A Universal Spectrum Symmetry-Free Algorithm for Routing and Spectrum Assignment (RSA)

George N. Rouskas, Mithil Ghinaiya
North Carolina State University

***Abstract*—We present the first spectrum symmetry-free model for the routing and spectrum assignment (RSA) problem. This model allows for the design of more efficient algorithms as it eliminates from consideration an exponential number of equivalent symmetric solutions. By sidestepping symmetry, the RSA solution space is naturally and optimally decomposed into a routing space and a connection permutation space. Building upon this property, we introduce a two-parameter, symmetry-free algorithm that is *universal* in that it can be used to tackle any RSA variant in a uniform manner. The algorithm is amenable to multi-threaded execution to speed up the search process and the value of the parameters can be adjusted to strike a balance between running time and solution quality. Our evaluation provides insight into the relative benefits of path diversity (which determines the size of the routing space) and connection diversity (which determines the size of the permutation space).**

I. Introduction

Routing and spectrum assignment (RSA) and its variants are fundamental problems in the design and control of elastic optical networks [1]–[5], and underlie a range of optimization problems including virtual topology design [6], traffic grooming [7], [8] and network survivability [9]. The RSA problem is NP-hard [3], even in simple network topologies [10], and a wide range of integer linear programming (ILP) formulations have been developed to tackle it [4], [11]–[13].

Conventional ILP formulations for RSA face a significant challenge due to *spectrum symmetry*, i.e., the fact that spectrum slots are *interchangeable* [14]. As explained in [15] with reference to the RWA problem, "as the wavelengths are interchangeable, given an optimal solution of the RWA problem or of one of its continuous relaxation, one can derive a large number of equivalent solutions using any permutation of the wavelengths." In other words, ILP solvers must evaluate an exponential number of distinct but equivalent optimal solutions and hence their running time can be unnecessarily long [15].

Recently, we showed that first-fit (FF) is a *universal algorithm for all known variants of the spectrum assignment (SA) problem* (i.e., when routing is fixed and not part of the optimization) [16]. Specifically, FF (in its pure form or modified to account for variant-specific constraints) may be used to 1) construct a solution equivalent to, or better than, any solution obtained by any other algorithm, and 2) construct an optimal solution. This universality property implies that, to find an optimal solution to any SA problem variant, it is sufficient to consider only the connection permutations. Based on this insight, we introduced the first *symmetry-free model* for the spectrum assignment (SA) problem on networks of general topology [14]. Our model completely eliminates symmetric solutions from consideration and allows for the design of efficient SA algorithms.

In this work we extend the spectrum symmetry-free model to the RSA problem and make several contributions. First, we show that by eliminating symmetric solutions, the solution space naturally and optimally decomposes into a routing space and a connection permutation space. Second, we introduce a universal two-parameter RSA algorithm that explores exhaustively a subset of the solution space that encompasses the complete solution space for a subset of the connections. The algorithm can be readily implemented as it makes use of the well-known First-Fit (FF) algorithm [17]; it is also *universal* in that it can be applied to any RSA variant by appropriately modifying the FF algorithm as we discuss in [16]. Third, the size of the solution space can be adjusted to strike a balance between running time and solution quality. Finally, our study provides insight into the relative benefits of path diversity (which determines the size of the routing space) and connection diversity (which determines the size of the permutation space).

The remainder of the paper is organized as follows. In Section II, we define the RSA problem that we consider in this work and introduce the spectrum symmetry-free model. In Section III we present the universal two-parameter algorithm for the RSA problem. We evaluate the algorithm in Section IV, and we conclude the paper in Section V.

II. Spectrum Symmetry-Free RSA

Consider an optical network with topology graph $G = (V, A)$, where V is the set of nodes and A is the set of directed fiber links. Let $N = |V|$ be the number of nodes and $L = |A|$ be the number of links. We are given a set $\mathcal{T} = \{T_i\}$, of C connections, such that each connection is a tuple $T_i = (s_i, d_i, t_i, \mathcal{P}_i), i = 1, \cdots, C$, where: s_i and d_i are the source and destination nodes, respectively, of the connection, t_i is the amount of spectrum (e.g., in units of spectrum slots) required to carry the traffic from s_i to d_i, and $\mathcal{P}_i$ is a set of k paths $\{p_i^{(1)}, \cdots, p_i^{(k)}\}$ between nodes s_i and d_i. Unlike earlier research that assumes a small number C of

This work was supported by the National Science Foundation under Grant CNS-1907142.

connections, in this paper we consider connections between all node pairs in the network and let $C = N(N-1)/2$.

We assume that k is a small integer and that the k paths of a connection are pre-determined. The paths of a connection may be calculated as the k shortest paths between the particular source-destination pair, or using any other desirable criteria. Further, we assume that splitting the spectrum demand of a connection over multiple paths is not allowed.

We consider the following general definition of the routing and spectrum assignment (RSA) problem[1]:

Definition 2.1 (RSA): For each connection T_i, select one of the physical paths $p_i^{(1)}, \cdots, p_i^{(k)}$ and assign t_i spectrum slots along this path so as to 1) optimize spectrum assignment (e.g., under the min-max or min-frag criteria of [16]), and 2) satisfy the contiguity, continuity, non-overlap, and other variant-specific constraints (e.g., guard-band constraints [16]).

It is well known that the RSA problem is intractable [4], [17]. Moreover, conventional algorithms for the RSA problem must unnecessarily tackle an exponential number of symmetric and equivalent solutions. We now show how our recent work on symmetry-free solutions to the spectrum assignment (SA) problem may be extended to the more general RSA problem. This discussion provides the motivation for the new, universal algorithm we present in Section III.

An RSA solution (optimal or not) determines one of the k paths for each connection. Let us define a *routing configuration* R_j, as an assignment of one path to each connection, whereby the path assigned to connection T_i is selected among the set $\mathcal{P}_i$ of k paths input to the RSA problem. Then, an RSA solution encompasses selecting one of the k^C routing configurations.

Consider a specific routing configuration R_j. When the path of each connection is fixed to the one specified in R_j (i.e., routing is not subject to optimization), the RSA problem reduces to the spectrum assignment (SA) problem. While simpler, the SA problem is also intractable in general topologies [10].

In recent work [14], [16] we introduced the first symmetry-free model for the SA problem in general topologies. Specifically, we showed that to solve optimally *any* SA problem variant, it is sufficient to examine only the solutions obtained by applying the FF algorithm[2] to each of the $C!$ permutations of the C input connections. We also developed Recursive First-Fit (RFF) [14], [18], an optimal, parallel, branch-and-bound algorithm that recursively searches the permutation space and applies the (pure or modified) FF algorithm incrementally as it builds the various permutations. While the size of the solution space (i.e., the number of permutations) is exponential, to the best of our knowledge, our approach is the first that completely eliminates from consideration the exponential number of *additional* symmetric solutions (i.e., solutions that cannot be the result of the FF algorithm).

We now make the observation that the various routing configurations of the RSA problem are pairwise independent. Therefore, our work in [14], [18] can be naturally extended to a spectrum symmetry-free algorithm for the RSA problem:

> *Run the RFF algorithm on all routing configurations and return the routing configuration and connection permutation that results in the best solution.*

In essence, the solution space of the RSA problem is optimally decomposed into a routing space of k^C routing configurations, and a permutation space of $C!$ permutations that must be explored separately for each routing configuration. While the above algorithm completely sidesteps all symmetric solutions, it amounts to an exhaustive search of a combined solution space of size $k^C \times C!$. Assuming that there is a connection between each node pair in the network, then $C = O(N^2)$ and both the number of routing configurations and the number of connection permutations are exponential in the size of the network. Therefore, it is prohibitive to search exhaustively the combined solution space for network topologies encountered in practice.

In the following, we present a solution approach that performs an exhaustive search only on a part of the combined solution space whose size can be calibrated appropriately. Importantly, the algorithm may be applied directly to all RSA variants with only minor modifications to the underlying FF algorithm, depending on the variant-specific spectrum constraints [16].

III. A Universal Symmetry-Free RSA Algorithm

Our goal is to bridge the gap between an exhaustive search of the entire solution space, which is prohibitively expensive computationally for deployed networks, and greedy heuristic approaches, by introducing a parameterized approach that achieves a desirable tradeoff between running time and quality of solution. Our universal algorithm for RSA problem variants is characterized by two parameters:

1) the number $c < C$ of high-priority connections, and
2) the number k of paths for each high-priority connection.

Network designers may apply any appropriate criteria to decide which connections are included in the high-priority set. For instance, connections may be characterized as high-priority based on: 1) the size of their demands, 2) the distance between their endpoints, 3) a measure of the importance of the traffic they carry, 4) the revenue they produce, or 5) any combination thereof. Each high-priority connection is provided with $k > 1$ alternate paths; each low-priority connection, on the other hand, may use only a single (fixed) path. Any appropriate routing algorithm and link weights may be used to generate these paths.

Consider an RSA problem with C connections and define the *high-priority subproblem* as one that includes only the

[1] In this definition we only consider spectrum-specific constraints. We assume that any routing-specific constraints (e.g., in terms of path length), are applied in determining the paths passed as input to this RSA problem.

[2] As we explained in [16], the FF algorithm must be appropriately modified to account for any variant-specific constraints beyond the contiguity, continuity, and non-overlap constraints, but such modifications are straightforward.

$c < C$ high-priority connections (i.e., the subproblem created by eliminating the $C - c$ low-priority connections from the original problem). The solution space for this high-priority subproblem is the combination of the routing configurations (of size k^c) and permutations (of size $c!$) for the c high-priority connections. Since c and k are small integers selected by the network designer, the size of this solution space can be considered as fixed and can be carefully adjusted to match the available computational resources.

Our approach to tackling the original RSA problem with C connections is to explore exhaustively a subset of its solution space that encompasses the entire solution space of the high-priority subproblem. This strategy collectively optimizes the allocation of resources to the connections that the network designer regards as important. For instance, let us assume that priority is proportional to the demand size and/or distance of a connection. Intuitively, connections with large demands or that travel long distances require correspondingly large resources. Hence, their path and spectrum must be optimized, not only individually but in combination with other such connections, to ensure that network resources are allocated efficiently. Furthermore, a small fraction of all connections may account for a considerable fraction of total demand (refer to Section IV), so that exploring the entire solution space of such connections may be computationally feasible.

Figure 1 provides a pseudo-code description of the symmetry-free RSA algorithm with parameters (k, c). The preprocessing step of the algorithm constructs the solution space to explore. Specifically, Steps 1-5 generate all $c!$ permutations of high-priority connections and extend each (at Step 4) by appending the $C - c$ low-priority connections in a fixed order to create $c!$ permutations of all C connections. Similarly, Steps 6-10 generate all k^c routing configurations of high-priority connections and extend each (at Step 9) with the single path of each low-priority connection to create a routing configuration for all C connections. Finally, the main algorithm in Steps 11-19 exhaustively searches this solution space by applying the FF algorithm to each combination of routing configuration/permutation. Since each such combination is independent of any other, the execution of Steps 11-15 may be easily parallelized by 1) partitioning the combinations into pair-wise disjoint and collectively exhaustive subsets, and 2) deploying multiple threads running concurrently, each thread working on a different subset.

The preprocessing step takes time $O(c! + k^c)$, but since c and k have small integer values determined by the network designer, this step can be considered as taking a fixed amount of time. Note that a network designer may have to solve multiple instances for a given RSA problem defined by the network topology $G = (V, A)$ and number of connections C; for instance, this may be due to carrying out a "what-if" analysis to explore the sensitivity of design decisions to forecast traffic demands. In this case, the designer only needs to perform the preprocessing step once, store the permutations and routing configurations, and use them to solve all instances that are part of the analysis. Therefore, the computational cost of this step can be amortized over multiple problem instances.

The main part of the algorithm in Lines 11-15 simply runs the FF algorithm on each of the $M = k^c \times c!$ combinations of permutations and routing configurations generated in the preprocessing step. Each application of the FF algorithm takes time $O(CL)$, as each permutation consists of C connections and each connection may involve any of the L links in the network. Therefore, the total running time of this part of the algorithm is $O(CLM)$, where M is again considered as having a fixed value.

Note that parameter k (respectively, c) represents the degree of path (respectively, connection) diversity. Each parameter independently controls the size of the solution space of the routing subproblem and spectrum allocation subproblem, respectively, hence the family of algorithms represents a wide spectrum of RSA solution strategies. Specifically, 1) when $c = 0$, the algorithm reduces to FF as it considers a single path for all connections and one permutation; 2) when $c = C$ and $k = 1$, it reduces to the RFF algorithm [14] and, given sufficient time to run, it explores all connection permutations on a single routing configuration, and 3) when $c = C$ and $k > 1$, it is an extension to RFF that, given sufficient time to run, is optimal for the given set of routing paths. By carefully selecting values for the two parameters k and c, a network designer may strike a desirable balance between the running time and the quality of the final solution.

Finally, we emphasize again that the algorithm in Figure 1 is applicable to any variant of the RSA problem, not just the basic variant of Definition 2.1. For instance, the k paths may be calculated so as to take into account reach, various available modulation formats [19], intra- or inter-core crosstalk [20], etc. Additional constraints may eliminate some of the routing configurations, reducing the effective size of the routing space well below k^c and, thus, allowing for larger values for parameters k and/or c. Due to page limitations, however, the study in the next section only focuses on the basic RSA problem with just the contiguity, continuity, and non-overlap constraints.

IV. Simulation Study

Recall from Section III that when $c = C$ and $k > 1$, the RSA algorithm in Figure 1 is optimal for the given set of paths as it examines all combinations of routing configurations and connection permutations. In practice, however, it would not be possible to search the entire solution space for anything but toy networks. Importantly, we expect that as the values of parameters c and k increase, the incremental improvement in solution quality will drop off due to the diminishing returns of considering low-priority (e.g., small) demands and long, circuitous paths. Therefore, our objective is to investigate the relative benefits of increasing path diversity (i.e., value of k) and connection diversity (i.e., value of c) on solution quality.

For this simulation study we create RSA problem instances characterized by two parameters: the network topology and the distribution used to generate random traffic demands. We use

Algorithm 1 Universal Spectrum Symmetry-Free RSA

Input:
$G = (V, A)$: network topology
C: number of connections
c: number of high-priority connections
k: number of paths for each high-priority connection
$\mathcal{T} = \{T_i = (s_i, d_i, t_i, \mathcal{P}_i)\}$: set of connections

Output:
$BestSOL$: RSA solution

SymFree-RSA(k, c)
Preprocessing
1: q: List of $C - c$ low-priority connections in decreasing priority;
2: Generate all $c!$ permutations q_l of the c high-priority connections;
3: **for** $l = 1; l \leq c!; l$++ **do**
4: $\quad Q_l \leftarrow$ Append q to q_l (permutation of all C connections);
5: **end for**
6: r: routing configuration with single path for the $C - c$ low-priority connections;
7: Generate all k^c routing configurations r_j for the c high-priority connections;
8: **for** $j = 1; j \leq k^c; j$++ **do**
9: $\quad R_j \leftarrow$ Append r to r_j (routing configuration for all C connections);
10: **end for**
Main Algorithm
11: **for** $j = 1; j \leq k^c; j$++ **do**
12: $\quad$ **for** $l = 1; l \leq c!; l$++ **do**
13: $\quad\quad SOL \leftarrow$ solution obtained by FF on routing configuration R_j and permutation Q_l
14: $\quad\quad$ **if** $SOL < BestSOL$ **then**
15: $\quad\quad\quad BestSOL = SOL$;
16: $\quad\quad$ **end if**
17: $\quad$ **end for**
18: **end for**
19: return BestSOL;

---

two network topologies, the 14-node, 21-link NSFNET and the larger 32-node, 54-link GEANT2 network, and for each topology we generate connections between all node pairs in the network as follows. We consider data rates of 10, 40, 100, 400, and 1000 Gbps. For a given problem instance, we generate a random value for the demand between a pair of nodes based on one of three distributions:

- *Uniform*: each rate is selected with equal probability;
- *Skewed low*: the rates above are selected with probability $0.30, 0.25, 0.20, 0.15$, and 0.10, respectively; or
- *Skewed high*: the five rates are selected with probability $0.10, 0.15, 0.20, 0.25$, and 0.30, respectively.

Then, we determine the number of spectrum slots that each demand requires based on its data rate and path length by assuming a slot width of 12.5 GHz and adopting the parameters of [2]. For each topology and traffic distribution we generate 100 random problem instances, for a total of 600 instances for this study.

We run the experiments on the Henry2 Linux HPC cluster at NC State University [21] which consists of more than 1,000 compute nodes and over 10,000 cores. We deployed $R = 32$ parallel threads, the maximum number available to us on the Henry2 cluster, to run the main steps 11-15 of the algorithm in Figure 1. In the experiments, we vary the number of paths, $k = 2, \cdots, 7$, and the number of high-priority connections, $c = 1, 2, \cdots, 7$. We also impose a running time limit of 1,000 seconds for each problem instance; hence, as shown in the figures below, any combination of (k, c) values for which the running time would exceed this time limit is not considered. For the algorithm of Figure 1, we select the c high-priority connections as those with the largest demands, and we use the depth first search (DFS) algorithm to calculate the k shortest paths for the high-priority connections.

The performance measure we consider is the maximum number of spectrum slots on any network link. For a given routing configuration R_j, a lower bound for this metric for an RSA problem instance can be calculated by ignoring the problem constraints and simply adding up the demands along each link and taking the maximum value over all links. Let *SPLB* denote this lower bound under shortest path routing, i.e., for the routing configuration consisting of the shortest path for all C connections. To make the results comparable across problem instances and (k, c) values, we normalize each solution SOL returned by the algorithm by taking the ratio

$$h = \frac{SOL - SPLB}{SPLB}. \tag{1}$$

The figures in this section plot this ratio which represents the normalized difference between SOL and $SPLB$. Each data point in the figures is the average of h over the 100 problem instances for the stated topology and traffic distribution. However, we emphasize that *SPLB does not represent a lower bound for RSA algorithms that use two or more paths for some connections.* As will see in a moment, with increasing path diversity the algorithm finds solutions that are *better* (i.e., lower) than the shortest path lower bound SPLB; in such cases, the value of h is negative. Nevertheless, we use the SPLB value for normalization because 1) it is a well-understood baseline quantity, and 2) it provides insight into the improvement that is possible with increasing path diversity.

A. Results and Discussion

Figure 1 plots the values of h (as a percentage) for the NSFNET topology and the skewed low distribution[3] as a function of the number c of high-priority connections. There are six curves in the figure, each corresponding to a number

[3]Due to page constraints, we only include results for one (different) traffic distribution for each of the two networks; however, the trends for the other two distributions are very similar to the ones in the figures we present here.

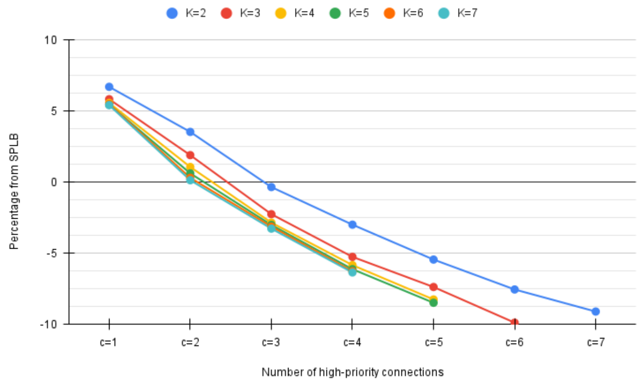

Fig. 1. Solution quality as function of c, NSFNET, skewed low distribution

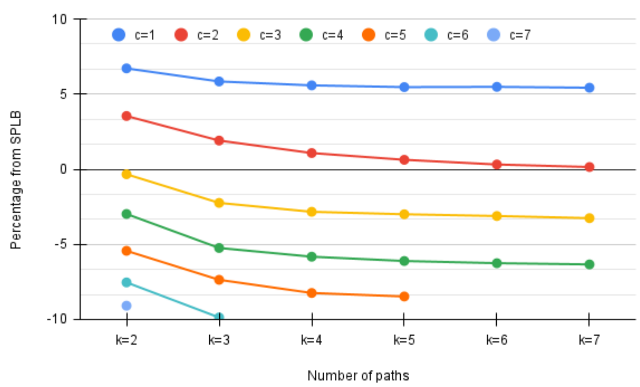

Fig. 2. Solution quality as function of k, NSFNET, skewed low distribution

$k, k = 2, \cdots, 7$ of alternate paths for the c high-priority connections (the $C - c$ low-priority connections are routed along their minimum-hop path). We observe that the solution quality of all curves improves significantly with c; for instance, with $k = 2$ paths, the solutions obtained by the algorithm improve, on average, by 16% relative to SPLB, i.e., from a value about 6.5% *above* SPLB to a value about 9.5% *below* SPLB. As expected, the curves start to level off after a while, depending on k, but it appears that there is room for further improvement as c increases beyond 7 (recall that we did not run experiments with $c = 8$ or higher as the running time would exceed our time budget of 1000 sec). We also observe that there is an increase in quality across all values of c as we move from $k = 2$ to $k = 3$ paths, but further increases in the number of alternate paths have little benefit.

Figure 2 presents the same results as Figure 1, but plots them instead as a function of k. This figure more clearly shows that 1) most benefits of path diversity are realized as soon as the number k of paths is 3 or 4, and further increases have little impact; and 2) the marginal gain in solution quality from incrementing connection diversity (i.e., c) is considerably higher than that from incrementing path diversity (i.e., k); even so, the effect of diminishing returns as c increases is also clear.

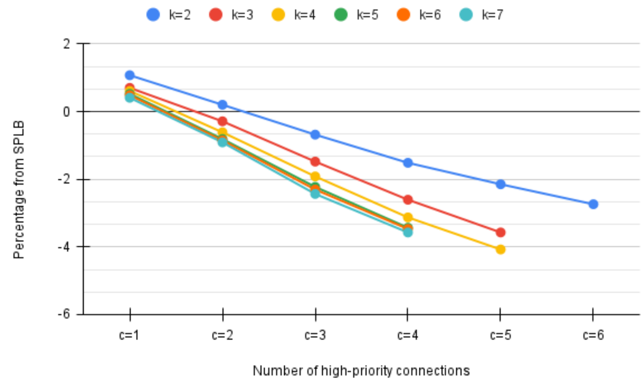

Fig. 3. Solution quality as function of c, GEANT2, skewed high distribution

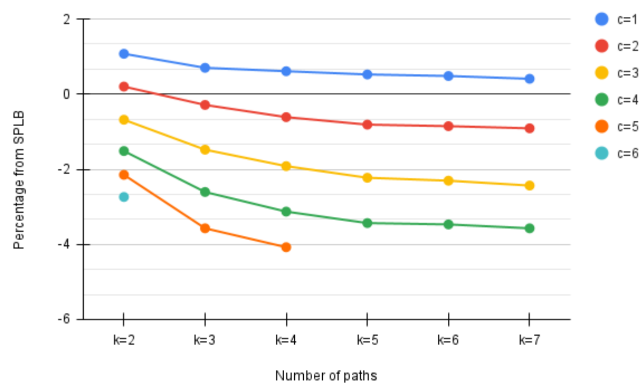

Fig. 4. Solution quality as function of k, GEANT2, skewed high distribution

Figures 3 and 4 are similar to Figures 1 and 2, respectively, but present results for the GEANT2 network and the skewed high distribution. We observe similar trends as for the NSFNET instances in terms of the relative benefits of path and connection diversity and their diminishing returns. One difference between the two sets of results is that the percent improvement in solution quality is smaller for the GEANT topology, which we explain shortly. Nevertheless, even a small improvement for the larger GEANT2 network (which represents a much larger total demand) translates into significant savings of network resources, especially since these spectrum savings apply to a larger number of links.

To put the results of Figures 1-4 into perspective, Figures 5 and 6 plot the the percentage of total demand that the c high-priority connections represent, for the two networks and three traffic distributions we considered in this study. Recall that the average demand size is smallest (respectively, largest) for the skewed low (respectively, high) distribution, with the demand size of the uniform distribution falling in between these two. Therefore, for a given value of c, the percentage of total demand represented by the c high-priority (i.e., largest in our study) connections is smallest for the skewed high distribution, followed by the uniform and skewed low distribution; this is

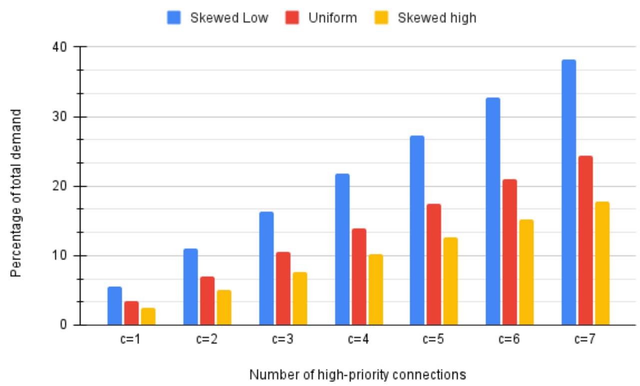

Fig. 5. Fraction of total demand for high-priority connections, NSFNET

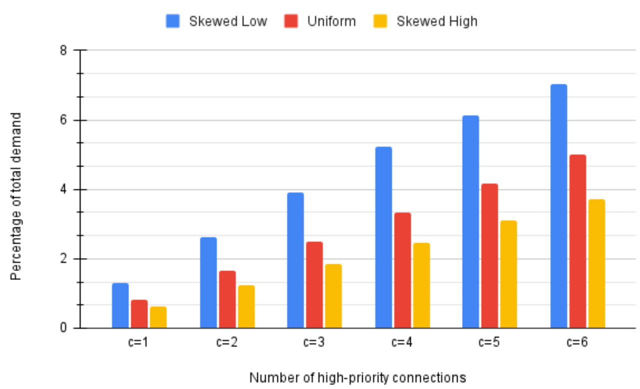

Fig. 6. Fraction of total demand for high-priority connections, GEANT2

consistent with the results shown in the two figures. However, there are 91 $(= 14 \times 13/2)$ connections for the NSFNET but 496 $(= 32 \times 31/2)$ connections for the larger GEANT2 network. Therefore, as we observe in the two figures, c connections make up a smaller fraction of total demand for the GEANT2 network relative to NSFNET. Consequently, optimizing for c connections is expected to have a smaller relative improvement in solution quality for GEANT2 than NSFNET, consistent with the results we presented earlier.

Overall, we can draw several general conclusions regarding the performance of our RSA algorithm across the network topologies and traffic distributions used in our study: 1) solution quality improves with both path and connection diversity; 2) connection diversity has relatively higher benefits than path diversity; 3) the gains of path diversity level off after $k = 4$, consistent with earlier studies of alternate path routing; and 4) exhaustively exploring the solution space of a few high-priority connections pays off significantly even when these connections represent a relatively small fraction of total demand.

V. Concluding Remarks

We have presented the first spectrum symmetry-free algorithm that can be applied to all variants of the RSA problem. Our method explores the whole solution space of a subset of connections, and our simulation results indicate that connection diversity is more beneficial than path diversity. Our group is working on scaling this approach in terms of the size of solution space that can be explored by appropriately extending the branch-and-bound RFF algorithm of [14].

References

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

[2] M. Jinno, B. Kozicki, H. Takara, A. Watanabe, Y. Sone, T. Tanaka, and A. Hirano, "Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network," *IEEE Communications Magazine*, vol. 48, no. 8, pp. 138–145, 2010.

[3] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Elastic bandwidth allocation in flexible OFDM–based optical networks," *Journal of Lightwave Technology*, vol. 29, no. 9, pp. 1354–1366, 2011.

[4] M. Klinkowski, P. Lechowicz, and K. Walkowiak, "Survey of resource allocation schemes and algorithms in spectrally-spatially flexible optical networking," *Optical Switching & Netw.*, vol. 27, no. C, pp. 58–78, 2018.

[5] S. Talebi, F. Alam, I. Katib, M. Khamis, R. Khalifah, and G. N. Rouskas, "Spectrum management techniques for elastic optical networks: A survey," *Optical Switching and Networking*, vol. 13, pp. 34–48, July 2014.

[6] R. Dutta and G. N. Rouskas, "A survey of virtual topology design algorithms for wavelength routed optical networks," *Optical Networks*, vol. 1, pp. 73–89, January 2000.

[7] R. Dutta and G. N. Rouskas, "Traffic grooming in WDM networks: Past and future," *IEEE Network*, vol. 16, pp. 46–56, Nov/Dec 2002.

[8] B. Chen, G. N. Rouskas, and R. Dutta, "Clustering methods for hierarchical traffic grooming in large-scale mesh WDM networks," *Journal of Optical Communications and Networking*, vol. 2, pp. 502–514, August 2010.

[9] D. Zhou and S. Subramaniam, "Survivability in optical networks," *IEEE Network*, vol. 14, pp. 16–23, November/December 2000.

[10] S. Talebi, E. Bampis, G. Lucarelli, I. Katib, and G. N. Rouskas, "Spectrum assignment in optical networks: A multiprocessor scheduling perspective," *Journal of Optical Communications and Networking*, vol. 6, pp. 754–763, August 2014.

[11] F. Bertero, M. Bianchetti, and J. Marenco, "Integer programming models for the routing and spectrum allocation problem," *TOP*, vol. 26, pp. 465–488, October 2018.

[12] L. Velasco, M. Ruiz, and M. Klinkowski, *The Routing and Spectrum Allocation Problem*, ch. 3, pp. 43–60. John Wiley & Sons, Ltd, 2017.

[13] J. Wang, M. Shigeno, and Q. Wu, "ILP models and improved methods for the problem of routing and spectrum allocation," *Optical Switching and Networking*, vol. 45, p. 100675, 2022.

[14] G. N. Rouskas and C. Bandikatla, "Recursive first fit: A highly parallel optimal solution to spectrum allocation.," *IEEE/Optica Journal of Optical Communications and Networking*, vol. 14, pp. 165–176, April 2022.

[15] B. Jaumard, C. Meyer, and B. Thiongane., "ILP formulations for the routing and wavelength assignment problem: Symmetric systems," in *Handbook of Optimization in Telecommunications*, pp. 637–677, Springer US, 2006.

[16] G. N. Rouskas, "First-Fit: a universal algorithm for spectrum allocation," in *Proceedings of IEEE GLOBECOM 2023*, December 2023.

[17] J. Simmons and G. N. Rouskas, "Routing and wavelength (spectrum) allocation," in *B. Mukherjee, I. Tomkos, M. Tornatore, P. Winzer, and Y. Zhao (Editors), Springer Handbook of Optical Networks*, Springer, 2020.

[18] G. N. Rouskas, S. Gupta, and P. Sharma, "Experimental evaluation of a symmetry-free parallel algorithm for spectrum allocation," in *Proceedings of IEEE GLOBECOM 2022*, December 2022.

[19] S. Talebi and G. N. Rouskas, "On distance-adaptive routing and spectrum assignment in mesh elastic optical networks," *Journal of Optical Communications and Networking*, vol. 9, pp. 456–465, May 2017.

[20] F. Tang, G. Shen, and G. N. Rouskas, "Crosstalk-aware shared backup path protection in multi-core fiber elastic optical networks.," *IEEE/OSA Journal of Lightwave Technology*, vol. 39, pp. 3025–3036, May 15 2021.

[21] N. State, "Henry2 Linux Cluster." https://projects.ncsu.edu/hpc/About/ComputeResources.php.