

Limiting Wavelength Converter Usage in Resilient WDM Networks

Miguel Razo*, Shreejith Billenahalli*, Wanjun Huang*, Arularasi Sivasankaran*, Limin Tang*, Hars Vardhan*, Paolo Monti†, Marco Tacca*, and Andrea Fumagalli*

* OpNeAR Lab, Erik Jonsson School of Engineering and Computer Science
The University of Texas at Dallas, Richardson, TX, USA

{mrazora, sxb071100, wxh063000, axs075200, lxt064000, hxv071000, mtacca, andrea}@utdallas.edu

† NeGONet Group, School of Information and Communication Technology, ICT-FMI
The Royal Institute of Technology, Kista, Sweden
pmonti@kth.se

Abstract—A careful wavelength assignment (WA) to lambda services must be performed to reduce the total number of wavelength converters (WCs) that are required when the wavelength continuity constraint cannot be met in wavelength division multiplexing (WDM) networks. With the successful introduction of reconfigurable optical add-drop multiplexers (ROADMs), WDM networks are now growing in size, both in the number of optical nodes and the number of wavelengths supported. Fast and memory efficient WA algorithms are required to design cost effective large WDM networks.

This paper presents a scalable and efficient WA heuristic algorithm aimed at reducing the total number of WCs that are required in (large) WDM networks bearing static lambda services. The WA algorithm is applied to both unprotected and (dedicated) protected lambda services. In the latter case, the wavelength continuity constraint between the working and protection path of a lambda service is taken into consideration when non-tunable optical transceivers are employed.

I. INTRODUCTION

End-to-end optical circuits may be established in optical transport networks to avoid electronic processing of transmitted data at the intermediate nodes. In WDM networks, an optical circuit is obtained by reserving a wavelength channel along the path from the source to the destination node. If possible, wavelength continuity is preferred when reserving the wavelength channel for the circuit, across all the fiber hops traversed by the circuit path. When the wavelength continuity constraint cannot be satisfied due to the already reserved wavelengths, a wavelength converter (WC) is necessary along the optical circuit, to shift the signal from the available wavelength in one fiber hop to the available wavelength in the next fiber hop. Due to their relatively high cost, WCs should be used minimally.

The problem of assigning wavelengths (WA) to optical circuits in some optimal way has been long studied. The min-RWA problem — optimal way of routing and wavelength assignment — is shown to be NP-hard [1]. The WA problem alone is shown to be equivalent to the graph coloring problem, under the assumption that WCs cannot be used [2]. Partition coloring may be applied to solving the WA problem sub-optimally [3]. Solutions are proposed to compute both routing and wavelength assignment (RWA) for unprotected lambda

services in order to minimize a given cost function [4], [5]. The cost function is user defined, and can be formulated to minimize the number of WCs required for each lambda service. Quite a number of other papers addresses the problem of minimizing the blocking probability of dynamically created optical circuits, for a given set of existing WCs in the network [6]. Recently, a number of solutions generalize both the WA and RWA problems to account for optical transmission impairments [7], [8], [9].

With the successful introduction of reconfigurable optical add-drop multiplexers (ROADMs) and related technologies, WDM network sizes are growing in the number of both optical cross-connect nodes and wavelengths supported. When facing the problem of minimizing the number of WCs that are required to support a number of static lambda services, the existing solutions [4], [10], [11] cannot handle large size networks. In fact these solutions make use of an auxiliary graph, whose number of vertices is given by the product of the number of nodes and number of wavelengths in the network. Another limitation of these solutions is they can only be applied to unprotected lambda services.

This paper presents a scalable and efficient WA algorithm that aims to reduce the total number of WCs in (large) WDM networks bearing a set of static lambda services. The WA algorithm is applicable to both unprotected and (dedicated) protected lambda services. In the latter case, the wavelength continuity constraint between the working and the protection path is taken into account when the employed transceivers are non-tunable. The proposed WA algorithm extends the coloring solution in [2] to take into account the possibility of using WCs. Given a set of precomputed routes for the lambda services, the WA algorithm runs in two steps. In the first step, the algorithm assigns one wavelength to every lambda service (both unprotected and protected) that can meet the wavelength continuity constraint. A graph coloring algorithm is used in this phase. The lambda services that cannot meet the wavelength continuity constraint are left *uncolored*. In the second step, every uncolored lambda service is assigned two or more wavelengths along with the necessary WCs to enable wavelength conversion. The problem of minimizing the

number of WCs in this second step is shown to be a modified version of the cover set problem. A heuristic is proposed to solve the modified cover set problem, which finally yields the WA and WC placement at optical nodes for every uncolored lambda service.

Due to its contained complexity, the proposed WA algorithm is applicable to WDM networks with up to 1500 optical nodes, and up to 160 wavelengths per fiber. One application of the proposed WA algorithm is illustrated in the paper while carrying out the study on the WC usage with and without the employment of tunable transceivers to support protected lambda services.

II. ASSUMPTIONS AND ALGORITHM DESCRIPTION

Let the optical transport network be modeled as undirected graph $G(N, E)$, where N is the set of nodes (representing network optical nodes), and E is the set of edges. Each edge represents a pair of fibers, one for each direction of propagation. Let each fiber carry up to W wavelengths. Let the given set of lambda service requests contain both a number of bidirectional unprotected services and a number of bidirectional (dedicated path) protected services. Every protected service requires two disjoint paths (edge, node or SRLG¹), i.e., a working path and a protection path. Assume that the routing for both the unprotected and the protected services is given. Bidirectional WCs may be required if a service (unprotected or protected) cannot be assigned the same identical wavelength value across every edge that belongs to the service path(s).

Protection switching is achieved in the optical layer, i.e., the signal is generated by the transmitter (laser) and optically switched to either the working or the protection path (1:1 protection)². The reverse procedure is used at the receiver, to collect the received signal from either the working or the protection path. Assume that the transmitter and receiver are both non-tunable. To avoid WC usage, the wavelength assigned to both the working and protection path must be the same, i.e., wavelength continuity constraint.

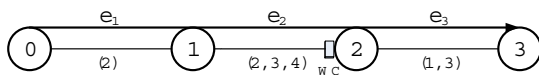


Fig. 1. Wavelength conversion along a path.

Fig. 1 depicts wavelength conversion along a path. The lambda service request is between node 0 and node 3, taking path 0-1-2-3. The unreserved wavelengths (colors) on each edge are shown in brackets, next to the edge. The wavelength continuity constraint cannot be met for this service. One solution is: color 2 is reserved on edges 0-1 and 1-2, and color 1 is reserved on edge 2-3. A WC is needed at node 2, to hop between color 2 and 1.

¹SRLG is an acronym for Shared Risk Link Groups.

²The proposed algorithm is described for the case of 1:1 dedicated path protection switching, however it can also be applied to the case of 1+1 dedicated path protection switching.

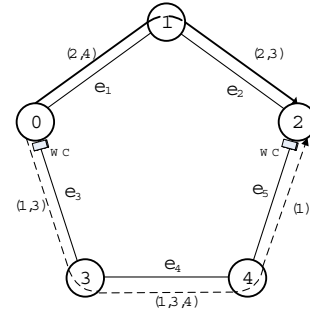


Fig. 2. Wavelength conversion at source/destination of a dedicated protected service.

Fig. 2 depicts wavelength conversion at the source (destination) of a dedicated protected service. The service request is between node 0 and node 2. The working path (solid arrow) is 0-1-2 and the protection path (dashed arrow) is 0-3-4-2. The unreserved wavelengths (colors) on each edge are shown in brackets, next to the edge. The wavelength continuity constraint cannot be met for this protected service. One solution is: color 2 is reserved on edges 0-1 and 1-2 (representing the working path), and color 1 is reserved on edges 0-3, 3-4, and 2-4 (representing the protection path). Note that by using tunable transceivers and dynamic wavelength switching, the 2 WCs required at the terminating nodes 0 and 2 can be avoided. This can be achieved by employing ROADM colorless-directionless architecture, where the add/drop ports are allowed to transmit (receive) on any wavelength and to (from) any direction [12].

Withstanding the above assumptions, the WA problem is formulated as follows: assign to each lambda service one unique wavelength (color) in every edge that belongs to the service path (both working and protection), with the objective of minimizing the total number of WCs, which are required in the network. To solve the WA problem in polynomial time, the following two-step heuristic is proposed.

Step 1. In the first step, as many lambda services as possible are assigned one single wavelength (color) across all of their edges. This is done by coloring a conflict graph with a technique similar to the one proposed in [2]. Every lambda service is associated with a node in the conflict graph. An edge is added between two nodes of the conflict graph if the paths of the two corresponding services share at least one edge in G . Note that if tunable transceivers are not available, both working and protection paths must be accounted for. If tunable transceivers are available, each protected lambda service is associated with two nodes in the conflict graph: one representing the working path and the other representing the protection path. In the latter case, working and protection paths are colored independently, as transceivers can be dynamically tuned while switching from one path to the other. Once the conflict graph is built, the coloring is computed using a graph coloring algorithm, such as the ones studied in [13]. If no more than W colors are required to color the conflict graph, no further step is necessary, as every node is colored with

an identifier that is within the W wavelengths. Hence, each service is assigned one wavelength and the algorithm stops here. No WCs are required in the network.

If more than W colors are required, colors assigned so far to the conflict graph are assumed to be *virtual* and are sorted as follows. Let S_i be the set of services, which are assigned color i . Let E_s be the set of edges that are in the path(s) of service s . Let ρ_i be the utilization of color i , defined as the size of the union of sets $E_s, \forall s \in S_i$. ρ_i counts the number of edges which have color i reserved for some service. Colors are then sorted by decreasing value of ρ_i . Ties ($\rho_i = \rho_j$) are resolved by counting the number of nodes assigned to each color, and ordering the colors by decreasing count. This sorting is placing first the virtual colors that either are used in many edges or are assigned to many services. Conversely, virtual colors that are placed last neither are used on many edges nor are assigned to many services. The first W virtual colors are then renamed to be color (wavelength) $1, 2, \dots, W$. These colors are final and they are used to permanently assign wavelengths to their corresponding services. None of these services require WCs. The remaining services, which are colored with the remaining bottom virtual colors ($W + 1, W + 2, \dots$), are said to be *uncolored* for now. It is easy to prove that none of the uncolored services can be assigned a single wavelength, unless one or more permanently colored services are unassigned their respective wavelengths. Services left uncolored require some WCs and are dealt with in step 2. A pseudo-code description of step 1 is given in Algorithm 1.

Algorithm 1 Wavelength Assignment Using Graph Coloring

- 1: Build conflict graph $C(V, L)$ with lambda services $\in V$
- 2: Add edge $l_{i,j}$ between V_i and V_j if their paths share one edge in G
- 3: Sort nodes $\in V$ in non increasing order of their nodal degrees
- 4: Initialize all the nodes to be uncolored
- 5: **for** each V_i in the sorted list **do**
- 6: $\text{maxColor} := 0$
- 7: **for** each of the neighboring nodes V_j already colored **do**
- 8: **if** $V_j.\text{color} > \text{maxColor}$ **then**
- 9: $\text{maxColor} = V_j.\text{color}$
- 10: **end if**
- 11: **end for**
- 12: $V_i.\text{color} = \text{maxColor} + 1$;
- 13: **end for**
- 14: Group all the nodes which are assigned the same color
- 15: **for** each color that has been assigned in the previous step **do**
- 16: Calculate the number of edges being used by all the nodes which are assigned the color and call it points
- 17: **end for**
- 18: Rank the colors based on maximum number of points
- 19: Break the tie of colors with same points by minimum number of nodes which as assigned the color
- 20: Nodes which are assigned the top W colors are permanently assigned a color equal to the rank of color
- 21: All the other nodes are left uncolored

Complexity Analysis of Algorithm 1: if the total number of lambda services is $|R|$, building the conflict graph requires $O(|R|)$ (line 1). Adding edges requires each fiber pair in the physical topology to be inspected. On each fiber

pair there could be up to W lambda services. Thus, adding edges requires $O(|E|W)$ (line 2). Sorting all nodes takes $O(|R| \log |R|)$ (line 3). The coloring algorithm (line 4 - 13) has complexity $O(|R|^2)$. Ranking the colors (line 14-21) requires $O(|R| + W|E| + W)$. So the complexity of Algorithm 1 is $O(|R|) + O(W|E|) + O(|R| \log |R|) + O(|R|^2) + O(|R| + W|E| + W) = O(W|E| + |R|^2)$. If $|R| \gg |E|$ or $|R| \gg W$, the complexity of Algorithm 1 is dominated by the coloring algorithm, which is $O(|R|^2)$.

Step 2. The problem of assigning two or more colors to uncolored lambda services, can be seen as the well known *Set Cover* (SC) problem [14], which is NP-Complete. The uncolored lambda services are considered individually, one at a time. The Universe U to be covered consists of the edges in G that belong to the working and (if any) protection path of the uncolored service. For each edge in U , define as *unreserved* the wavelengths (colors), which are not yet permanently assigned to any service. Define the i -th edge-subset S_{i,c_j} as the maximal subset of U , such that all the edges in S_{i,c_j} both have color c_j unreserved and constitute the edge set of a connected graph. (An example is shown in Fig. 3 and described at the end of this section.) Let F be the set of all possible edge-subsets.

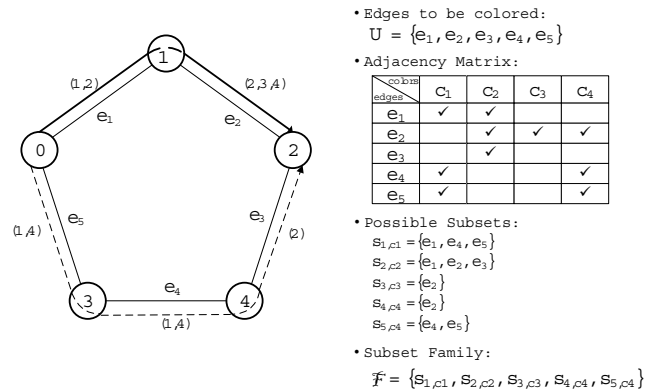


Fig. 3. Edge-subsets from unreserved colors on each edge.

The objective now is to find the minimum number of edge-subsets in F such that all the edges in U are contained in at least one of these subsets. It can be demonstrated that minimizing the number of edge-subsets — which are required to cover U — is equivalent to minimizing the number of WCs that are required for the service. To find a sub-optimal solution of the SC problem, a greedy approach is used, where at each iteration, the edge-subset with the largest number of uncolored edges is added to the cover solution. The color of the edge-subset is assigned to each of its uncolored edges. The procedure continues until no edge in the lambda service path is left uncolored. A pseudo-code description of step 2 is given in Algorithm 2.

Complexity Analysis of Algorithm 2: the number of uncolored lambda services after step 1 is at most $O(|R|)$. Each service has up to $O(|E|)$ edges. Therefore, Algorithm 2 has at most $O(|R|)$ iterations and each iteration needs to 1) sort edges by adjacency — $O(|E| \log |E|)$; 2) compute the edge-

Algorithm 2 Coloring lambda services using multiple colors

```

1: for each node  $V_i$  in graph  $C(V, L)$  not colored in Step 1 do
2:   Sort the edges of the node in the order of adjacency
3:   for each color from startColor to maxColors do
4:     Form the subset of edges which are adjacent and have this
       color available and assign the color to the subset
5:   end for
6:   EdgesColored := 0
7:   MaxEdges := Number of edges in node  $V_i$ 
8:   while EdgesColored < MaxEdges do
9:     Choose the subset with maximum number of edges which
       are not yet colored
10:    Assign the color of the subset to each of the edges which
       are not colored and the corresponding edge of the service
11:    Increment the EdgesColored by the number of edges which
       are assigned color above
12:   end while
13: end for

```

subsets, which requires to check all W colors on all $|E|$ edges — $O(|E|W)$; 3) assign a color to each edge — $O(|E|)$. Hence, the complexity of step 2 is $|R|(O(|E| \log |E|) + O(|E|W) + O(|E|)) = O(|R||E|(W + \log |E|))$.

Step 2 is the main contribution of the paper and its execution is illustrated with the help of Fig. 3. A protected service connects node 0 to node 2. The working path (solid arrow) takes route 0-1-2 and the protection path (dashed arrow) takes route 0-3-4-2. The unreserved wavelengths (colors) on each edge are shown in brackets, next to the edge. Wavelength continuity constraint cannot be met for this service, and step 2 is therefore required. Let e_1, e_2, e_3, e_4 and e_5 be the edges in U . The algorithm first sorts the edges based on adjacency. Starting from color c_1 , the edge-subsets are created by adding adjacent edges that have c_1 unreserved. Because of protection path, e_1 is considered to be adjacent to e_5 , which in turn is adjacent to e_4 . These three edges are grouped together to form subset S_{1,c_1} . Notice that color c_4 is unreserved in e_2, e_4 , and e_5 . As e_2 is not adjacent to e_4 or e_5 , the three edges are grouped to form two disjoint edge-subsets: S_{4,c_4} and S_{5,c_4} . After the computation of the edge-subsets, the greedy approach may choose either S_{1,c_1} or S_{2,c_2} , since both of them have the same number of edges. To break ties, first fit is used. Subset S_{1,c_1} is then chosen, and edges e_1, e_4 , and e_5 are colored with color c_1 . On the next iteration, S_{2,c_2} is chosen and e_2 and e_3 are colored using color c_2 . Note that e_1 is already colored as c_1 . All the edges in the service path(s) are now colored and the algorithm stops. In summary, the lambda service is assigned two wavelength values: wavelength 1 over edges e_1, e_4 and e_5 and wavelength 2 over edges e_2 and e_3 . Two bidirectional WCs are then required: one at node 4 and the other at node 1.

III. NUMERICAL RESULTS AND ANALYSIS

To investigate the performance of the proposed WA algorithm, a number of experiments are carried out using an Intel P4, 3.4 GHz machine with 1 GB memory. Numerical results are computed by averaging over ten experiments for number of request shown.

The fiber layout of the network is randomly generated as follows: edges are randomly added between nodes (avoiding parallel edges, i.e., two fibers on the same direction), till the desired average nodal degree is obtained. If the graph is not 2-connected, a new graph is generated till one with sufficient connectivity is found. The number of nodes ranges from $|N| = 10$ to $|N| = 1500$. The average nodal degree varies from 2.4 to 9.2. The number of lambda services is chosen to sufficiently load the network, i.e., WCs are required to fit all the services. The source-destination pair for each lambda service is chosen randomly using a uniform distribution over all of the possible node pairs.

The following solutions are compared. Solution CFZ-RWA is obtained running the RWA algorithm in [4]. Solution CFZ-R+WA is obtained using the routing solution found by solution CFZ-RWA, and solving the WA problem using the proposed WA algorithm. Solution R+WA is obtained using the simple sequential routing solution described next, and solving the WA problem using the proposed WA algorithm.

Simple Sequential Routing Algorithm. A simple sequential routing solution is used to compute the route for each lambda service, which is independent of the routing solution used in [4]. Thanks to its simplicity, this sequential routing solution is able to quickly compute routes even in large size networks. The computed routes are then used to test the scalability of the proposed WA algorithm in large networks.

The set of lambda services R and the topology description G are given as input. A time-varying cost is assigned to each edge to reflect its instantaneous load, i.e., the edge cost is incremented each time a new lambda service is routed through the edge. If the maximum capacity of the edge is reached (i.e., the number of lambda services using the edge is equal to W), the edge is removed from the graph. One at a time, the lambda services are routed using either shortest path (for unprotected) or Surballe's algorithm (for protected edge disjoint). As already mentioned, this simple routing technique is just one possible choice, and other efficient routing algorithms can be used to solve the routing problem.

A. Scalability

TABLE I
RESULTS CHANGING NETWORK SIZE AND TRAFFIC LOAD, $|W| = 40$.

N	E	CFZ-RWA		CFZ-R+WA		R+WA		
		Req.	WCs	WCs	Req.	WCs	Time[s]	
30	133	2966	61	24	2980	25	5	
40	179	3675	91	52	3682	51	7	
50	223	4296	171	101	4335	116	10	
55	246	4606	179.5	112	4646	112	12	
60	268	4926	245	155	4988	165	15	
75	335	-	-	-	5934	255	19	
100	446	-	-	-	7351	413	30	
150	666	-	-	-	10142	832	53	
200	890	-	-	-	12909	1343	61	
300	1357	-	-	-	18063	2564	108	
400	1789	-	-	-	23060	3932	174	
500	2231	-	-	-	24857	4781	254	
600	2676	-	-	-	32261	6860	351	
700	3123	-	-	-	36675	8281	470	
800	3577	-	-	-	41301	10032	594	
900	4029	-	-	-	45816	11659	725	
1000	4464	-	-	-	49972	13277	895	
1250	5572	-	-	-	60313	17543	1273	
1500	6705	-	-	-	69998	18177	1750	

Results shown in TABLE I are obtained by progressively increasing network size, in terms of number of nodes, edges, and lambda services (Req.), while keeping $W = 40$. Solution CFZ-RWA cannot be run successfully when the number of nodes is 65 or greater, due to the large size graph required to run the algorithm. For the same reason, solution CFZ-R+WA cannot be found for large networks as the routing for the lambda services cannot be obtained from solution CFZ-RWA. Solution R+WA runs with up to 1500 nodes. The last column in TABLE I shows the time (in seconds) required to solve the WA problem using the R+WA solution. It can be observed that the algorithm runs in reasonable time, even for large networks. Similar performance results were also obtained for $W=160$, which are discussed in [15].

B. Application

A useful application of the proposed algorithm is to quantify the trade-off between using tunable transceivers versus WCs in provisioning protected bidirectional lambda services. Assume that every lambda service in the network requires

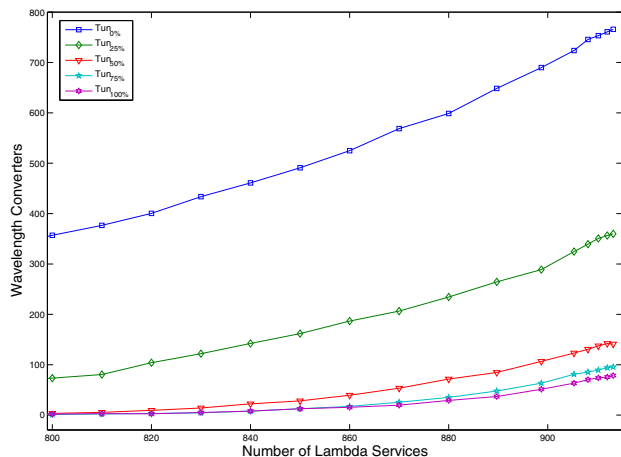


Fig. 4. Number of WCs as a function of the total number of dedicated protected lambda services. $|N| = 30$, $|E| = 103$, and $W = 40$.

one working path and one protection path, which are edge disjoint. Assume that a percentage of these lambda services will be provisioned using a pair of tunable transceivers, one at each end node. With the pair of tunable transceivers, it is possible to relax the wavelength continuity constraint across the working and protection path, as each path can be assigned an independent wavelength. This relaxation of the wavelength continuity constraint has a visible impact on the number of WCs, which are required in the network, as quantified in Fig. 4. The figure plots the number of WCs as a function of the number of protected lambda services in a $|N| = 30$ nodes and $|E| = 103$ edges network. Five curves are shown, assuming that the percentage of tunable transceiver pairs varies, from top to bottom, 0, 25, 50, 75, and 100. The trend reveals that a considerable percentage of WCs can be saved by using a fraction of tunable transceivers. For example, an average of 63.5% WCs can be saved when using 25% tunable

transceivers. An average of 89.6% WCs can be saved when using 50% tunable transceivers.

IV. SUMMARY

In this paper the authors present a scalable and efficient WA algorithm that aims to reduce the number of WCs required in (large) WDM network to support a given set of lambda services. The WA algorithm is applicable to both unprotected and (dedicated) protected lambda services. Due to its low complexity, the proposed WA algorithm is shown to be applicable to WDM networks with up to 1500 nodes, and up to 160 wavelengths per fiber [15]. Running on the same computer, comparable algorithms already available in the literature (e.g., [4]) cannot handle WDM networks with more than 65 nodes and 40 wavelengths, and can only be applied to handle unprotected lambda services.

REFERENCES

- [1] T. Erlebach, K. Jansen, and C. Elvezia, "The complexity of path coloring and call scheduling," *Theoretical Computer Science*, vol. 255, p. 2001, 2001.
- [2] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high bandwidth optical wans," *IEEE Transaction on Communications*, vol. 40, no. 7, pp. 1171–1182, July 1992.
- [3] G. Li and R. Simha, "The partition coloring problem and its application to wavelength routing and assignment," *In Proceedings of the First Workshop on Optical Networks*, 2000.
- [4] I. Chlamtac, A. Farago, and T. Zhang, "Lightpath (wavelength) routing in large wdm networks," *Selected Areas in Communications, IEEE Journal on*, vol. 14, no. 5, pp. 909–913, Jun 1996.
- [5] D. Banerjee and B. Mukherjee, "A practical approach for routing and wavelength assignment in large wavelength-routed optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 903–908, 1996.
- [6] X. Chu, B. Li, and Z. Zhang, "A dynamic rwa algorithm in a wavelength-routed all-optical network with wavelength converters," *IEEE INFOCOM*, pp. 1795–1804, 2003.
- [7] Y. Huang, J. P. Heritage, and B. Mukherjee, "Connection provisioning with transmission impairment consideration in optical wdm networks with high-speed channels," *Journal of lightwave technology*, vol. 23, no. 3, p. 982, 2005.
- [8] R. Martinez, F. Cugini, N. Andriolli, L. Valcarenghi, P. Castoldi, L. Wosinska, J. Comellas, and G. Junyent, "Challenges and requirements for introducing impairment-awareness into the management and control planes of ason/gmpls wdm networks," *IEEE communications magazine*, vol. 44, no. 12, p. 76, 2006.
- [9] G. Markidis, S. Sygletos, A. Tzanakaki, , and I. Tomkos, "Impairment aware based routing and wavelength assignment in transparent long haul networks," *Lecture notes in computer science*, vol. 4534 LNCS, p. 48, 2007.
- [10] S. Gowda and K. M. Sivalingam, "Protection mechanisms for optical wdm networks based on wavelength converter multiplexing and backup path relocation techniques," *IEEE INFOCOM 2003*, 2003.
- [11] Y. Wang, L. Li, and S. Wang, "A new algorithm of design protection for wavelength-routed networks and efficient wavelength converter placement," *IEEE ICC 2001*, vol. 6, pp. 1807–1811, 2001.
- [12] P. Roorda and B. Collings, "Evolution to colorless and directionless roadm architectures," in *Optical Fiber communication/National Fiber Optic Engineers Conference, 2008. OFC/NFOEC 2008. Conference on*, Feb. 2008, pp. 1–3.
- [13] H. Al-Omari and K. E. Sabri, "New graph coloring algorithms," *American Journal of Mathematics and Statistics*, vol. 2, no. 4, pp. 439–441, 2006.
- [14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*. The MIT Press, September 2001.
- [15] "A Scalable Wavelength Assignment Algorithm Using Minimal Number of Wavelength Converters in Resilient WDM Networks," The University of Texas at Dallas, Tech. Rep., September 2009. [Online]. Available: <http://opnear.utdallas.edu/publications/reports/>