Link Selection Algorithms for Link-Based ILPs and Applications to RWA in Mesh Networks

Zeyu Liu, George N. Rouskas
Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206, USA

***Abstract*—RWA is a fundamental problem in the design and control of optical networks. We propose link selection algorithms that reduce the size of the link-based ILP formulation for RWA by pruning redundant link decision variables. The resulting formulation scales well to mesh topologies representative of backbone and regional networks. In our experiments, the new formulation decreases the running time by more than two orders of magnitude without any impact on optimality. The link selection techniques are general in that they may be applied to any optimization problem for which the ILP formulation consists of multicommodity flow equations as its core constraints.**

I. Introduction

The global network infrastructure is built on a foundation of optical networking technologies, first deployed in the backbone and regional parts of the network but now also reaching into the access part in the form of PON architectures. Therefore, the planning and design of optical networks [14] is crucial to the operation and economics of the Internet and its ability to support critical and reliable communication services. In optical networks, traffic is carried over *lightpaths* that are optically switched at intermediate nodes. The routing and wavelength assignment (RWA) problem is one of selecting a path and wavelength for each connection demand, subject to certain constraints. RWA is a fundamental problem in the engineering, control, and design of optical networks, and arises in most network design applications, including traffic grooming [3], [5], survivability design [12], and traffic scheduling [9].

Offline RWA [7] is a network design problem in which the input typically consists of a set of traffic demands. Several variants of the problem have been studied in the literature, mainly differing in the objective pursued. In general, these problems are NP-hard [1], and several integer linear program (ILP) formulations have been proposed to solve them. Recently, we developed an exact decomposition approach for an ILP formulation based on maximal independent sets that makes it possible to obtain optimal solutions to the RWA problem for maximum size (i.e., 16-node) SONET rings in only a few seconds using commodity CPUs [16]. Unfortunately, current optimization methods cannot be used to solve optimally mesh network instances arising in practice. Consequently, many heuristic solution methods have been developed under various assumptions and network settings; e.g., refer to the surveys in [4], [17]. Nevertheless, the lack of scalability of optimal methods makes it difficult to characterize the performance of heuristic algorithms, and severely limits the application of "what-if" analysis to explore the sensitivity of network design decisions to forecast traffic demands, capital and operational cost assumptions, and service price structures. Currently, such analysis requires substantial investments in computational resources, time, and relevant expertise.

In this paper, we propose link selection algorithms for link-based ILP formulations that reduce the problem size by pruning redundant link decision variables. The link selection techniques are general in that they may be applied to any optimization problem for which the ILP formulation consists of multicommodity flow equations as its core constraints. By applying these techniques to the offline RWA problem, the resulting formulation scales well to mesh topologies representative of backbone and regional networks. In our experiments, the new formulation decreases the running time by more than two orders of magnitude without any impact on optimality.

The paper is organized as follows. In Section II, we review the complexity of the link-based formulation that is the starting point of our work. In Section III, we introduce two link selection algorithms to reduce the problem size. We present an experimental study to investigate the effectiveness of this approach in Section IV, and we conclude the paper in Section V.

II. Link ILP Formulation of the RWA Problem

Consider a connected graph $G = (V, A)$, where V denotes the set of nodes and A denotes the set of directed links (arcs) in the network. We define $N = |V|$ and $L = |A|$ as the number of nodes and links, respectively. Each directed link l consists of an optical fiber that may support W distinct wavelengths. Let $T = [t_{sd}]$ denote the traffic demand matrix, where t_{sd} is a non-negative integer representing the number of lightpaths to be established from source node s to destination node d. In general, traffic demands may be asymmetric, i.e., $t_{sd} \neq t_{ds}$. We also make the assumption that $t_{ss} = 0, \forall s$.

There are three classes of ILP formulations for the RWA problem depending on the types of variables used: (1) link-based [13], (2) path-based [11], or (3) maximal independent set (MIS)-based [13], [16]. A comparison of link and path based formulations was carried out in [8], while several RWA algorithms based on LP relaxations of such formulations were designed and studied in [2].

Path- and MIS-based formulations require the pre-selection of paths, hence they are suboptimal in mesh networks. There-

This work was supported in part by the National Science Foundation under Grant CNS-1113191.

fore, we focus on the link ILP formulation in which the entities of interest (i.e., decision variables) are link related. The link formulation is based on expressing the RWA problem as a multicommodity flow problem. Let us define the following sets of decision variables:

- $c_{sd}^{lw} \in \{0, 1\}$: binary variable that indicates whether there exists a lightpath from node s to node d that uses wavelength w on link l;
- $u^w \in \{0, 1\}$: binary variable that indicates whether wavelength w is used anywhere in the network; and
- ω_{total}: the number of wavelengths used in the network.

With these notations, the link ILP formulation can be expressed as:

$$Minimize: \omega_{total}$$

Subject to:

$$\sum_{\substack{\text{links } l \in L \\ \text{outgoing from } n}} c_{sd}^{lw} - \sum_{\substack{\text{links } l \in L \\ \text{incoming to } n}} c_{sd}^{lw} = \begin{cases} 0, n \neq s, d \\ t_{sd}, n = s \\ -t_{sd}, n = d \end{cases} \quad \forall n, s, d, w \tag{1}$$

$$\sum_{s,d} c_{sd}^{lw} \leq 1, \quad \forall l \in A, \forall w \tag{2}$$

$$\sum_{s,d} \sum_{l} c_{sd}^{lw} \leq u^w N(N-1)L, \quad \forall w \tag{3}$$

$$\omega_{total} \geq w u^w, \quad \forall w \tag{4}$$

$$u^w = 0, 1, \quad \forall w; \qquad c_{sd}^{lw} = 0, 1, \quad \forall s, d, l, w \tag{5}$$

Expressions (1) are the multicommodity flow equations at node n. Specifically, if n is an intermediate node in the path from some source s to some destination d, the traffic coming into n should be equal to the traffic going out of n, as such traffic is not dropped at, or originates from, this node; hence the left hand side of (1) must be equal to zero. If n is the source node s, the first sum of the left hand side is equal to the traffic t_{sd} to node d and the second sum is zero. Similarly, if n is the destination node d, the second sum of the left hand side is equal to t_{sd} and the first sum is zero. This set of constraints ensures that all traffic demands are satisfied. Moreover, they also take care of the wavelength continuity constraints: the right hand side of the equation is zero for any intermediate node n in the path from a source to a destination, ensuring that if traffic arrives at n on some wavelength, it will leave n on the same wavelength. Expressions (2) represent the distinct wavelength constraints of the RWA problem, such that no two connections share the same wavelength on one link. Expressions (3) make sure that u^w is set to 1 if wavelength w is used on any link by any connection. Expressions (4) count the number of used wavelengths by making ω_{total} equal to the index of the highest wavelength used. Expressions (5) are the integrality constraints for the decision variables.

The scalability of the link ILP formulation depends directly on its size. This size, in turn, is determined by the number of decision variables and constraints. The number of the c_{sd}^{lw} variables is equal to $N(N-1)LW$ for general topology networks. There are also W variables u^w, and the decision variable ω_{total}. Hence, the total number of variables in the link formulation for general network topologies is $O(N^2LW)$.

In terms of the number of constraints (ignoring the integrality constraints (5)), expressions (1) correspond to N^2LW constraints, expressions (2) yield LW constraints, expressions (3) consists of W constraints, as does expression (4). Overall, the formulation consists of $O(N^2LW)$ constraints.

Given that the size of the link ILP formulation grows as $O(N^2LW)$, it is not surprising that it cannot be applied directly to topologies representative of regional, national, or international backbone networks. In the next section, we present two link selection techniques that can be used to reduce the size of the formulation in terms of both the number of variables and the number of constraints. Our approach is general in the sense that it can be applied to the link-based ILP formulation of *any* multicommodity flow problem that includes constraints similar to (1), regardless of the exact form of the objective function or other constraints.

III. Link Selection Algorithms

Based on our discussion of the link-based formulation represented by expressions (1)-(5), it is clear that its size $O(N^2LW)$ is determined by the multicommodity flow equations (1). Specifically, there is one variable c_{sd}^{lw} for each link l of the network, even if such a link cannot be on the path from source s to destination d in any optimal solution to the problem. For instance, consider a network covering the continental United States. It is highly unlikely that an optimal solution to the RWA problem would route a lightpath between two cities in the western part of the country (e.g., from Los Angeles to Seattle) over a path that includes a link in the eastern part of the country (e.g., a link from Philadelphia to New York). Indeed, such a circuitous route is likely to require more resources (in this case, wavelengths) than necessary, resulting in a corresponding increase in the objective function. Instead, it is likely that in any optimal solution this lightpath will be routed along a more direct path consisting of links that are geographically located close to the western part of the country.

Based on this observation, we are interested in defining decision variables c_{sd}^{lw} only for links that are in the "neighborhood" of each source-destination pair (s, d). By eliminating a large number of these decision variables for each pair (s, d), the size of the formulation may be reduced significantly with corresponding gains in running time. Clearly, however, the elimination of decision variables restricts the solution space, potentially leading to suboptimal solutions. Therefore, one has to be careful in how to define the neighborhood of links of a source-destination pair.

We note that in the pure link-based ILP formulation, the neighborhood of each source-destination node pair is implicitly defined as the whole network, regardless of the relative location of each node. We believe that this is a crude approach that guarantees optimality at the expense of extremely high running time. At the other extreme, if the neighborhood is defined as the shortest path between a pair of nodes, the solution can be obtained in polynomial time (e.g., using Dijkstra's algorothm) but it is likely to be far from optimal. Therefore, we propose two parameterized algorithms for selecting the links on which the decision variables c_{sd}^{lw} are defined for each source-destination pair (s, d). In essence, each algorithm corresponds to a distinct method of defining the neighborhood of links for a given pair (s, d). Furthermore, the parameter of each algorithm can be tuned to select a desirable tradeoff between running time and optimality of the solution.

A. D-Thresh: Select links close to the source and destination nodes

The key idea of the D-Thresh selection algorithm is to select those links that are geographically close to both the source s and the destination d of a given lightpath demand, as these are the links that are more likely to be part of the optimal routes between the two nodes. Let $dist(s, d)$ denote the cost of the shortest path between nodes s and d. Let D be a distance threshold; D is a parameter of the algorithm. Let l be a link with end points i and j. Then, link l is considered as part of the neighborhood of the source-destination pair (s, d) if and only if the following expression is true:

$$dist(s, i) + 1 + dist(j, d) \leq dist(s, d) + D. \tag{6}$$

In other words, the above expression admits a link $l = (i, j)$ in the neighborhood if the path from s to d fromed by the concatenation of the shortest path from s to i, the link l and the shortest from j to d has a cost that is no larger than the cost of the shortest path from s to d plus the threshold D. The threshold D is used to limit the number of links to be considered as part of the neighborhood, and may be calibrated to strike a good balance between running time and optimality.

B. K-Path: Select links on the route of k-shortest paths

Note that the D-Thresh algorithm selects links for a source-destination pair (s, d) solely based on distance. However, the importance of a link is likely to also depend on its relative position with respect to the node pair (s, d), and in particular, on whether or not it lies on a shortest path from s to d. Hence we introduce the K-Path selection algorithm where K is a parameter, consisting of the following steps:

1) Use a k- shortest path algorithm to compute the first K shortest paths between s and d.
2) Define the neighborhood of links for the pair (s, d) as the union of the links in the K shortest paths.

Intuitively, the K-Path link selection algorithm has two main advantages over D-Thresh:

- K-Path takes into account both the distance of a link from the source and destination nodes and its relative location (i.e., whether it lies on a shortest path). Therefore, the algorithm tends to select higher quality links that are more likely to carry traffic in an optimal solution. The D-Thresh algorithm, on the other hand, may include certain links of low quality in that they are not situated on a promising path between the source and destination nodes.
- Since several shortest paths may share common links, the set of selected links expands relatively slowly as the value of parameter K increases. On the other hand, the D-Thresh algorithm will tend to include a set of links that is rapidly increasing as the value of parameter D increases from a low value, and will include all links as D approaches one-half the network diameter.

C. Modified Link-Based ILP Formulation

Once the neighborhood of links for each source-destination pair has been selected, we modify the formulation shown in expressions (1)-(5) as follows:

- we remove the variable c_{sd}^{lw} for each link l that is not in the neighborhood of node pair (s, d); and
- we remove a constraint (1) for a node n and node pair (s, d) whenever n is not an endpoint of a link l in the neighborhood of (s, d).

As a result both the number of decision variables and the number of constraints in the formulation decreases, resulting in a more compact size. Note that, for a given value of parameters K and D, the amount of reduction in formulation size increases rapidly with the size of the network due to the greater opportunity for link elimination.

Figure 1 illustrates the relative behavior of the two link selection algorithms, in terms of the increase in the number of decision variables c_{sd}^{lw}, as a function of the value of the parameter of each algorithm. The results shown in this figure were obtained on the 17-node, 52-link German network [6]. The figure plots the number of c_{ij}^{wl} variables for the D-Thresh and K-Path algorithms against the value of the corresponding parameter (i.e., D and K, respectively). Note that parameter D starts at zero, in which case the neighborhood of links for each source-destination pair includes only the links along the corresponding shortest path. Parameter K, on the other hand, starts at one; for this value of K, the K-Path algorithm only selects links along the shortest paths as well. Hence, the two curves start at the same number of decision variables.

As the value of parameter D increases, the number of decision variables included by the D-Thresh algorithm increases rapidly; for $D = 10$ the D-Thresh algorithm includes all links in the network for all source-destination pairsl; in other words, for $D = 10$ the resulting formulation is equivalent to the original formulation in expressions (1)-(5)[1]. Under the K-Path algorithm, the number of decision variables also increases but much more gradually. This behavior is consistent with the observation we made earlier that the K shortest paths share common links. When $K = 10$, we can see that the number of

[1]Recall that we define links l to be directional. Therefore, even though the diameter of the network is less than ten, it takes a value of $D = 10$ to include variables for all links and all source-destination pairs in the formulation.

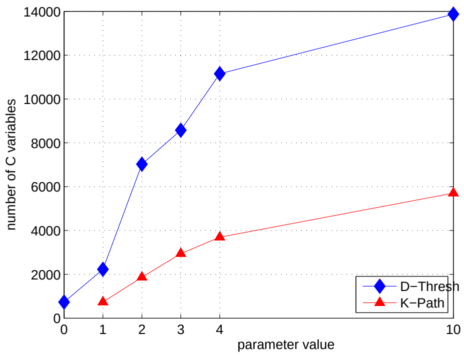


Fig. 1. Number of c_{sd}^{lw} variables vs. parameter value, German network

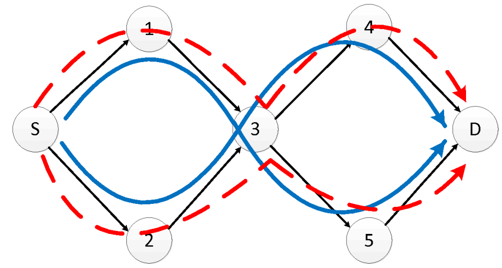


Fig. 2. An illustration on the advantage of k-path link selection algorithm

variables in the resulting formulation is less than one-half the total number of variables in the original formulation.

D. Comparison to Path-Based ILP Formulation

A path-based ILP formulation for the RWA problem is set up by pre-computing a set of K shortest paths between each pair of nodes, and only allows lightpaths to take these paths. A natural question that arises, then, is how a link-based formulation resulting from the K-Path selection algorithm, compares to the path-based formulation. The following lemma answers this question.

Lemma 3.1: Consider a set of K shortest paths for each source-destination pair, a path-based formulation set up on this set of paths, and a modified link-based formulation set up using the K-Path algorithm to select links for the c_{sd}^{lw} variables from the same set of paths. Then, the solution to the modified link-based formulation is no greater than the solution to the path-based formulation.

To see that the lemma is true, consider Figure 2 that illustrates a part of a network between a source s and destination d, including $K = 2$ shortest paths denoted by the solid (blue) lines. These two paths represent the solution space of the path-based formulation for demands from s to d. The modified link formulation, on the other hand, is set up with links, not paths, as the entities of interest. As a result, the solution space for demands from s to d within this formulation includes, in addition to the two shortest paths above, the two paths denoted by the dashed (red) lines, each consisting of one link from one of the shortest paths and one link from the other. In general, therefore, the solution space (in terms of paths) of the modified link-based formulation is larger than that of the path-based formulation, hence the optimal solution of the former cannot be worse than that of the latter. In the worst case, if the K paths are link-disjoint, the solution space of the two formulations will be exactly the same.

IV. Experimental Study

In this section, we present the results of an experimental study we conducted to investigate the performance of the new link-based formulation with wise link selection using the D-Thresh and K-Path algorithms. We are interested in two metrics: scalability (running time) and quality of solution. All results were obtained by running the IBM Ilog CPLEX 12 optimization tool on a cluster of identical compute nodes with dual Woodcrest Xeon CPU at 2.33GHz with 1333MHz memory bus, 4GB of memory and 4MB L2 cache.

Our study involves a large set of problem instances defined on several network topologies with random traffic matrices. In particular, we consider the following topologies (numbers refer to directed links): (1) the 11-node, 52-link Cost-239 network; (2) the 14-node, 42-link NSFNet [15]; (3) the 17-node, 52-link German network [6]; and (4) the 20-node, 78-link EON network [10]. These networks have irregular topologies of increasing size that are representative of existing backbone networks, and have been used extensively in optical networking research. For each topology, we generate the traffic demand matrix $T = [t_{sd}]$ by drawing the (integer) traffic demands (in units of lightpaths) uniformly at random in the interval $[0, T_{max}]$.

A. D-Thresh vs. K-Path

Figures 3(a) and (b) present the performance of the D-Thresh and K-Path algorithms, respectively, for problem instances generated on the German network topology with $T_{max} = 2$. The x axis of each figure represents the value of the corresponding algorithm parameter, D or K. The left y axis represents the solution value (i.e., number of wavelengths, ω_{total}) obtained by solving to optimality the ILP resulting for a given parameter value, while the right y axis represents the running time (in CPU seconds) for solving the ILP to optimality. Recall that, if the shortest path is unique (e.g., for nodes that are adjacent to each other), the ILP for $D = 0$ (under D-Thresh) is equivalent to the one for $K = 1$ (under K-Path), in that they only include decision variables c_{sd}^{lw} only for links on the shortest path between each pair of nodes. Hence, these results correspond to solutions that route each lightpath demand along the shortest path.

As we can see, under both link selection algorithms, the solution value drops rapidly as the parameter value (D or K) grows to two, and then remains constant indicating that optimality has been reached. Another observation is that the running time of the D-Thresh algorithm grows rapidly with the value of D; on the other hand, the running time of K-Path grows much slower with the value of K (note that the running time scales in the two figures are dfferent). Specifically, at

$D = K = 2$, the running time of the new link formulation resulting from D-Thresh is around 8000 sec, while that of the link formulation resulting from K-Path is around 300 sec – with no difference in solution quality. When $D = K = 10$, the running time with D-Thresh is about 64,000 sec or more than 17 hrs, while that with K-Path is around 2200 sec. This result is consistent with our discussion in the previous section.

Figures 4(a) and (b) are similar to the above two but show results for traffic matrices with $T_{max} = 6$. Since the total traffic demand is higher, the solution value and running time are higher than in the earlier figures. For the D-Thresh algorithm we provide results only up to $D = 2$, as for higher values the running time exceeds the 36-hour limit we imposed. The link formulation resulting from the K-Path algorithm, on the other hand, only takes around 3,600 seconds to solve to optimality when $K = 2$; for $K = 10$ the running time is around 30 hrs.

It is important to note the fact that for all the problem instances included in the above four figures, setting $D = K = 2$ is sufficient for the two link selection algorithms to reach the same optimal solution as the original link-based formulation. This result is generally true for all the network topologies we have investigated in our experimental study. Therefore, $D = K = 2$ strikes a good balance between solution quality and running time. Hence, for the results presented next we use $D = K = 2$.

B. Running Time Comparison to Original Formulations

Figure 5 compares the original link and path formulations to the link formulations produced by the D-Thresh and K-Path link selection algorithms for $D = K = 2$ in terms of running time, for the four network topologies listed above and $T_{max} = 2$.

We observe that, within the 20-hour time limit we imposed, the original link-based formulation can solve to optimality problem instances on the three smaller topologies, but not on the 20-node EON network. Using the D-Thresh algorithm, the resulting link formulation runs more than one order of magnitude faster, and can be used to solve the EON network within the time limit. The new link formulation resulting from the K-Path algorithm performs even better, reducing the running time by more than two-and-a-half orders of magnitudes compared to the original link formulation; in fact, it can solve all four network topologies within just a few hundreds seconds. It even outperforms the original path-based formulation (which is generally preferred to the link-based formulation as it is faster) by more than one order of magnitude.

C. Solution Quality Comparison

Finally, let us investigate the quality of the solutions obtained by the link formulation. Figure 6 plots the solution value obtained by the original link-based formulation, the path-based formulation, and the new link formulation under the K-Path algorithm ($K = 2$), as a function of T_{max} for the NSFNet; note that the solutions obtained by the link formulation under D-Path with $D = 2$ are identical to those shown here for K-Path. We observe that the new link formulation with link selection yields the optimal values obtained by the original link formulation. On the other hand, the solution obtained by the path formulation is suboptimal (i.e., 59 vs. 56). We have obtained similar results for the other three topologies, but they are omitted due to the page limit. These results indicate that the new link formulation with the K-Path link selection algorithm can be used to obtain optimal solutions to realistic problem instances in reasonable time, and outperforms the path formulation in both speed and solution quality.

V. Concluding Remarks

We have presented a new link formulation for the RWA problem that uses wise link selection to reduce significantly the formulation size. In our experiments, we have observed a more than two-and-a-half orders of magnitude improvement in running time compared to the original formulation without any impact on optimality. The new formulation outperforms the path-based formulation in terms of both running time and solution quality. More importantly, the wise link selection techniques may be applied to any optimization problems that can be formulated as an ILP with multicommodity flow equation as the core constraints.

References

[1] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: An approach to high bandwidth optical WANs. *IEEE Transactions on Communications*, 40(7):1171–1182, July 1992.

[2] K. Christodoulopoulos, K. Manousakis, and E. A. Varvarigos. Offline routing and wavelength assignment in transparent WDM networks. *IEEE/ACM Trans. Netw.*, 18(5):1557–1570, Oct. 2010.

[3] R. Dutta, A. E. Kamal, and G. N. Rouskas, editors. *Traffic Grooming in Optical Networks*. Springer, 2008.

[4] R. Dutta and G. N. Rouskas. A survey of virtual topology design algorithms for wavelength routed optical networks. *Optical Networks*, 1(1):73–89, January 2000.

[5] R. Dutta and G. N. Rouskas. Traffic grooming in WDM networks: Past and future. *IEEE Network*, 16(6):46–56, November/December 2002.

[6] W.D. Grover and D.P. Onguetou. A new approach to node-failure protection with span-protecting p-cycles. In *Proc. of ICTON 2009*.

[7] B. Jaumard, C. Meyer, and B. Thiongane. ILP formulations and optimal solutions for the RWA problem. In *Proc. of GLOBECOM'04*, vol. 3, pp. 1918–1924, Nov. 29-Dec. 3 2004.

[8] B. Jaumard, C. Meyer, and B. Thiongane. Comparison of ILP formulations for the RWA problem. *Optical Switching and Networking*, 3-4:157–172, 2007.

[9] J. Kuri, N. Puech, M. Gagnaire, E. Dotaro, and R. Douville. A review of routing and wavelength assignment of scheduled lightpath demands. *IEEE J. Selected Areas Comm.*, 21(8):1231–1240, October 2003.

[10] M. O'Mahony, D. Simeonidou, A. Yu, and J. Zhou. The design of the European optical network. *Journal of Lightwave Technology*, 13(5):726–733, May 1995.

[11] A. E. Ozdaglar and D. P. Bertsekas. Routing and wavelength assignment in optical networks. *IEEE/ACM Trans. Netw.*, 11(2):259–272, Apr. 2003.

[12] S. Ramamurthy and B. Mukherjee. Survivable WDM mesh networks, part I – protection. In *Proc. INFOCOM '99*, pp. 744–751, Mar. 1999.

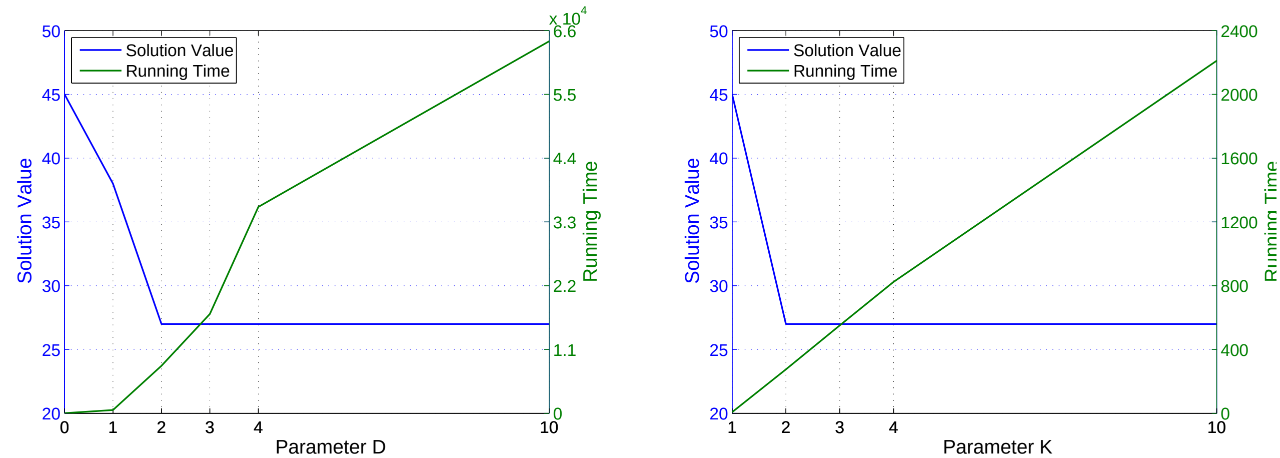
[13] R. Ramaswami and K. Sivarajan. Routing and wavelength assignment in all-optical networks. *IEEE/ACM Trans. Netw.*, 3(5):489–500, Oct. 1995.

[14] J. M. Simmons. *Optical Network Design and Planning*. Springer, 2008.

[15] B. Mukherjee *et al.* Some principles for designing a wide-area WDM optical network. *IEEE/ACM Trans. Netw.*, 4(5):684–696, Oct. 1996.

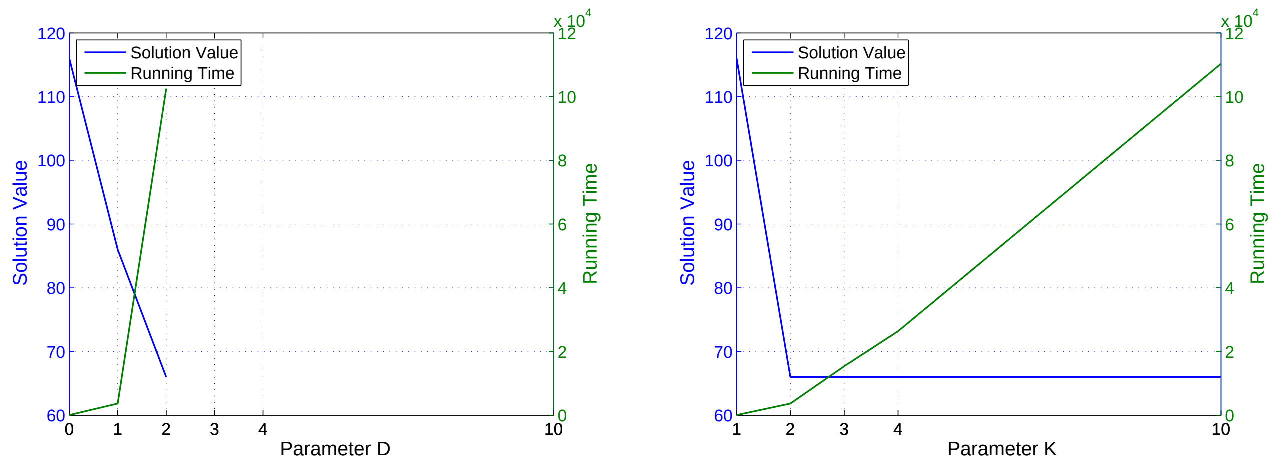
[16] E. Yetginer, Z. Liu, and G. N. Rouskas. Fast exact ILP decompositions for ring RWA. *Journal of Optical Communications and Networking*, 3(7):577–586, July 2011.

[17] H. Zang, J. P. Jue, and B. Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *Optical Networks*, 1(1):47–60, January 2000.



(a) D-Thresh algorithm (b) K-Path algorithm

Fig. 3. Running time (sec) and solution value (wavelengths) vs. parameter value, German Network with $T_{max} = 2$



(a) D-Thresh algorithm (b) K-Path algorithm

Fig. 4. Running time (sec) and solution value (wavelengths) vs. parameter value, German Network with $T_{max} = 6$

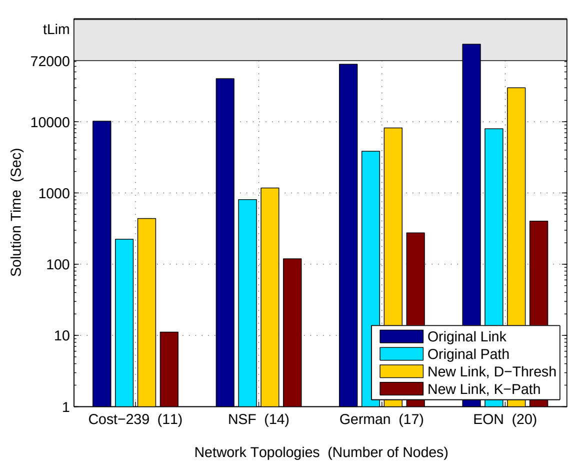


Fig. 5. Running time vs. network topology

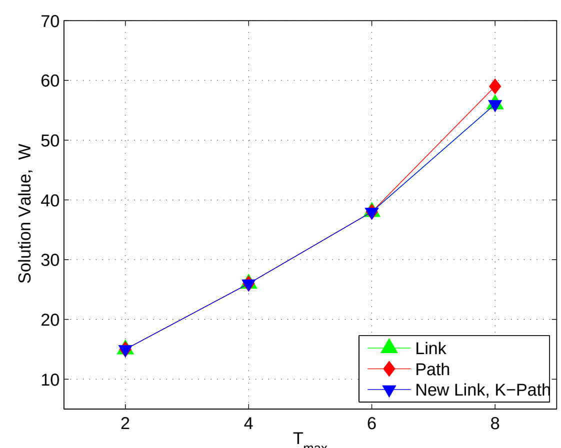


Fig. 6. Solution value against T_{max}, NSFNet