (ms)	Avg. time to best sol. (ms)	Avg. wave-lengths	Avg. iter. found best	Avg. time/iter (ms)	Avg. time to best sol. (ms)	
2	14.33	28.48	396.90	11329.26	13.85	77.42	7.66	595.97	10.08
3	13.78	59.92	399.40	37730.77	12.65	13.82	4.21	59.90	
4	12.47	240.12	404.43	98217.44	11.68	36.45	4.03	147.18	
5	11.70	321.03	404.24	130825.00	11.20	96.38	3.89	368.59	
$DP\_RWA\_SLD$				$DP\_RWA\_SLD^*$					
Avg. wavelengths		Avg. execution time (ms)		Avg. wavelengths		Avg. execution time (ms)			
11.90		0.94		10.63		1.07			

TABLE IV

HYPOTHETICAL U.S. NETWORK [9],  $M = 30$ , WORST CASES: TEST CASES FOR WHICH THE BEST SOLUTION WAS FOUND IN THE HIGHEST ITERATION, THE CORRESPONDING ITERATION, AVG. EXECUTION TIME PER ITERATION AND THE EXECUTION TIME TO BEST SOLUTION FOR ALGORITHMS  $TS_{cg}/GGC$  [9] AND  $TS_{cn}/GGC$

$\delta$	$K$	$TS_{cg}/GGC$ [9]				$TS_{cn}/GGC$			
		Test case	Iteration found best	Avg. time/iter. (s)	Time to best solution (s)	Test case	Iteration found best	Avg. time/iter. (s)	Time to best solution (s)
0.01	2	52	2	0.3921	0.784	54	86	0.0077	0.666
	3	21	158	0.4059	64.137	21	45	0.0047	0.210
	4	21	277	0.4035	111.770	21	141	0.0044	0.624
	5	24	265	0.4012	106.344	21	149	0.0045	0.671
0.8	2	39	1509	0.3965	598.240	39	1348	0.0078	10.467
	3	22	1204	0.4006	482.276	17	219	0.0051	1.112
	4	41	2419	0.4020	972.498	39	457	0.0036	1.649
	5	33	2989	0.4041	1207.822	54	1525	0.0042	6.442

relation 0.8,  $DP\_RWA\_SLD$  outperforms  $TS_{cg}/GGC$  for cases where  $K = 2, 3$ , and 4, and outperforms  $TS_{cn}/GGC$  for cases where  $K = 2$  and 3.  $DP\_RWA\_SLD$  has the shortest execution time among all the mentioned algorithms for all cases. The  $DP\_RWA\_SLD^*$  algorithm outperforms  $TS_{cn}/GGC$ ,  $TS_{cg}/GGC$  and  $DP\_RWA\_SLD$  for all values of  $K$  in solution quality and both tabu search algorithms in execution time. Since the  $TS_{cn}/GGC$  algorithm uses less wavelengths than  $TS_{cg}/GGC$  for all test cases, and the  $DP\_RWA\_SLD^*$  algorithm uses less wavelengths than the  $DP\_RWA\_SLD$  algorithm in all cases, we compare the results of  $TS_{cn}/GGC$  and  $DP\_RWA\_SLD^*$ . The test cases where the solutions obtained by  $TS_{cn}/GGC$  and  $DP\_RWA\_SLD^*$  differed for time correlation 0.8 are shown in Fig. 4. The  $TS_{cn}/GGC$  algorithm performed better in 4 cases, while the  $DP\_RWA\_SLD^*$  algorithm performed better

TABLE V

HYPOTHETICAL U.S. NETWORK [9],  $M = 30$ : AVERAGE NEIGHBORHOOD SIZE FOR ALGORITHM  $TS_{cn}/GGC$

$K$	Average neighborhood size	
	$\delta = 0.01$	$\delta = 0.8$
2	0.810	2.035
3	0.711	1.946
4	0.662	1.879
5	0.661	1.825

in 14 cases.

Since the neighborhood of the  $TS_{cn}/GGC$  algorithm is adaptive, we recorded the average neighborhood sizes for the  $TS_{cn}/GGC$  algorithm. These results are shown in Table V. We can see that the proposed neighborhood reduction technique dramatically reduces the size of the neighborhood

TABLE VI

HYPOTHETICAL U.S. NETWORK [9],  $M = 30$ : AVG. PHYSICAL HOP LENGTH OF THE LIGHTPATHS IN THE SOLUTIONS OBTAINED BY ALGORITHMS  $TS_{cg}/GGC$  [9],  $TS_{cn}/GGC$ ,  $DP\_RWA\_SLD$  AND  $DP\_RWA\_SLD^*$

$K$	Time correlation $\delta = 0.01$				Time correlation $\delta = 0.8$			
	$TS_{cg}/GGC$ [9]	$TS_{cn}/GGC$	$DP\_RWA\_SLD$	$DP\_RWA\_SLD^*$	$TS_{cg}/GGC$ [9]	$TS_{cn}/GGC$	$DP\_RWA\_SLD$	$DP\_RWA\_SLD^*$
2	3.755	3.828			3.847	3.987		
3	3.888	3.788	3.818	3.819	3.983	4.001	3.980	3.993
4	4.006	3.876			4.468	4.167		
5	4.032	3.912			4.631	4.248		

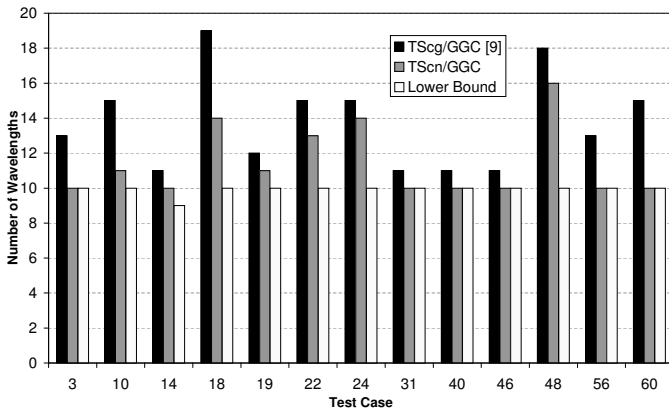


Fig. 3. Hypothetical U.S. network [9],  $\delta = 0.8$ ,  $M = 30$ : The number of wavelengths of the solutions obtained by algorithms  $TS_{cg}/GGC$  [9] and  $TS_{cn}/GGC$ , and lower bound  $W'_{LB}$  for the test cases where the number of wavelengths differ.

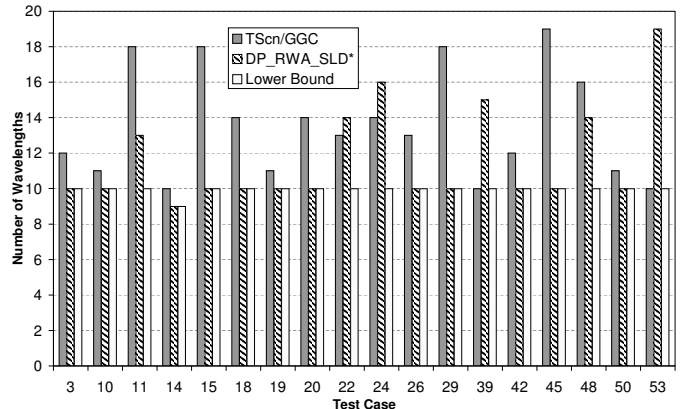


Fig. 4. Hypothetical U.S. network [9],  $\delta = 0.8$ ,  $M = 30$ : The number of wavelengths of the solutions obtained by algorithms  $TS_{cn}/GGC$  and  $DP\_RWA\_SLD^*$ , and lower bound  $W'_{LB}$  for the test cases where the number of wavelengths differ.

and yet obtains good results. The average neighborhood size for test cases with time correlation 0.01 is less than one since for many of the test cases, solutions can be found where none of the SLDs overlap in both time and space due to the very small time correlation. Such solutions give conflict graphs where none of the nodes representing lightpaths of different SLDs are adjacent, and thus RWA is trivial.

The average physical hop lengths of the lightpaths established by each of the algorithms are shown in Table VI. For test cases with time correlation 0.01, the  $TS_{cn}/GGC$  algorithm established shorter lightpaths than the  $TS_{cg}/GGC$  algorithm for cases where  $K = 3, 4$ , and  $5$ . The  $DP\_RWA\_SLD^*$  set up shorter lightpaths than the tabu search algorithms for all cases but  $K = 2$  for  $TS_{cg}/GGC$  and  $K = 3$  for  $TS_{cn}/GGC$ . For test cases with time correlation 0.8, the  $TS_{cn}/GGC$  and  $DP\_RWA\_SLD^*$  algorithms were better than  $TS_{cg}/GGC$  for  $K = 4$  and  $5$ , while the latter performed better for  $K = 2$  and  $3$ . The  $DP\_RWA\_SLD$  algorithm established the shortest lightpaths for both time correlations.

The algorithms were also tested on a reference European core network topology shown in Fig. 5 which was designed as part of the COST Action 266 project [24]. This network consists of 14 nodes and 39 edges. 20 test cases with time correlation  $\delta = 0.95$  and  $M = 200$  SLDs were generated, where each SLD can request at most 10 lightpaths. The tabu search algorithms were run with  $K = 5$ . The average number of wavelengths and the average execution times to reach the best solution are shown in Table VII. For the European network, all three proposed algorithms significantly

outperform the  $TS_{cg}/GGC$  algorithm with respect to both the number of wavelengths and execution time<sup>6</sup>. The wavelengths required for the specific test cases are shown in Fig. 6. The average neighborhood size for the  $TS_{cn}/GGC$  algorithm was 1.540. The average physical hop lengths of the established lightpaths were as follows: 3.503, 3.655, 2.717 and 2.730 for algorithms  $TS_{cg}/GGC$ ,  $TS_{cn}/GGC$ ,  $DP\_RWA\_SLD$  and  $DP\_RWA\_SLD^*$ , respectively. Here,  $TS_{cg}/GGC$  outperformed  $TS_{cn}/GGC$ , but  $DP\_RWA\_SLD$  and  $DP\_RWA\_SLD^*$  again established shorter lightpaths than the tabu search algorithms.

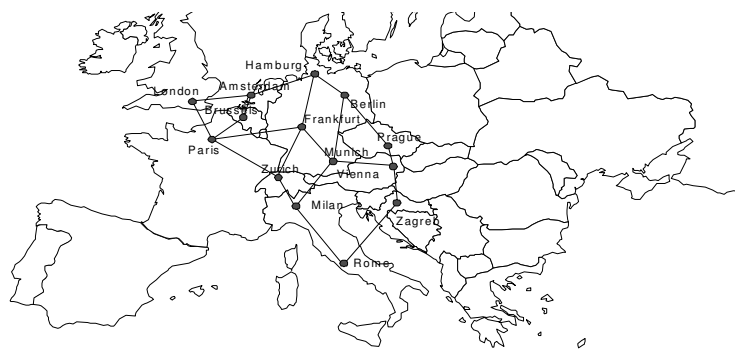


Fig. 5. The hypothetical European core network from [24].

<sup>6</sup>The run time for the  $GGC$  algorithm for the test cases generated for the European network was about 220 seconds.

TABLE VII

HYPOTHETICAL EUROPEAN NETWORK [24],  $\delta = 0.95$ ,  $M = 200$ : AVG. NO. OF WAVELENGTHS, AVG. ITER. IN WHICH THE BEST SOLUTION WAS OBTAINED, AVG. EXEC. TIME PER ITERATION AND AVG. EXEC. TIME TO BEST SOLUTION FOR ALGORITHMS  $TS_{cg}/GGC$  [9] AND  $TS_{cn}/GGC$ ; AVG. NO. OF WAVELENGTHS AND AVG. EXEC. TIME FOR ALGORITHMS  $DP\_RWA\_SLD$  AND  $DP\_RWA\_SLD^*$ , AND LOWER BOUND  $W'_{LB}$ .

$K$	$TS_{cg}/GGC$ [9]				$TS_{cn}/GGC$				Lower bound Avg. $W'_{LB}$
	Avg. wave-lengths	Avg. iter. found best	Avg. time/iter (s)	Avg. time to best sol. (s)	Avg. wave-lengths	Avg. iter. found best	Avg. time/iter (s)	Avg. time to best sol. (s)	
5	29.00	935.40	1.403	1301.228	22.70	905.00	0.076	69.837	13.05
	$DP\_RWA\_SLDC$				$DP\_RWA\_SLD^*$				
	Avg. wavelengths		Avg. execution time (s)		Avg. wavelengths		Avg. execution time (s)		
	21.80		0.0203		19.45		0.0227		

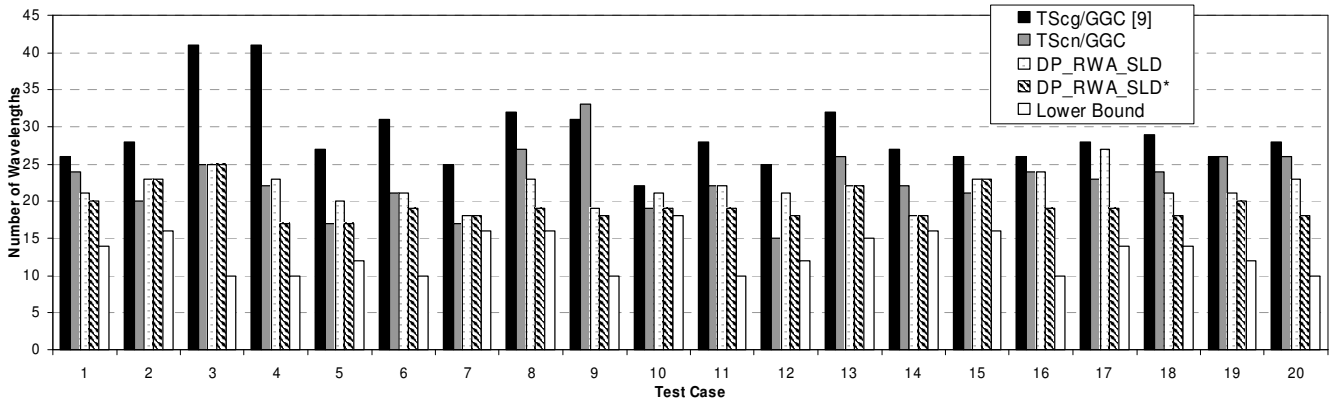


Fig. 6. Hypothetical European network [24],  $\delta = 0.95$ ,  $M = 200$ : The number of wavelengths of the solutions obtained by  $TS_{cg}/GGC$  [9],  $TS_{cn}/GGC$ ,  $DP\_RWA\_SLD$ , and  $DP\_RWA\_SLD^*$ , and lower bound  $W'_{LB}$ .

## B. Discussion

All three proposed algorithms give better quality solutions in less time than the  $TS_{cg}/GGC$  [9] algorithm for the data tested in this paper. The proposed tabu search algorithm,  $TS_{cn}/GGC$ , uses less wavelengths than  $TS_{cg}/GGC$  and yet evaluates only a few neighbors in each iteration. The very efficient neighborhood reduction technique, in addition to improving the quality of the solutions, drastically reduces the execution time per iteration with respect to the previous art. The time per iteration of the  $TS_{cn}/GGC$  algorithm is not only dramatically shorter than that of the  $TS_{cg}/GGC$  algorithm, but surprisingly decreases as  $K$  increases for the cases tested. One of the reasons for this is that, for this data set, the average neighborhood size decreased as  $K$  increased (see Table V). The neighborhood size depends on the topology of the conflict graph and is therefore dependent on  $K$ . Although, in general, the neighborhood size does not necessarily decrease as  $K$  increases, such was the case for the data instances evaluated in this paper. Examining the behavior of the algorithm further, we found that when  $K$  is small, it occurs more frequently that all neighboring solutions are on the tabu list. In such cases, alternative neighboring solutions outside the reduced neighborhood set are examined until a valid neighbor is found. This slightly increases the run-time of the algorithm.

Another point worth mentioning, regarding the  $TS_{cn}/GGC$  algorithm, is that the number of iterations required to reach the best solution is higher when  $K = 2$  than when  $K > 2$ . Since neighborhood reduction is so drastic, the search is too

restrictive when  $K$  is very small. The search technique is much more effective when  $K$  is larger, which is convenient since these are the cases when the problem size is bigger and the corresponding combinatorial optimization problem is harder.

Regarding the proposed greedy algorithms, both  $DP\_RWA\_SLD$  and  $DP\_RWA\_SLD^*$  outperform  $TS_{cg}/GGC$  in all cases with respect to the number of wavelengths and execution time. These algorithms also establish shorter lightpaths. The  $DP\_RWA\_SLD$  and  $DP\_RWA\_SLD^*$  algorithms are easy to implement, give good quality solutions and can be applied to large networks due to their very short execution times.  $DP\_RWA\_SLD^*$  is negligibly slower and establishes slightly longer lightpaths than  $DP\_RWA\_SLD$ , but performs significantly better with respect to the number of wavelengths used.

Although the greedy algorithm  $DP\_RWA\_SLD^*$  is better on average than the proposed tabu search algorithm  $TS_{cn}/GGC$ , for specific test cases this is sometimes not true (see Figure 4 and 6). An effort was made to determine a pattern in test cases in which the tabu search algorithm performed better than the greedy algorithm, and *vice versa*. However nothing conclusive was found. This is not surprising since both strategies (greedy and tabu) are heuristics and the search trajectory can be unpredictable depending on input data. If the input data in an instance is such that a greedy strategy provides an effective minimization direction, it is possible that nothing better will be obtained by the improvement mechanism of tabu search. Also, the initial solution used in tabu search can sometimes be inefficient (far away from the optimal solution),

in which case it might be difficult to reach a very good suboptimal solution via a restricted neighborhood search. In some other instances, input data can be such that a good initial solution and effective improvements are provided with tabu search strategy, while a greedy strategy lacks flexibility in search directions and ends with an inferior solution. Due to short computational times, for smaller problems both  $TS_{cn}/GGC$  and  $DP\_RWA\_SLD^*$  could be applied and the better solution selected. For larger problems it might be better to run the greedy algorithm, compare the solution with the available lower bound, and in case of a significant gap between the solution and its lower bound, the tabu search algorithm could be applied in an attempt to improve the solution.

### VIII. CONCLUSION

In order to efficiently utilize resources in wavelength-routed optical networks, it is necessary to successfully solve the problem of Routing and Wavelength Assignment. Scheduled lightpath demands, where the set-up and tear-down times of lightpaths are known *a priori*, could be considered by RWA algorithms in order to utilize the network's resources even further. In this work, efficient heuristic algorithms are proposed for the routing and wavelength assignment of scheduled lightpath demands in networks with no wavelength converters. Testing and comparing with an existing algorithm for the RWA SLD problem shows that these algorithms not only provide solutions of better or equal quality, but are dramatically faster. New lower bounds for the RWA SLD problem are also proposed. Further avenues of research will include developing similar algorithms for routing and wavelength assignment in networks with full or limited wavelength conversion. Networks equipped with a limited number of transmitters and receivers at each node and/or a limited number of wavelengths on each link will also be considered. Furthermore, routing and wavelength assignment algorithms which consider physical layer QoS (Quality of Service) demands, such as target BER (Bit Error Rate) levels, could prove interesting research topics. Fault tolerant RWA and restoration schemes for scheduled lightpaths demands are important issues which could also be addressed.

### ACKNOWLEDGMENT

The author would like to thank J. Kuri for sending the Perl script for generating test data, D. Kirovski for sending the source code of the *GGC* algorithm, and the anonymous referees for their insightful comments and helpful suggestions which significantly improved the quality and presentation of this paper.

### REFERENCES

- [1] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high-bandwidth optical WANs," *IEEE Trans. Commun.*, vol. 40, pp. 1171–1182, July 1992.
- [2] D. Banerjee and B. Mukherjee, "A practical approach for routing and wavelength assignment in large wavelength-routed optical networks," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 903–908, June 1996.
- [3] J. S. Choi, N. Golmie, F. Lapeyere, F. Mouveaux, and D. Su, "A functional classification of routing and wavelength assignment schemes in DWDM networks: Static case," in *Proc. VII Int. Conf. on Optical Communication and Networks*, Nagoya, Japan, Jan. 2000.

- [4] P. Manohar, D. Manjunath, and R. K. Shevgaonkar, "Routing and wavelengths assignment in optical networks from edge disjoint paths algorithms," *IEEE Commun. Lett.*, vol. 6, pp. 211–213, May 2002.
- [5] A. Ozdaglar and D. Bertsekas, "Routing and wavelength assignment in optical networks," *IEEE/ACM Trans. Networking*, vol. 11, pp. 259–272, Apr. 2003.
- [6] X. Jia, X.-D. Hu, and D.-Z. Du, *Multiwavelength Optical Networks*. Norwell, MA: Kluwer Academic Publishers, 2002.
- [7] B. Mukherjee, *Optical Communication Networks*. New York: McGraw-Hill, 1997.
- [8] C. Murthy and M. Gurusamy, *WDM Optical Networks: Concepts, Design, and Algorithms*. New Jersey: Prentice Hall, 2002.
- [9] J. Kuri, N. Puech, M. Gagnaire, E. Dotaro, and R. Douville, "Routing and wavelength assignment of scheduled lightpath demands," *IEEE J. Select. Areas Commun.*, vol. 21, pp. 1231–1240, Oct. 2003.
- [10] D. Kirovski and M. Potkonjak, "Efficient coloring of a large spectrum of graphs," in *Proc. 35th Conf. Design Automation*, San Francisco, CA, June 1998, pp. 427–432.
- [11] J. Kuri, N. Puech, and M. Gagnaire, "Diverse routing of scheduled lightpath demands in an optical transport network," in *Proc. Design of Reliable Communication Networks (DCRN2003)*, Oct. 2003, pp. 69–76.
- [12] C. Saradhi, L. Wei, and M. Guruswami, "Provisioning fault-tolerant scheduled lightpath demands in WDM mesh networks," in *Proc. Broadband Networks*, Oct. 2004, pp. 150–159.
- [13] F. Glover and M. Laguna, *Tabu Search*. Boston, MA: Kluwer Academic Publishers, 1997.
- [14] D. S. Johnson and M. A. Trick, Eds., *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, ser. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, 1993, vol. 26.
- [15] R. Brooks, "On coloring the nodes of a network," in *Proc. Cambridge Phil. Soc.*, vol. 37, 1941, pp. 194–197.
- [16] L. Stacho, "New upper bounds for the chromatic number of a graph," *J. Graph Theory*, vol. 36, no. 2, pp. 117–120, Feb. 2001.
- [17] D. Eppstein, "Finding the k shortest paths," *SIAM J. Computing*, vol. 28, no. 2, pp. 652–673, 1998.
- [18] J. Kuri, "Optimization problems in WDM optical transport networks with scheduled lightpath demands," Ph.D. dissertation, Ecole Nationale Supérieure des Télécommunications, France, Sept. 2003.
- [19] J. M. Kleinberg, "Approximation algorithms for disjoint paths problems," Ph.D. dissertation, MIT, Cambridge, May 1996.
- [20] N. Skorin-Kapov, "Routing and wavelength assignment in optical networks using bin packing based algorithms," *Europ. J. Op. Res.*, submitted for publication.
- [21] R. Floyd, "Algorithm 97: Shortest paths," *Comm. Of the AMC*, vol. 5, no. 6, p. 345, 1962.
- [22] R. Ramaswami and K. N. Sivarajan, "Design of logical topologies for wavelength-routed optical networks," *IEEE J. Select. Areas Commun.*, vol. 14, no. 5, pp. 840–851, June 1996.
- [23] S. Kirckpatrick and E. Stoll, "A very fast shift-register sequence random number generator," *J. Computational Physics*, vol. 40, pp. 517–526, 1981.
- [24] B. M. R. Inkret, A. Kuchar, *Advanced Infrastructure for Photonic Networks: Extended Final Report of COST Action 266*. Zagreb: Faculty of Electrical Engineering and Computing, University of Zagreb, 2003, pp. 19–21.



**Nina Skorin-Kapov** was born in Zagreb, Croatia, in 1981. She received her B.Sc. degree in telecommunications from the Faculty of Electrical Engineering at the University of Zagreb, Croatia, in 2003. She is currently working towards her Ph.D. degree at the Department of Telecommunications at the same university.

She has been working as a Research Assistant at the Department of Telecommunications, Faculty of Electrical Engineering, University of Zagreb, since 2003. Her research interests include heuristic algorithms, optimization in telecommunications (particularly in WDM wide-area optical networks), network routing algorithms and network topology design.