Parameterized Exhaustive Routing with First Fit for RSA Problem Variants

George N. Rouskas, Chaitanya Bandikatla
North Carolina State University

***Abstract*—We present a new single-step solution approach for the routing and spectrum allocation (RSA) problem that integrates the first-fit (FF) heuristic with a new routing strategy that we refer to as "parameterized exhaustive routing." Our approach is to explore the whole routing space for a subset of the traffic requests, e.g., those with the largest demands or those of higher priority or importance. For each of the remaining requests we employ a greedy heuristic to select one of the candidate paths jointly with spectrum allocation. Our solution represents a two-parameter family of algorithms that bridges the gap between an exhaustive search of the routing space and current two-step methodologies for the RSA problem that select paths for each traffic request in isolation. The parameter values may be used to trade off the quality of the final solution and the computational requirements. Our results indicate that exploring the joint routing space of even a few large requests leads to better solutions than purely greedy approaches.**

I. Introduction

Routing and spectrum allocation (RSA) is a fundamental problem in the design, operation, and control of elastic optical networks [1]–[6], and underlies a range of optimization problems including virtual topology design [7], traffic grooming [8], [9], and network survivability [10]. The choice of route affects several features of an optical connection, including delay, availability, cost, etc. Since connections in flexible (elastic) networks may be assigned variable amounts of spectrum, spectrum allocation involves challenging contention issues. Specifically, finding a path with adequate bandwidth (i.e., spectrum slots) is not sufficient to accommodate a connection, as the slots must be contiguous (consecutive) in the available spectrum. Therefore, maintaining a high level of capacity efficiency in elastic optical networks is a difficult problem. In fact, the RSA problem has been shown to be NP-hard [3], even in simple network topologies [11].

Several integer linear programming (ILP) formulations have been proposed for the RSA problem; for a recent survey, the reader is referred to [12]. ILP models represent single-step solutions to the RSA problem in that the routing and spectrum allocation aspects are tackled together in an integrated manner. Nevertheless, algorithm scalability is an important concern in elastic networks since, due to spectrum slicing, each network link may have many more spectrum slots for allocation than wavelengths in a traditional network. Since solving the ILP directly is feasible only in small problem instances [13], most solution approaches treat the routing and spectrum allocation (as well as aspects related to modulation, regeneration, etc., in various RSA problem variants) as separate steps.

In a typical two-step heuristic approach, first a path is selected for each traffic request, and then spectrum is assigned [14]. While the two steps are carried out separately, they inform each other. Usually, in the routing step a set of candidate paths is pre-calculated for each source-destination pair. Then either link load or some measure of spectrum fragmentation [15] is used to select one of the paths for a connection. In the spectrum allocation step, the paths of all connections are known and several heuristic algorithms have been developed to assign spectrum slots to each connection, including first-fit, best-fit, most-used, and least-loaded [16]. In particular, the simple first-fit heuristic is commonly used for spectrum assignment as it requires no global knowledge and has good performance across various network topologies and sets of traffic demands [14], [17], [18].

Our work is motivated by two shortcomings of typical two-step solution approaches. First, the routing step selects a path for each traffic request in isolation, i.e., one at a time, usually starting with requests for which finding a route is more challenging (that is, those having large demands or requiring longer paths). The routing algorithm indirectly takes into account the impact of requests that have been routed earlier, e.g., through link costs that represent the current link load or spectrum fragmentation. However, once the path for a connection has been selected, it does not change. Therefore, when selecting the path for a later request, it is not possible to take into account candidate paths for an earlier request other than the one that has already been selected. Similarly, the spectrum allocation step is limited to considering only one path for each connection, the one selected in the first step of the process, and hence all additional information (and potential benefits) regarding candidate paths not selected is lost.

Based on the observation that the first-fit (FF) algorithm works well in practice and the FF property we derived in [19], we present a new single-step solution approach for the RSA problem that integrates FF with a new routing strategy in a way that 1) jointly considers candidate paths for multiple traffic requests, and 2) takes into account all candidate paths of a traffic request when allocating spectrum. Specifically, our approach is to explore the whole routing space for a subset of the traffic requests; this subset may consist of the

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

requests with the largest demands, or those that are deemed the most important under any reasonable criteria. For each of the remaining requests we employ a greedy heuristic to select one of the candidate paths jointly with spectrum allocation. Our solution represents a two-parameter family of algorithms that we refer to as "parameterized exhaustive routing." This algorithm family bridges the gap between an exhaustive search of the routing space and current two-step approaches that select paths for each request in isolation. The parameter values may be used to trade off the quality of the final solution and the computational requirements.

The remainder of the paper is organized as follows. In Section II, we define the RSA problem that we consider in this work and review our recent results in tackling its subproblems. In Section III we introduce a two-parameter family of algorithms that integrate a new routing strategy with first-fit spectrum allocation as a single-step solution to the RSA problem. We evaluate the algorithm family in Section IV, and we conclude the paper in Section V.

II. The Routing and Spectrum Allocation (RSA) Problem

We consider an optical network with a topology described by graph $G = (V, A)$, where V is the set of vertices (nodes) and A is the set of arcs (directed fiber links) in the network. Let $N = |V|$ be the number of nodes and $L = |A|$ be the number of directed links; without loss of generality, we assume that if there is a fiber link from some node A to some other node B in the network, then there is a fiber link in the opposite direction, from node B to node A. We are given a set $\mathcal{T} = \{T_i, i = 1 \cdots, M\}$, of M traffic requests, such that each request is a tuple $T_i = (s_i, d_i, t_i, \mathcal{P}_i)$, where:

- s_i and d_i are the source and destination nodes, respectively, of the request,
- 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 physical paths $\{p_i^{(1)}, \cdots, p_i^{(K)}\}$ between nodes s_i and d_i in the network.

We assume that K is a small integer, e.g., $K = 2 - 5$. The K paths of a request are pre-determined (i.e., they are provided as input to the problem) and are fixed (i.e., they do not change once they have been selected). The paths of a request may be calculated as the K shortest paths between the particular source-destination pair, or using any other desirable criteria. Further, we assume that the spectrum demand t_i of a request must be carried entirely on one of the paths in $\mathcal{P}_i$, hence splitting the spectrum demand over multiple paths is not allowed.

We consider the following basic definition of the routing and spectrum allocation (RSA) problem:

Definition 2.1 (RSA): Given a graph $G = (V, A)$ and a set $\mathcal{T} = \{T_i = (s_i, d_i, t_i, \mathcal{P}_i)\}$ of traffic requests, select one of the physical paths $p_i^{(1)}, \cdots, p_i^{(K)}$, for each request T_i and assign t_i spectrum slots along this path so as to minimize the total amount of spectrum used on any link in the network, under three constraints: 1) each request T_i is assigned a block of t_i contiguous spectrum slots (contiguity constraint), 2) each request is assigned the same block of spectrum slots along all links of its path (spectrum continuity constraint), and 3) requests whose paths share a link are assigned non-overlapping spectrum slots (non-overlapping spectrum constraint).

It is well known that the RSA problem is intractable [4], [14]. Specifically, the two subproblems of RSA, namely, the routing subproblem and the spectrum allocation subproblem are coupled. We now discuss these two subproblems in more detail and describe our recent work in developing recursive algorithms to solve each problem optimally but separately from the other. This discussion provides the motivation for the new algorithm we present in the following section.

A. Subproblems of RSA and Integrated Solution

The routing subproblem of RSA involves selecting one of the K paths for each request. Let us define a *routing configuration* $R_j, j = 1, \cdots, C$, as an assignment of one path to each traffic request, whereby the path assigned to a request T_i is selected among the set $\mathcal{P}_i$ of K paths input to the RSA problem. Then, the routing subproblem is equivalent to selecting one of the C routing configurations, where $C = K^M$. Assuming that there is a traffic request between each node pair in the network, then $M = O(N^2)$ and the number of routing configurations is exponential in the size of the network. Furthermore, even if K is fixed to a small integer (e.g., $K = 2$), C can be a very large number.

In recent work [20] we showed that by searching the entire routing space of C configurations the routing and resource allocation aspects of the RSA problem can be optimally decoupled. Accordingly, we developed a branch-and-bound algorithm to search the entire routing space exhaustively yet efficiently. The algorithm is designed to apply the first-fit (FF) heuristic on each configuration as it searches the routing space, and selects the configuration that yields the best solution in terms of spectrum allocation.

The spectrum allocation (SA) subproblem, on the other hand, assumes that the path for each request is fixed (given), in which case the objective is to assign spectrum resources to carry the traffic demand of each request subject to the three constraints of Definition 2.1. We considered the SA problem in [19] and we showed that there exists a permutation of the M traffic requests such that applying the FF heuristic to the requests in the order implied by this permutation yields an optimal solution. We then developed an efficient recursive first-fit (Rec-FF) algorithm to search the entire space of $O((N^2)!)$ permutations for the optimal one.

Our work in [19], [20] suggests a possible approach to tackling the two subproblems in an integrated (i.e., single-step) manner so as to find an optimal solution to the original RSA problem: namely, apply the branch-and-bound algorithm in [20] to search the routing space but modify it to employ the optimal Rec-FF algorithm in [19], rather than the FF heuristic, to search the entire space of request permutations in determining the spectrum allocation for each routing configuration. While our group is exploring this direction, we emphasize that there exist at least two challenges. First, the branch-and-bound algorithm builds routing configurations recursively by considering one traffic request at a time and incrementally applying the FF heuristic; hence it is not straightforward to modify it to apply the Rec-FF algorithm that searches over all request permutations. Second, even if the two algorithms can be combined appropriately, the size of the combined solution space, equal to the product of the sizes of the spaces of the individual subproblems, may be prohibitive to search exhaustively.

In the following, we present a solution approach that integrates routing and first-fit spectrum allocation in an efficient manner and performs an exhaustive search only on a part of the routing space whose size is determined in advance by the values of certain configuration parameters.

III. A Family of Parameterized Exhaustive Routing with First Fit (PER-FF) Algorithms

Our goal is to bridge the gap between an exhaustive search, which may be prohibitively expensive computationally for real-life networks, and typical heuristic approaches, by introducing a parameterized approach that achieves a desirable tradeoff between running time and quality of solution. Specifically, we present parameterized exhaustive routing and first-fit, PER-FF(k, m), a family of algorithms for RSA problem variants characterized by two parameters:

1) the number of paths, k, for each traffic request, and
2) the number $m < M$ of the requests for which all routing configurations are explored.

The PER-FF(k, m) algorithm uses the FF heuristic to assign spectrum to the requests, but it handles requests differently depending on their demands. For the m largest requests, the algorithm employs an exhaustive search and applies the FF heuristic to all possible k^m routing configurations. For each of the $M - m$ smaller requests, on the other hand, it employs a greedy heuristic to select one of the k paths on which to route each request. This approach has two benefits: 1) it explores all routing combinations for the largest requests, which collectively represent a significant fraction of total demand, while also 2) it takes into account the routing paths of smaller requests, albeit in a greedy manner.

Figure 1 provides a pseudo-code description of the PER-FF(k, m) algorithm. Note that parameter k represents the amount of path diversity and parameter m the scope of the exhaustive search. In particular, for $k \geq 2$, PER-FF reduces to a pure greedy algorithm for $m = 0$ and becomes an exhaustive search of the whole routing space at the other extreme of $m = M$. By carefully selecting values for the two parameters k and m, a network designer may strike a desirable balance between the running time required to explore all k^m routing configurations and the quality of the final solution.

PER-FF(k, m) Algorithm

1) **Ordering of Traffic Requests**
   Sort the M requests $T_1, \cdots, T_M$, in decreasing order of demand, i.e., such that $t_i \geq t_{i+1}, i = 1, \cdots, M - 1$, with ties broken arbitrarily.
2) **Exhaustive Enumeration of Routing Configurations**
   Consider the $m < M$ largest requests $T_1, \cdots, T_m$, and enumerate all k^m routing configurations for these requests. Each routing configuration R_j is a tuple
   $$R_j = [p_1, \cdots, p_m],\ j = 1, \cdots, k^m, \quad (1)$$
   where $p_i \in \mathcal{P}_i, i = 1, \cdots, m$, is one of the k paths for request T_i.
3) **Joint Routing and Spectrum Allocation**
   For each routing configuration $R_j, j = 1, \cdots, k^m$ perform spectrum allocation by considering the traffic requests in the sorted order $T_1, \cdots, T_M$:
   a) **FF for the m Largest Requests**
      Assign spectrum to the m largest requests $T_1, \cdots, T_m$ using FF along the path for each request indicated in R_j.
   b) **FF with Greedy Routing for the $M - m$ Smallest Requests**
      For each request $T_i, i = m + 1, \cdots, M$:
      i) Select the path p among $p_i^{(1)}, \cdots, p_i^{(k)}$ such that assigning spectrum to T_i using FF results in the best solution.
      ii) Extend the routing configuration R_j to include path p for request T_i.
4) **Return** the extended routing configuration R_j that represents the best solution

Fig. 1. The parameterized exhaustive search with first-fit (PER-FF) algorithm

In Figure 1 and the simulations we present in the next section we assume that the algorithm performs exhaustive search over the routing space of the m requests with the *largest demand*. Intuitively, large requests require correspondingly large resources and hence must be routed carefully, not only individually but in combination with other large requests, to ensure that spectrum is allocated efficiently. Furthermore, a small fraction of all requests may account for a considerable fraction of total demand (refer also to the footnote in Section IV), hence, exploring the entire routing space of such requests may be computationally feasible. However, there is no inherent requirement that the PER-FF algorithm consider the m largest requests. A network operator may apply the algorithm so that it exhaustively explores the routing space of *any* m requests, e.g., requests that are somehow important or have higher priority than others regardless of the demand size.

We note that the PER-FF family of algorithms 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 [21], intra- or inter-core crosstalk [22], etc. Additional constraints may eliminate some of the routing configurations, reducing the effective size of the routing space well below k^m and, thus, allowing for larger values for parameters k and/or m. Nevertheless, due to page limitations our study focuses on the basic RSA problem and we plan to consider constrained variants of the problem in future work.

IV. Simulation Study

We now evaluate the PER-FF algorithm and compare it to baseline algorithms on a number of RSA problem instances.

A. Generation of Problem Instances

Each RSA problem instance is characterized by two parameters: the network topology and the distribution used to generate random traffic demands. In our evaluation study, we used two network topologies, the 14-node, 21-link NSFNet and the 32-node, 54-link GEANT2 network. For each topology, we create problem instances by generating traffic requests 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 of the five rates 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.

Once the traffic rates between each node pair have been generated, we calculate the corresponding spectrum slots by assuming that the slot width is 12.5 GHz, and adopting the parameters of [2] to determine the number of spectrum slots that each demand requires based on its data rate and path length. For each request, we also use the depth first search (DFS) algorithm to calculate the K shortest paths between the corresponding source-destination pair.

B. Evaluation Metric and Algorithms

The performance measure we consider is the maximum number of spectrum slots on any network link. In our simulation experiments, we have compared the solutions obtained by three algorithms:

1) **SP-FF: Shortest path routing with FF**. Each request is routed over its shortest path and the FF heuristic is used to assign spectrum to the requests in decreasing order of their spectrum demands.
2) **SP-Rec-FF: Shortest path routing with Recursive FF**. Each request is routed over its shortest path, and we use the Rec-FF algorithm we developed in [19], and which we described in Section II-A, to allocate spectrum to the traffic requests.
3) **PER-FF(k, m)**. This is the parameterized exhaustive routing and FF algorithm in Figure 1. For the experiments we present in this section, we use various values for parameters k and m.

We note that the SP-FF algorithm above belongs to the family of PER-FF(k, m) algorithms: it is equivalent to PER-FF($k = 1, m = M$) since there is only one routing configuration in which all M requests are routed along their only (shortest) path. For a meaningful comparison between different problem instances, we also calculate:

4) **SP-LB: Lower bound under shortest path routing.** The lower bound is calculated by ignoring the problem constraints and simply adding up the demands along each link and taking the maximum value over all links.

Fig. 2. Normalized solutions to 300 problem instances, NSFNET

We normalize the solutions returned by the SP-FF, SP-Rec-FF, and PER-FF algorithms by dividing with the lower bound SP-LB for the corresponding instance; this normalization makes the results of the algorithms comparable across problem instances. We note that since the SP-FF and SP-Rec-FF algorithms use shortest path routing, then SP-LB is a lower bound on the solutions that these two algorithms return. However, we emphasize that *SP-LB does not represent a lower bound for RSA algorithms that use two or more paths for each request.* In our experiments, PER-FF uses two or more paths for each request, and hence, as we will show in a moment, it finds solutions better than SP-LB.

C. Results and Discussion

Figures 2 and 3 present results for the NSFNet and GEANT2 topologies, respectively. Each figure includes three sub-figures, one each for demand matrices generated by the uniform, skewed low, and skewed high distributions, respectively. Each sub-figure plots the normalized value of SP-LB (shown as the horizontal line at $y = 1.0$), and the normalized values of the SP-FF, SP-Rec-FF, and PER-FF($k = 3, m = 12$) solutions, for each of 100 random problem instances generated for the stated parameters (i.e., network topology and traffic

Fig. 3. Normalized solutions to 300 problem instances, GEANT2

demand distribution). For these parameter values for PER-FF, each request may be routed in one of three shortest paths, and the algorithm explores all routing configurations for the largest twelve requests.

We first note that the SP-FF algorithm produces solutions of good quality that are within 30% (respectively, 12%) of the SP-LB lower bound for the 300 NSFNet (respectively, GEANT2) problem instances. These results are consistent with earlier research indicating that the FF algorithm performs well. By considering all possible permutations of the M requests, the SP-Rec-FF algorithm finds better solutions than FF in most instances, as we explained in [19]. However, both the SP-FF and SP-Rec-FF algorithms are limited by the fact that they only use the shortest path for each request, and therefore their solutions are bounded from below by SP-LB. Nevertheless, the SP-Rec-FF algorithm often finds solutions that are equal to SP-LB and, hence, optimal under the additional constraint that all requests be routed over their shortest paths.

The PER-FF($k = 3, m = 12$) algorithm, on the other hand, is not limited by shortest path routing, and it also explores the routing configurations of the largest $m = 12$ traffic requests[1]. As a result, the algorithm not only outperforms the two algorithms that use shortest path routing, SP-FF and SP-Rec-FF, but it produces solutions that, for most instances shown in Figures 2 and 3 are well below the SP-LB value. In particular, while for the NSFNet topology the PER-FF algorithm sometimes produces solutions that are higher than SP-LB, for the larger GEANT2 topology all but one PER-FF solutions are strictly below SP-LB. These results are consistent across the two topologies and traffic demand distributions and indicate that 1) exploring the routing space while considering requests in one specific order (of decreasing demand), as PER-FF does, results in better solutions than limiting requests to the shortest path and exploring the order of requests, as SP-Rec-FF does, and 2) exploring the routing space is more effective in larger networks with higher path diversity.

Table I summarizes the average relative performance of the SP-FF, SP-Rec-FF, and the PER-FF($k = 3, m = 12$) algorithm in terms of how far their solutions are from SP-LB, both in percentage-wise and in terms of spectrum slots. Note that positive (respectively, negative) values mean that the corresponding solution is worse or higher (respectively, better or lower) than the SP-LB value. The table confirms that SP-Rec-FF performs better than SP-FF, but both solutions are higher than SP-LB. The PER-FF($k = 3, m = 12$) algorithm, on the other hand, finds solutions that are, on average, well below the SP-LB value. Specifically, the PER-FF solutions are between 15.9-19.92% lower than SP-LB for the NSFNet, and between 21.39-26.83% lower than SP-LB for GEANT2. Importantly, the larger percentage difference for the larger GEANT2 network with a much larger total demand, translates into a significantly decrease in the number of slots required, especially since these spectrum savings apply to a larger number of links.

Table II compares four algorithms in the PER-FF family on the NSFNet; we have obtained similar results for the GEANT2 topology, which, therefore, we omit. The algorithms differ in the number of paths they consider for each request ($k = 2, 3$) and the number of largest requests whose routing combinations are explored ($m = 0, 12$). In particular, the two algorithms with $m = 0$ are pure greedy algorithms in that they apply the greedy Step 3.b of Figure 1 to all M requests.

From the results shown in Table II and in combination with those in Table I, we make several important observations. First, all algorithms produce results that are below the SP-LB value. Since under shortest path routing even optimal spectrum allocation (i.e., Rec-FF from Table I is bounded below by SP-LB, this indicates that introducing path diversity is crucial in obtaining good solutions to the RSA problem. Second, both parameters k and m make a difference in terms of the quality of final solution. Specifically, using three rather than two paths results in better solutions both for the greedy version of the algorithm (i.e., with $m = 0$) and the algorithm that exhaustively explores the routing configurations of $m = 12$ requests. Similarly, for the same number of paths, exhaustively exploring the routing configurations for $m = 12$ requests produces better solutions than the greedy algorithm. Finally,

[1]For the NSFNet, the $m = 12$ largest requests are 13.2% of the 91 requests between all source-destination pairs, and for the traffic distributions we consider here, represent between 30.5-50.0% of the total demand. For GEANT2, the $m = 12$ largest requests are 2.4% of the 496 requests and represent between 7.3-13.0% of the total demand, depending on the traffic distribution.

TABLE I
RELATIVE PERFORMANCE OF SP-FF, SP-REC-FF, AND PER-FF($k = 3, m = 12$) ALGORITHMS

	Traffic	SP-FF	SP-Rec-FF	PER-FF($k = 3, m = 12$)	
		% from SP-LB	% from SP-LB	% from SP-LB	Avg Diff from SP-LB (slots)
NSFNet	Skewed High	9.28%	5.46%	-15.90%	-16.39
	Skewed Low	11.73%	6.55%	-19.92%	-10.96
	Uniform	10.12%	6.01%	-17.50%	-13.82
GEANT2	Skewed High	2.66%	1.22%	-26.83%	-162.07
	Skewed Low	6.58%	3.54%	-21.39%	-57.54
	Uniform	2.88%	1.37%	-25.65%	-113.22

TABLE II
RELATIVE PERFORMANCE OF FOUR PER-FF(k, m) ALGORITHMS AS % BELOW SP-LB, NSFNET

Traffic	PER-FF(k, m)			
	$k = 2, m = 0$	$k = 3, m = 0$	$k = 2, m = 12$	$k = 3, m = 12$
Skewed High	-5.35%	-7.81%	-14.53%	-15.90%
Skewed Low	-12.35%	-16.60%	-19.33%	-19.92%
Uniform	-8.30%	-10.25%	-15.76%	-17.50%

exploring routing configurations leads to better solutions than simply increasing the number of paths.

V. CONCLUDING REMARKS

We have developed a parameterized exhaustive routing methodology that can be used to tackle RSA problem variants. Our method represents a two-parameter family of algorithms that exhaustively explores the joint routing space of a subset of traffic requests. The values of the two parameters directly determine the size of the routing space to be explored, and hence the choice of values helps reach a desirable middle ground between computational requirements and quality of solution. We have presented simulation results which indicate that exploring the routing space of a relatively small fraction of large requests leads to better results (compared to purely greedy approaches) and significant savings in terms of spectrum slots. Our group is currently investigating the potential of this approach in terms of the size of routing space that can be explored, taking into account any diminishing returns.

REFERENCES

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

[2] M. Jinno, *et al.*, "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, *et al.*, "Elastic bandwidth allocation in flexible OFDM–based optical networks," *J. Lightwave Tech.*, vol. 29, no. 9, pp. 1354–1366, 2011.

[4] M. Klinkowski, *et al.*, "Survey of resource allocation schemes and algorithms in spectrally-spatially flexible optical networking," *Optical Switching and Networking*, 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, 2014.

[6] V. Sivaraman and G. N. Rouskas, "HiPeR-ℓ: A High Performance Reservation protocol with ℓook-ahead for broadcast WDM networks," in *Proceedings of INFOCOM '97*, pp. 1272–1279, IEEE, April 1997.

[7] 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.

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

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

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

[11] 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.

[12] 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.

[13] B. Jaumard, C. Meyer, and B. Thiongane, "Comparison of ILP formulations for the RWA problem," *OSN*, vol. 3-4, pp. 157–172, 2007.

[14] J. Simmons and G. N. Rouskas, "Routing and wavelength (spectrum) allocation," *Springer Handbook of Optical Networks*, Springer, 2020.

[15] X. Wang, Q. Zhang, I. Kim, P. Palacharla, and M. Sekiya, "Utilization entropy for assressiong resource fragmentation in optical networks," in *Proceedings of Optical Fiber Communication and National Fiber Optic Engineers Conference (OFC/NFOEC)*, p. 0Th1A.2, March 2012.

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

[17] Y. Zhu, G. N. Rouskas, and H. G. Perros, "A comparison of allocation policies in wavelength routing networks," *Photonic Network Communications*, vol. 2, pp. 265–293, August 2000.

[18] G. N. Rouskas, "Routing and wavelength assignment in optical WDM networks," in *J. Proakis (Editor), Wiley Encyclopedia of Telecommunications*, John Wiley & Sons, 2001.

[19] G. N. Rouskas and C. Bandikatla, "Scalable recursive first fit: An optimal solution to the spectrum allocation problem," in *Proceedings of ONDM 2021*, June 2021.

[20] M. Fayez, I. Katib, G. N. Rouskas, T. F. Gharib, and H. Faheem, "A scalable solution to network design problems: Decomposition with exhaustive routing search," *Proc. IEEE GLOBECOM 2020*, Dec. 2020.

[21] 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.

[22] 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.