

Efficient Spectrum Utilization in Large Scale RWA Problems

Brigitte Jaumard, *Senior Member, IEEE*, and Maryam Daryalal

Abstract—While the routing and wavelength assignment (RWA) has been widely studied, a very few studies attempt to solve realistic size instances, namely, with 100 wavelengths per fiber and a few hundred nodes. Indeed, state of the art is closer to around 20 nodes and 30 wavelengths, regardless of what the authors consider, heuristics or exact methods with a very few exceptions. In this paper, we are interested in reducing the gap between realistic data sets and test bed instances that are often used, using exact methods. Even if exact methods may fail to solve in reasonable time very large instances, they can, however, output ε -solutions with a very good and proven accuracy. The novelty of this paper is to exploit the observations that optimal solutions contain a very large number of light paths associated with shortest paths or k -shortest paths with a small k . We propose different RWA algorithms that lead to solve exactly or near exactly much larger instances than in the literature, i.e., with up to 150 wavelengths and 90 nodes. Extensive numerical experiments are conducted on both the static and dynamic incremental planning RWA problem.

Index Terms—Routing, wavelength assignment, network provisioning, decomposition method, k -shortest paths.

I. INTRODUCTION

OFFLINE Routing and Wavelength Assignment (RWA) is the key provisioning problem of optical networks. The continuous growth of traffic, nourished by the emerging content-rich, high-rate and bursty applications, such as video on demand, on-line gaming, high-definition television (HDTV), and cloud computing, can only be met with the abundant capacity provided by optical transport networks. The following numbers provide an idea of what growth of traffic means in 2015. For instance, Global Internet Protocol (IP) traffic has increased more than fivefold in the past 5 years, and will increase nearly threefold over the next 5 years according to a Cisco report [1]. In terms of the current size of the optical networks, a recent study (Durairajan *et al.* [2]) of the US long-haul fiber-optic infrastructure, based on the detailed fiber deployment maps from 5 tier-1 and 4 major cable providers (AT&T, Comcast, Cogent, EarthLink, Integra, Level3, Suddenlink, Verizon and Zayo), suggests that the US long-haul fiber network has 273 nodes/cities, 2,411 links, and 542 conduits.

Manuscript received March 22, 2016; revised August 22, 2016; accepted October 13, 2016; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor A. Eryilmaz. Date of publication March 6, 2017; date of current version April 14, 2017. This work was supported in part by the Concordia University Research Chair (Tier I) and in part by the Natural Sciences and Engineering Research Council of Canada Grant.

B. Jaumard is with the Department of Computer Science and Software Engineering, Concordia University, Montreal, QC H3G 2W1, Canada (e-mail: bjaumard@cse.concordia.ca).

M. Daryalal was with the Department of Computer Science and Software Engineering, Concordia University, Montreal, QC H3G 2W1, Canada. She is now with Morgan Stanley, Montreal, QC H3C 3S4, Canada.

Digital Object Identifier 10.1109/TNET.2016.2628838

The RWA problem was the focus of many studies in the 80's and numerous heuristics have been proposed. However, they have not been tested and compared on very large data sets except for very few papers, see, e.g., [3]–[7], and about half of the studies do not provide any information on the accuracy of their solutions. On the other hand, exact methods have difficulties to scale. The objective of the present paper is to revisit the Integer Linear Program (ILP) model based on independent sets [8], [9] and to increase its scalability through several enhancements. Firstly, we will combine three different ILP formulations, i.e., link, path and configuration ones, in the solution process and take advantage of the strengths of each of them at different stages of the solution process. Difference with [9] lies in the use of the path formulation for solving the pricing problem in a heuristic fashion before moving to a link formulation to guarantee an ε -optimality of the solutions. Secondly, we design an efficient heuristic rule for selecting the pre-computing paths in order to feed the path formulation. Such a heuristic takes into account the distances between the node pairs, the amount of requested traffic, and the increased length of the routes when using k -shortest paths. Lastly, we propose some new RWA heuristics and compare their solutions with the ε -optimal solutions output by the exact approaches.

Given a set of lightpath requests, two variants of the RWA problem have been studied in the literature: max-RWA and min-RWA. For the former problem, the focus is on the provisioning over an existing network, and the objective is to maximize the number of lightpath requests that can be routed with a given number of wavelengths. The latter one has a planning/dimensioning concern and the objective is to minimize the number of required wavelengths to route all the requests.

Several surveys have been written on the RWA problem, where the reader can find a comprehensive survey of the various mathematical models that have been investigated, see, e.g., [10] for symmetrical traffic and [11] for asymmetrical traffic. Among the papers that explored exact solutions, we find three classes of Integer Linear Programming (ILP) formulations for the RWA problem: (i) link-based, (ii) path-based, or (iii) Maximal Independent Set (MIS)-based (or configuration based, or Column Generation (CG) ILP model). Comparison of link and path based can be found in [9] and [10] with the objective of maximizing the grade of service. The authors showed that very often, but not always, the Linear Programming (LP) bound is equal to the ILP value, with however an LP solution that is usually with fractional values. Conclusions of the current paper are that, while for small data sets, the LP bound is rather accurate, it is not anymore true for large ones.

TABLE I
COMPARISON OF THE NUMBER OF VARIABLES AND
CONSTRAINTS IN THE ILP RWA MODELS

ILP models	# variables	# constraints
Link formulations, results adapted from Table I in [10] using the formulations in Appendix A		
RWA _k	Link formulation #1: request indexed $W K (1 + L)$	$(n + 2 + L)W K + L W + K $
RWA _s	Link formulation #2: source indexed $nW L $	$n(n - 1)(W + 2) + L W$
RWA _{sd}	Link formulation #3: node pair indexed $n(n - 1)W L $	$n(n - 1)(n - 2)W + 2n(n - 1) + W L $
Path formulations		
z_{PATH}	$W \mathcal{P} $	$W L + \mathcal{SD} $
Configuration (CG) formulations (see Sections III and IV for the details)		
CG ILP ^(*)	Master Problem	
	$2 \mathcal{SD} + 1$	$2 \mathcal{SD} + 1$
	Pricing Problem (Link Formulation)	
	$ L \mathcal{SD} $	$ L + 2 \mathcal{SD} + (n - 2) \mathcal{SD} $
	Pricing Problem (Path Formulation with One Shortest Path per Node Pair)	
$ \mathcal{SD} $		$ L + \mathcal{SD} $
n, L, W, \mathcal{SD} : see Section II for their definition \mathcal{P} = set of all possible paths in the optical network K = set of individual requests $(*)$ with a column generation implementation in which we keep only the basis variables, see, e.g., [14]		

In [10], a theoretical and experimental study is proposed for comparing the different LP bounds provided by the various existing ILP RWA models: all LP bounds are shown to be exact. Moreover, a comparison of the models is made with respect to their numbers of variables and constraints: it clearly shows that the CG model is the most economical one. We provide a synthesis of all the results discussed in [9]–[11] in Table I and include the CG ILP model for the max-RWA problem (described in Section III), see Appendix for the other ILP models. Other surveys can be found in Miliotis *et al.* [12] and Zang *et al.* [13].

The largest instances that have been solved exactly so far for mesh networks are the EON network (20 nodes, and 39 optical links) with 24 wavelengths, the Brazil network (27 nodes and 70 optical links) with 14 wavelengths, both for the GoS (Grade of Service) maximization [10]. In terms of meta-heuristics, large instances have been solved by Martins *et al.* [5]: 26 realistic instances with up to either 90 nodes or 175 links. Therein, the quality of the meta-heuristic solutions is evaluated thanks to available lower bounds (min-RWA) in other studies. Other large instances have been solved with different heuristics or meta-heuristics in [6], [7], and [15], or with ILP models using a limited pre-computed set of paths and a rounding off technique to derive integer solutions, e.g., [16].

The paper is organized as follows. In the next section, we provide a detailed statement of the RWA problem (both static and dynamic), and we define the notations to be used throughout the paper. In Section III, we first recall the decomposition model (i.e., CG ILP model) that is based on maximal independent sets, and then propose two alternate reformulations of the pricing problem that is in charge of generating augmented wavelength configurations, i.e., subsets of lightpaths routed on a given wavelength, which can contribute to the improvement of the incumbent RWA solution. Section IV proposes several exact and heuristic solution schemes that take advantage of

the different formulations of the pricing problems of the previous section. Extensive numerical results are presented in Section VI, both for the static and the dynamic RWA problem. Conclusions are drawn in the last section.

II. STATEMENT OF THE RWA PROBLEM

Consider a WDM optical network represented by a multigraph $G = (V, L)$ with node set V indexed by v , where each node is associated with a node of the physical network, and with link set L indexed by ℓ where each link is associated with a fiber link of the physical network: the number of links from v to v' is equal to the number of fibers supporting traffic from v to v' . Connections and fiber links are assumed to be directional, and the traffic to be asymmetrical. The set of available wavelengths is denoted by Λ , and is indexed by λ with $W = |\Lambda|$. The traffic is defined by a $n \times n$ matrix D where D_{sd} defines the number of requested connections (i.e., wavelengths) from v_s to v_d . All wavelengths are assumed to have the same transport capacity. Let $\mathcal{SD} = \{(v_s, v_d) \in V \times V : D_{sd} > 0\}$ be the set of node pairs with traffic. Let $\omega^+(v)$ (resp. $\omega^-(v)$) be the set of outgoing (resp. incoming) fiber links at node v . We assume that the same wavelength is used from the source to the destination for all connection requests. Note that it has been shown (see [17], [18]) that wavelength conversion (i.e., multiple-hop connections) does not help very much in order to reduce the blocking rate (max-RWA problem).

The static RWA problem can then be formally stated as follows: given a multigraph G corresponding to a WDM optical network, and a set of requested connections, find a suitable lightpath (p, λ) for each granted connection, where a lightpath is defined by the combination of a routing path p and a wavelength λ , so that no two paths sharing a fiber link of G are assigned the same wavelength. We study the objective of minimizing the blocking rate, that is equivalent to maximizing the number of granted connections (also called Grade of Service or GoS for short), leading to the so-called max-RWA problem.

Incremental planning is performed periodically to decide how the transport capacity of a backbone optical network has to be updated to serve the monthly/yearly global growth traffic during the next planning period, see [19] and [1], [20] for some numbers about the growth of the traffic. The dynamic incremental RWA problem can be formally stated as follows: given a WDM optical network represented by a multigraph G , a matrix D^{LEGACY} containing already provisioned traffic, and a set of newly requested connections indicated by a matrix D^{NEW} , assign available lightpaths to the new incoming connection requests such that there is no wavelength clash between the legacy traffic and the newly granted connection requests. Considering the objective of maximizing the GoS leads to the dynamic max-RWA problem.

III. A DECOMPOSITION OPTIMIZATION MODEL

Several authors have already investigated modelling the RWA problem with a decomposition model, within the framework of exact solution schemes, see, e.g., [8], [19]. We revisit those models here, with the goal of enhancing them in order to

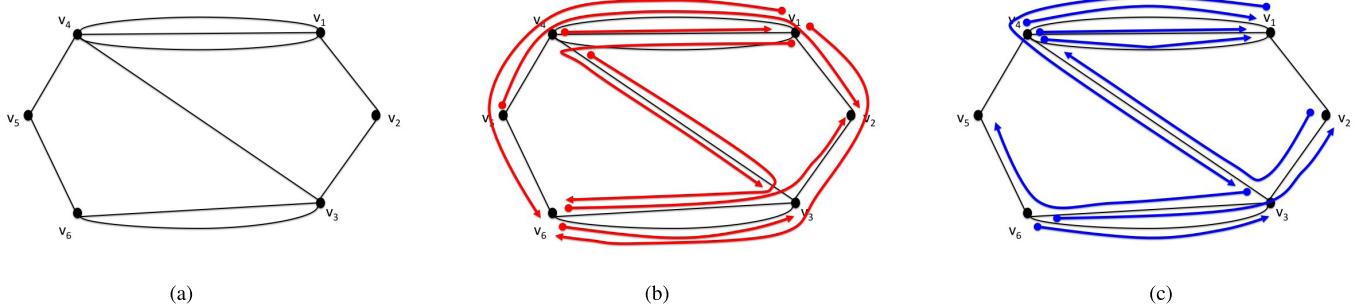


Fig. 1. Potential wavelength configurations. (a) Physical topology: each edge represents two directional fibers, one in each direction. (b) Potential wavelength configuration #1: it provides, e.g., 2 bandwidth units from v_1 and v_6 . (c) Potential wavelength configuration #2.

solve much larger RWA instances. We first recall the decomposition optimization model based on maximal independent set, as initially proposed by [8] and improved by [9], and then discuss how to improve it further.

A. Wavelength Configurations

Lee *et al.* [8] as well as Jaumard *et al.* [9] introduced the concept of independent routing configurations, where each configuration is associated with a set of paths that can be used for satisfying a given fraction of the connections with the same wavelength, see Figure 1. Within a wavelength configuration, routes must be pairwise link independent, i.e., they must not share a link as two lightpaths can not share a link if they use the same wavelength.

An independent routing configuration c can be formally represented by a non negative vector a^c such that $a_{sd}^c =$ number of connection requests from v_s to v_d that are supported by configuration c , with $a_{sd}^c \leq D_{sd}$ for $(v_s, v_d) \in \mathcal{SD}$. Each configuration is associated with a generic wavelength.

We denote by C the set of all possible wavelength configurations.

B. Static Case: Decomposition RWA Model

We use two sets of variables. The first set of variables, z_c , enables the selection of the best configurations and of their number of occurrences (i.e., to how many wavelengths they apply). The second set of variables, y_{sd} , compute the GoS for each node pair (v_s, v_d) , so that their sum provides the overall GoS.

The basic model is written as follows. Assuming that the configurations are at hand, the model selects the best wavelength configurations to maximize the grade of service (GoS), i.e., the number of granted connections.

$$\max \sum_{(v_s, v_d) \in \mathcal{SD}} y_{sd} \quad (1)$$

subject to:

$$\sum_{c \in C} z_c \leq W \quad (2)$$

$$y_{sd} \leq \sum_{c \in C} a_{sd}^c z_c \quad (v_s, v_d) \in \mathcal{SD} \quad (3)$$

$$y_{sd} \leq P_{sd} \quad (y_s, y_d) \in \mathcal{SD} \quad (4)$$

$$z \in \mathbb{Z}^+, c \in C \quad (5)$$

$$y_{i+1} \geq 0, \quad (x_i - x_{i+1}) \in S\mathcal{D} \quad (6)$$

Observe that, since $D_{sd} \in \mathbb{Z}^+$, we have $y_{sd} \in \mathbb{Z}^+$. However, we do not need to explicitly enforce it.

Constraints (2) ensure that we do not select more wavelength configurations than the number of available wavelengths. Constraints (3) compute the GoS for node pair (v_s, v_d) : equality is enforced with the combination of (1) and (3), and we do not explicitly enforce the equality constraints in order to ease the solution of the linear relaxation of (1)–(6). Constraints (4) prevent from granting more connections than requested. Constraints (5) and (6) define the domains of the variables.

Note that Model (1)–(6) can be written with only one set of variables in a very compact formulation, which is less self explanatory, as follows:

$$\max \sum_{c \in C} \sum_{(v_s, v_d) \in \mathcal{SD}} a_{sd}^c z_c \quad (7)$$

$$\text{subject to: } \sum_{c \in C} z_c \leq W \quad (8)$$

$$\sum_{c \in C} a_{sd}^c z_c \leq D_{sd} \quad (v_s, v_d) \in \mathcal{SD} \quad (9)$$

$$z_c \in \mathbb{Z}^+ \quad c \in C. \quad (10)$$

C. Dynamic Case: Decomposition RWA Model

Let D^{LEGACY} and D^{NEW} be the sets of already provisioned and newly received traffic, respectively. Note that the granted connections for the traffic requests of D^{LEGACY} have made some lightpaths unattainable for the connection requests of D^{NEW} . For these new connection requests, dynamic max-RWA problem tries to find lightpaths among the available ones in order to maximize the number of granted connections.

Comparing with the static case, there is a slight change in model (1)–(6), when modelling the dynamic RWA problem. Since some wavelengths have already been provisioned while granting the requests of D^{LEGACY} , unlike the static case, not all wavelengths have the same available set of links. Thus, unless they are not at all provisioned yet (Λ^{AVAIL}), the wavelengths ($\lambda \in \Lambda^{\text{USED}}$) and their associated set of configurations need to be labelled. Assuming the configurations are at hand, for every wavelength $\lambda \in \Lambda^{\text{USED}}$, a set of configurations C_λ is defined. C_λ corresponds to the set of potential configurations that have been defined considering wavelength $\lambda \in \Lambda^{\text{USED}}$. In every wavelength configuration $c \in C_\lambda$ with $\lambda \in \Lambda^{\text{USED}}$,

the set of available set of links is limited to those that do not belong to any lightpath with wavelength λ in the provisioning of the legacy traffic. The overall set of configurations is $C = \bigcup_{\lambda \in \Lambda^{\text{USED}}} C_\lambda \cup C^{\text{FREE}}$, where C^{FREE} is the set of configurations associated with $\lambda \in \Lambda^{\text{AVAIL}}$. As before, z_c represents the number of times configuration c occurs, except when it refers to $c \in C_\lambda$ with $\lambda \in \Lambda^{\text{USED}}$, in which case, it is restricted to $\{0, 1\}$: we must choose exactly one configuration among all possible ones associated with $c \in C_\lambda$ and $\lambda \in \Lambda^{\text{USED}}$. Configurations from C_λ all share the same lightpaths for the legacy traffic, but differ in the way they accommodate some newly incoming requests from D^{NEW} .

The dynamic RWA model is as follows:

$$\max \sum_{(v_s, v_d) \in \mathcal{SD}^{\text{NEW}}} y_{sd} \quad (11)$$

subject to:

$$\sum_{c \in C_\lambda} z_c = 1 \quad \lambda \in \Lambda^{\text{USED}} \quad (12)$$

$$\sum_{c \in C^{\text{FREE}}} z_c \leq W - |\Lambda^{\text{USED}}| \quad (13)$$

$$y_{sd} \leq \sum_{c \in C} a_{sd}^c z_c \quad (v_s, v_d) \in \mathcal{SD}^{\text{NEW}} \quad (14)$$

$$y_{sd} \leq D_{sd}^{\text{NEW}} \quad (v_s, v_d) \in \mathcal{SD}^{\text{NEW}} \quad (15)$$

$$z_c \in \{0, 1\} \quad c \in C_\lambda, \lambda \in \Lambda^{\text{USED}} \quad (16)$$

$$z_c \in \mathbb{Z}^+ \quad c \in C^{\text{FREE}} \quad (17)$$

$$y_{sd} \geq 0 \quad (v_s, v_d) \in \mathcal{SD}^{\text{NEW}}, \quad (18)$$

where $\mathcal{SD}^{\text{NEW}} = \{(v_s, v_d) \in V \times V : D_{sd}^{\text{NEW}} > 0\}$.

For every wavelength $\lambda \in \Lambda^{\text{USED}}$, constraints (12) restrict the number of chosen (enlarged) configurations to be 1. Constraint (13) makes sure that, considering the newly available wavelengths ($\lambda \in \Lambda^{\text{AVAIL}}$), we do not exceed the number of overall available wavelengths, i.e., W . Constraints (14)–(18) are equivalent to constraints (3)–(6) in the static case, except that variables z_c , $c \in C_\lambda$, $\lambda \in \Lambda^{\text{USED}}$ are limited to take binary values, and a_{sd}^c is equal to the number of granted traffic units of D_{sd}^{NEW} .

Note that the overall GoS is equal to $\text{GoS}^{\text{LEGACY}} + \sum_{(v_s, v_d) \in \mathcal{SD}^{\text{NEW}}} y_{sd}$.

IV. STATIC CASE: SOLUTION PROCESS AND ALGORITHMS

A. Implicit Enumeration of the Wavelength Configurations

The model proposed in III-B has an exponential number of variables, and therefore is not scalable if solved using classical ILP (Integer Linear Programming) tools. Indeed, we need to use column generation techniques in order to manage a solution process that only requires an *implicit* enumeration of the wavelength configurations (interested readers may refer to Chvatal [14]). Column generation method allows the exact solution of the linear relaxation of model (1)–(6), i.e., where constraints $z_c \in \mathbb{Z}^+$ are replaced by $z_c \geq 0$, for $c \in C$. It consists in solving alternatively a restricted master problem (the model of III-B with a very limited number of columns/variables) and the pricing problem (generation of a

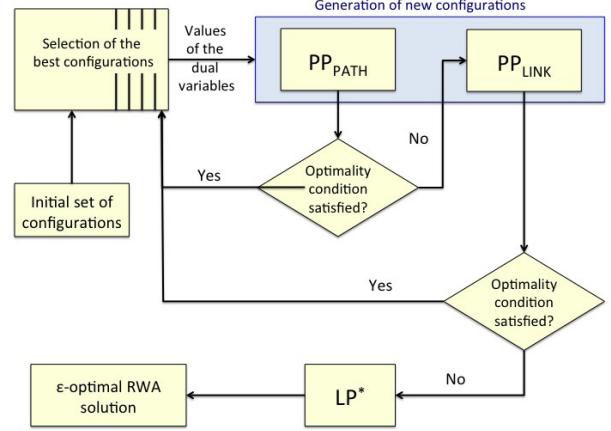


Fig. 2. Generic flowchart for all algorithms.

new wavelength configuration) until the optimality condition is satisfied (i.e., no wavelength configuration with a negative reduced cost). In other words, when a new wavelength configuration is generated, it is added to the current restricted master problem only if its addition implies an improvement of the optimal value of the current restricted master problem. This condition, indeed an optimality condition, can be easily checked with the sign of the reduced cost, denoted by COST , see (19) for its expression (the reader who is not familiar with linear programming concepts is referred to [14]), of variables z_c .

Once the optimal solution of the LP (Linear Programming) relaxation (z_{LP}^*) has been reached, we solve exactly the last restricted master problem, i.e., the restricted master problem of the last iteration in the column generation solution process, using a branch-and-bound method, leading then to an ϵ -optimal ILP solution (\tilde{z}_{ILP}), where

$$\epsilon = \frac{z_{\text{LP}}^* - \tilde{z}_{\text{ILP}}}{z_{\text{LP}}^*}.$$

Branch-and-price methods can be used in order to find optimal solutions, if the accuracy (ϵ) is not satisfactory, see, e.g., [9], [21].

B. ϵ -Optimal Algorithms CG, CG+ and CG++

In the context of the present study, we investigated different algorithms, in which the differences lie in the generation process of new augmenting configurations, i.e., of configurations that give rise to an improvement of the value of the current restricted master problem, when solving its linear relaxation.

The first algorithm, denoted by CG, relies on a mathematical model with a link formulation, in order to generate new augmented wavelength configurations. In the context of column generation, the configuration generator corresponds to the so-called pricing problem. The pricing problem with a link formulation will be called PP_LINK (see Section IV-D for its detailed description), and corresponds to the algorithm used in [9]. The flowchart of algorithm CG is represented in Figure 2, under the assumption that box PP_PATH is omitted.

Based on the observation made by several researchers, and investigated later in Section VI, that a very high percentage of lightpaths are supported by shortest paths, or k -shortest paths with a small k , we propose to investigate a path formulation, called PP_{PATH}, for the pricing problem (see Section IV-E for its detailed description), with different strategies for selecting the paths. Since we cannot consider all possible paths, otherwise the pricing problem would not be scalable, we need to combine the use of PP_{PATH} with PP_{LINK} in order to get an ε -optimal algorithm. Indeed, when PP_{PATH} is no more able to output an augmenting wavelength configuration, we switch to PP_{LINK}, and check whether it is still possible to generate an augmented wavelength configuration using more diverse paths than those considered in PP_{PATH}. Flowchart of the corresponding algorithm is represented in Figure 2.

We investigated two variants for PP_{PATH}. In the first one, we consider only the shortest paths, and in the second one, we consider the shortest paths, as well as a selection of k -shortest paths (see Section IV-F for how we made the selection). The resulting ε -optimal algorithms are called CG+ and CG++, respectively.

C. Heuristic Algorithms CG^H+ and CG^H++

We derive two heuristic algorithms from the CG+ and CG++ algorithms, with the elimination of the recourse to PP_{LINK} in order to limit the computational times. The resulting heuristic algorithms are called CG^H+ and CG^H++ and are summarized in the flowchart represented in Figure 2, assuming PP_{LINK} box is not used. Both CG^H+ and CG^H++ are associated with different path selections.

In the next two sections, we provide the detailed mathematical formulations of PP_{LINK} and PP_{PATH}.

D. Pricing Problem - Link Formulation

As always with the column generation method, the objective of the pricing problem (i.e., generator of new wavelength configurations) is the reduced cost ($\overline{\text{COST}}_c^{\text{LINK}}$) of variable z_c . In order to alleviate the notations, index c will be omitted in the remainder of this section.

Let $u^{(2)}$ and $u_{sd}^{(3)}$ be the values of the dual variables associated with constraints (2) and (3) in the optimal solution of the linear relaxation of the current restricted master problem (see the flowchart in Figure 2). Consider the following set of variables: $\alpha_\ell^{sd} = 1$ if link ℓ is used in a route from v_s to v_d , 0 otherwise. The link formulation of the pricing problem can be written as follows:

Wavelength Configuration Generator - Link Model PP_{LINK}

$$\max \quad \overline{\text{COST}}^{\text{LINK}} = -u^{(2)} + \sum_{(v_s, v_d) \in \mathcal{SD}} \sum_{\ell \in \omega^+(v_s)} \alpha_\ell^{sd} u_{sd}^{(3)} \quad (19)$$

subject to:

$$\sum_{(v_s, v_d) \in \mathcal{SD}} \alpha_\ell^{sd} \leq 1 \quad \ell \in L \quad (20)$$

$$\sum_{\ell \in \omega^+(v)} \alpha_\ell^{sd} = \sum_{\ell \in \omega^-(v)} \alpha_\ell^{sd} \quad (v_s, v_d) \in \mathcal{SD}, \quad v \in V \setminus \{v_s, v_d\} \quad (21)$$

$$\sum_{\ell \in \omega^+(v_s)} \alpha_\ell^{sd} \leq D_{sd} \quad (v_s, v_d) \in \mathcal{SD} \quad (22)$$

$$\sum_{\ell \in \omega^-(v_s)} \alpha_\ell^{sd} = \sum_{\ell \in \omega^+(v_d)} \alpha_\ell^{sd} = 0 \quad (v_s, v_d) \in \mathcal{SD} \quad (23)$$

$$\alpha_\ell^{sd} \in \{0, 1\} \quad \ell \in L, (v_s, v_d) \in \mathcal{SD}. \quad (24)$$

Constraints (20) ensure wavelength continuity, i.e., that a link cannot be traversed by more than one route in any given wavelength configuration. Routes are established with the help of the flow conservation constraints (21): if no route is selected for node pair (v_s, v_d) , then $\alpha_\ell^{sd} = 0$ for all links $\ell \in L$, otherwise, the sum of the outgoing flow values at the source node ($\sum_{\ell \in \omega^+(v_s)} \alpha_\ell^{sd}$) gives the number of link-disjoint routes from v_s to v_d in the wavelength configuration under construction. Constraints (22) avoid exceeding the demand in terms of the number of lightpaths. Constraints (23) prevent loops around the source or the destination nodes from arising. Constraints (24) define the domain of variables α_ℓ^{sd} .

Correspondence between variables of the pricing problem and coefficients of the master problem:

$$a_{sd} = \sum_{\ell \in \omega^+(v_s)} \alpha_\ell^{sd}. \quad (25)$$

Observe that nothing prevents from selecting several pairwise link disjoint paths for a given pair (v_s, v_d) of source and destination nodes. Indeed, a_{sd} is equal to the number of link-disjoint paths from v_s to v_d in the configuration under construction.

E. Pricing Problem - Path Formulation

In the path formulation, we provide a set P_{sd} of paths for each source and destination pair of nodes, see Section IV-F for the definition of P_{sd} .

The path formulation for the wavelength configuration generator is denoted by PP_{PATH}. It uses the set of decision variables: $\beta_p^{sd} = 1$ if path p is used in the wavelength configuration under construction, 0 otherwise. PP_{PATH} is written as follows:

Wavelength Configuration Generator - Path Model PP_{PATH}

$$\max \quad \overline{\text{COST}}^{\text{PATH}} = -u^{(2)} + \sum_{(v_s, v_d) \in \mathcal{SD}} \sum_{p \in P_{sd}} \beta_p^{sd} u_{sd}^{(3)} \quad (26)$$

subject to:

$$\sum_{(v_s, v_d) \in \mathcal{SD}} \sum_{p \in P_{sd}} \delta_\ell^p \beta_p^{sd} \leq 1 \quad \ell \in L \quad (27)$$

$$\sum_{p \in P_{sd}} \beta_p^{sd} \leq D_{sd} \quad (v_s, v_d) \in \mathcal{SD} \quad (28)$$

$$\beta_p^{sd} \in \{0, 1\} \quad \ell \in L, (v_s, v_d) \in \mathcal{SD}. \quad (29)$$

Pairwise link disjointness for paths is guaranteed thanks to constraints (27), in which δ_ℓ^p is a binary value representing the presence of link ℓ in path p . Constraints (28) enforce not to exceed the lightpath demand. Constraints (29) define the domain of variables β_p^{sd} .

Correspondence between variables of the pricing problem and coefficients of the master problem:

$$a_{sd} = \sum_{p \in P_{sd}} \beta_p^{sd}.$$

F. Computation and Selection of k-Shortest Paths

In order for the proposed CG++ and CG^H++ solution algorithms to be effective, we must carefully choose the paths to consider: a sufficiently large number so that we can maximize the GoS, but not too many in order not to increase too much the size of the constraint matrix in PP_{PATH}. In addition, considering paths that are much longer than the shortest paths may lead to an inefficient use of the spectrum.

We explore several strategies in which paths are first selected in the set of shortest paths, and next in the set of k -shortest paths. Very efficient algorithms already exists for enumerating such paths, see [22], [23], and we use the open library [24] that provides an implementation of Yen's algorithm [22]. We denote by P_{sd}^1 the set of all shortest paths from v_s to v_d , and then by P_{sd}^k the set of paths with the k shortest distinct value. For instance, if the ordered list according to the length of paths is: p_1, p_2, \dots, p_9 of length 1, 1, 2, 2, 2, 4, 4, 4, 4 respectively, then $P^1 = \{p_1, p_2\}$, $P^2 = \{p_3, p_4, p_5\}$ and $P^3 = \{p_6, p_7, p_8, p_9\}$.

1) **Strategy 1:** We consider the same number of k -shortest paths, in addition to the shortest paths, for each node pair, say k^{sp} . After extensive numerical experiments on different data sets with various network topologies, a good compromise we found between computing times and best GoS was $k^{sp} = 15$.

2) **Strategy 2:** Drawbacks of Strategy 1 is: (i) to choose at random the paths in the last P^i set that is considered for reaching the number of selected paths, and (ii) to select a number of paths that is independent of the traffic demand. So, we modify the criterion for selecting the best paths in the last considered P^i as follows: Firstly, taking into account traffic demand, we select the number of paths with the following fairness criterion: $(D_{sd}/\text{number of candidate paths from } v_s \text{ to } v_d) = \rho$, where experiments showed that a good compromise between the computing times and the maximum GoS led to a ρ constant value equal to .5. Secondly, we enumerate all the paths p of P^i and order them in the increasing order of

$$\sum_{\ell \in p} \text{LOAD}_\ell,$$

where LOAD_ℓ is an estimate on the number of lightpaths going through ℓ when maximizing the GoS using the following routing formulation that ignores the wavelength continuity constraints and omits integrality requirements for the φ_ℓ^{sd} variables:

$$\max \sum_{(v_s, v_d) \in \mathcal{SD}} d_{sd} \quad (30)$$

subject to:

$$\sum_{\ell \in \omega^+(v_s)} \varphi_\ell^{sd} = \sum_{\ell \in \omega^-(v_d)} \varphi_\ell^{sd} = 0 \quad (v_s, v_d) \in \mathcal{SD} \quad (31)$$

$$\sum_{\ell \in \omega^+(v_s)} \varphi_\ell^{sd} = \sum_{\ell \in \omega^+(v_d)} \varphi_\ell^{sd} = d_{sd} \quad (v_s, v_d) \in \mathcal{SD} \quad (32)$$

$$\sum_{\ell \in \omega^+(v)} \varphi_\ell^{sd} = \sum_{\ell \in \omega^-(v)} \varphi_\ell^{sd} \quad v \in V \setminus \{v_s, v_d\}, \\ (v_s, v_d) \in \mathcal{SD} \quad (33)$$

$$\sum_{(v_s, v_d) \in \mathcal{SD}} \varphi_\ell^{sd} \leq W \quad \ell \in L \quad (34)$$

$$\varphi_\ell^{sd} \geq 0 \quad \ell \in L, (v_s, v_d) \in \mathcal{SD}. \quad (35)$$

$$0 \leq d_{sd} \leq D_{sd} \quad (v_s, v_d) \in \mathcal{SD}. \quad (36)$$

Each variable d_{sd} represents the fraction of the demand that is granted for node pair (v_s, v_d) , and each variable φ_ℓ^{sd} defines the amount of traffic going through ℓ with respect to the demand of node pair (v_s, v_d) .

Note that constraints (31)–(33) are multi-commodity flow constraints.

3) **Strategy 3:** Network topologies are usually mesh ones, and consequently, the lengths of the shortest paths vary from one node pair to the next. In addition, the traffic is not uniform. So, in this third strategy, we consider a number of paths that is larger if the traffic demand is larger, and larger as the length of the shortest path is longer: we generate the first $(D_{sd}/\text{length of shortest paths}) = (D_{sd}/\ell_{sd}^{\text{short}})$ paths, where ℓ_{sd}^{short} is the length of the shortest path from v_s to v_d . When $\ell_{sd}^{\text{short}} = 1$, we only select the one hop paths made of (v_s, v_d) (note that there might be more than one if there are more than one fiber link). Again, we choose the paths of the last considered P^i set according to the same criterion as in Strategy 2.

Observe that we can force the lightpath of one hop requests to be routed on a one-hop route without loss of generality, thanks to the following result.

Theorem 1: Consider the max-RWA problem. There is at least one optimal solution in which one hop requests are provisioned (lightpaths) on one hop path(s) if they can be granted, i.e., if $D_{sd} \leq W r_{sd}$, where r_{sd} is the number of links from v_s to v_d .

Proof: Consider a pair $(v_s, v_d) \in \mathcal{SD}$ with $0 < D_{sd} \leq r_{sd}W$ such that v_s and v_d are connected in G by at least one link. Denote it by ℓ_{sd} one of the links connecting v_s to v_d .

Assume that there is an optimal RWA solution such that D_{sd} is not routed completely on links from P_{sd}^1 , i.e., on one-hop lightpaths. We next show that such an optimal solution can be modified in order that it satisfies the property stated in the theorem.

Four cases need to be distinguished, and the transformation described for each of them can be repeated on each link from v_s to v_d possibly several times, until the optimal solution satisfies the stated property. Indeed, each transformation reduces the number of demands that do not satisfy the property by one unit at a time.

- **There is at least one D_{sd} demand that is routed on a path with at least two hops and ℓ_{sd} has at least one available wavelength:** Rerouting that latter demand on ℓ_{sd} does not change the GoS and reduces by one unit the demand that does not satisfy the stated property.
- **There is at least one D_{sd} demand, say k , that is routed on a path with at least two hops and all wavelengths of ℓ_{sd} (as well as on all links from v_s to v_d) are used in a lightpath.** Assume that k is provisioned with lightpath (p, λ) . Observe that $\ell_{sd} \notin p$ as otherwise p is not a simple path (i.e., without loop). Since all wavelengths of ℓ_{sd} are used, there exists a multi-hop lightpath (p', λ) , with $\ell_{sd} \in p'$, which is used to provision a request, say k' .

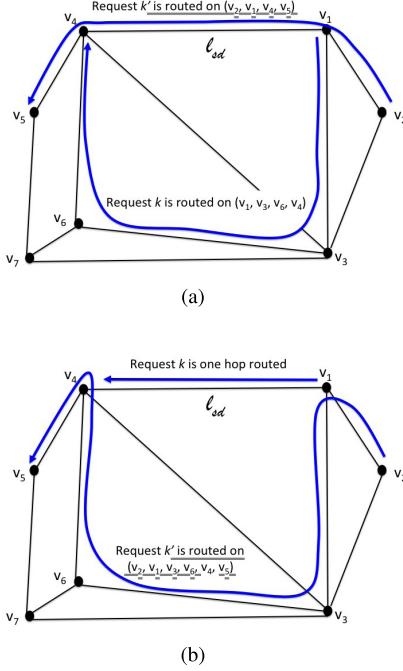


Fig. 3. Lightpath modifications. (a) Original routing of the lightpaths. (b) Modified lightpaths.

For example, in the left network of Figure 3, $k = (v_1, v_4)$ and $k' = (v_2, v_5)$. Modify the provisioning of k' so that, instead of using ℓ_{sd} with wavelength λ , it uses the multi-hop lightpath of k with wavelength λ . Re-provision k on the one-hop lightpath (ℓ_{sd}, λ) . The fraction of D_{sd} that does not satisfy the stated property, reduces by one unit. See Figure 3 for an example of the provisioning transformations. Wavelength continuity is respected, as both lightpaths (p, λ) and (p', λ) are using the same wavelength.

- There is at least one D_{sd} demand that is denied and ℓ_{sd} has at least one available wavelength: Contradicts the assumption that the solution is optimal, therefore that case can be omitted.
- There is at least one D_{sd} demand that is denied and ℓ_{sd} (as well as on all links from v_s to v_d) are used in a lightpath: then, there exists a one multi-hop, say k , which is routed through a link from v_s to v_d . Assume it is on ℓ_{sd} . Denying k and granting one demand unit of D_{sd} on ℓ_{sd} reduces by one unit the demand that does not satisfies the stated property. \square

V. DYNAMIC CASE: SOLUTION PROCESS AND ALGORITHMS

As stated in Subsection III-C, the input of the dynamic RWA problem is a set of legacy traffic (D^{LEGACY}), and a set of new connection requests (D^{NEW}). In the solution scheme for dynamic RWA, lightpaths are assigned to the new incoming connection requests, using model (11)–(18). As before, due to the exponential number of possible configurations, not all of them are explicitly enumerated; rather a restricted master

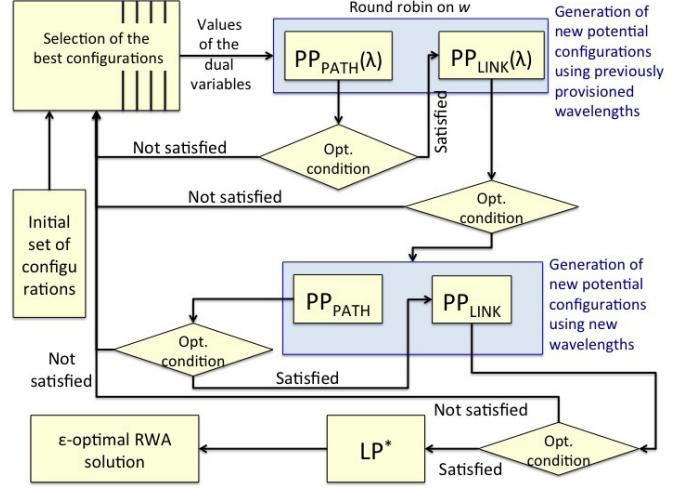


Fig. 4. Network topologies. (a) NSF. (b) USA. (c) GER. (d) NTT. (e) ATT. (f) BRAZIL.

problem is built with a small number of configurations and a wavelength configuration generator problem that implicitly considers them all, keeps on adding new augmented configurations until an optimality condition is satisfied. Once the linear relaxation of (11)–(18) is optimally solved, an integer ε -optimal solution is derived.

The details of the proposed algorithm for solving dynamic max-RWA problem are as follows. Let \mathcal{LP} be the set of lightpaths used to grant connections for the legacy traffic. At the initial step, for every wavelength λ , the set of available links L_λ is:

$$L_\lambda = L \setminus \{\ell \in L : \exists p \text{ with } \ell \in p \text{ and } (p, \lambda) \in \mathcal{LP}\}. \quad (37)$$

In the next step, the dynamic max-RWA problem is solved using a slightly modified version of the solution schemes presented for the static RWA problem. For path-based pricing problems arising in the algorithms such as CG+ and CG++, let P_{sd}^λ be the set of pre-computed paths for the pair (v_s, v_d) , obtained by considering L_λ instead of L . Sets L_λ and P_{sd}^λ may vary from one wavelength to the next, consequently problems PP_{LINK} and PP_{PATH} have to be distinguished for already provisioned wavelengths (Λ^{USED}). Therefore, $|\Lambda^{\text{USED}}|$ distinctive PP_{LINK} and PP_{PATH} are defined: when indexed with λ , they use L_λ and P_{sd}^λ instead of L and P_{sd} , leading to pricing problems $PP_{\text{LINK}}(L_\lambda)$ and $PP_{\text{PATH}}(L_\lambda, P_{sd}^\lambda)$. For the wavelengths that are newly available ones, i.e., set Λ^{AVAIL} , generic PP_{LINK} and PP_{PATH} are solved in every iteration, using L and P_{sd} , i.e., $PP_{\text{LINK}}(L)$ and $PP_{\text{PATH}}(L, P_{sd})$.

In summary, the overall solution algorithm computes the sets L_λ and P_{sd}^λ for all $\lambda \in \Lambda^{\text{USED}}$ and all pairs $(v_s, v_d) \in \mathcal{SD}^{\text{NEW}}$. Then, starting with the initial configurations, model (11)–(18) restricted to a small subset of configurations and pricing problems PP_{LINK} and PP_{PATH} are solved alternatively, until the optimality condition is satisfied. In choosing the wavelength for consideration in each iteration, a round robin approach is employed. The general procedure is illustrated in Figure 5.

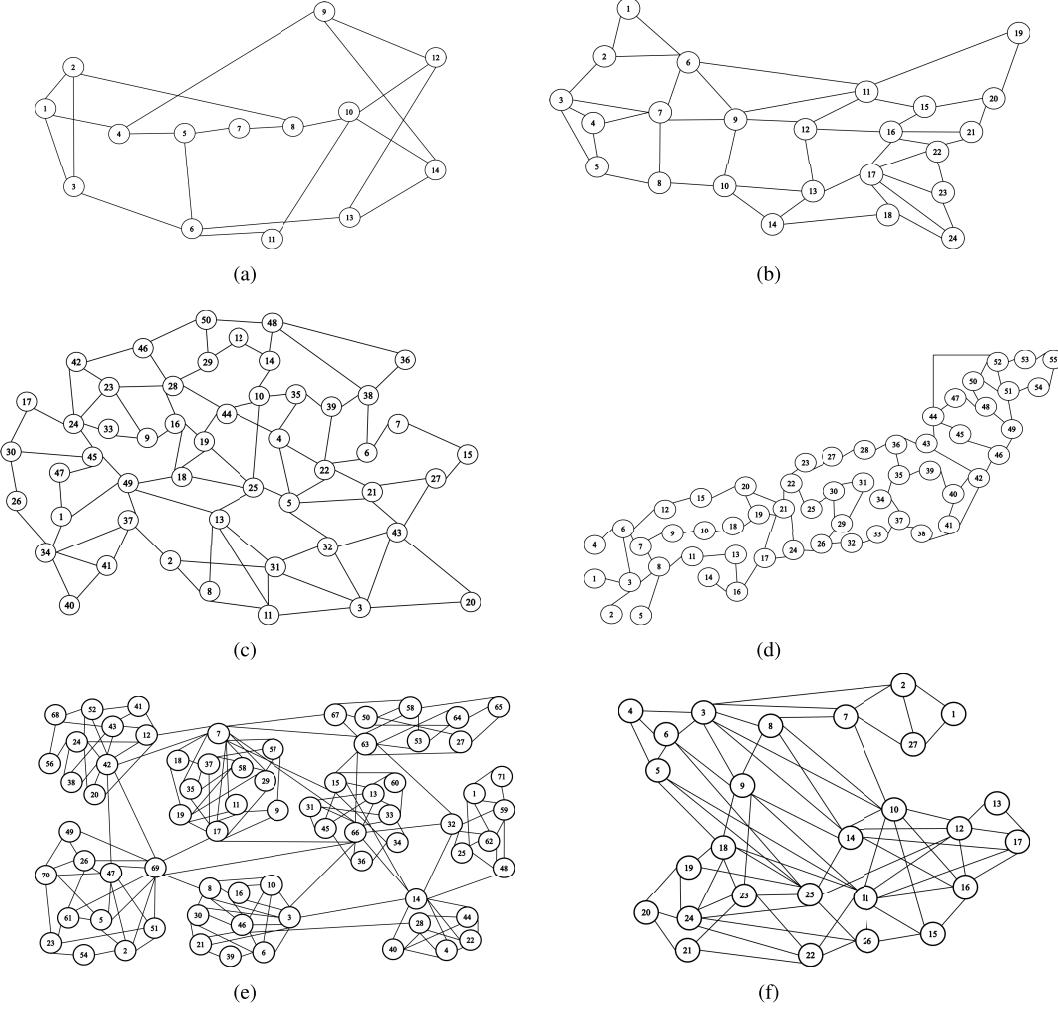


Fig. 5. Solution approach for dynamic requests. (a) NSF. (b) USA. (c) GER. (d) NTT. (e) ATT. (f) BRAZIL.

Note that a unique generic pricing problem is solved for all wavelengths $\lambda \in \Lambda^{\text{AVAIL}}$. In other words, at every round of the round robin algorithm, we solve distinct pricing problems $\text{PP}_{\text{LINK}}(L_\lambda)$ and $\text{PP}_{\text{PATH}}(L_\lambda, P_{sd}^\lambda)$ for wavelengths $\lambda \in \Lambda^{\text{USED}}$, and a unique pair of generic pricing problems, i.e., $\text{PP}_{\text{LINK}}(L)$ and $\text{PP}_{\text{PATH}}(L, P_{sd})$ for the remaining available wavelengths, see [25] for the detailed descriptions of the various pricing problems.

VI. NUMERICAL RESULTS

We present here the numerical experiments that were conducted in order to validate and test the performance of the proposed algorithms: the three ε -optimal algorithms, CG, CG+, and CG++, as well as the two heuristic algorithms, CG^H+ and CG^H++. After describing the data sets (Section VI-A), we discuss the results for the static case (Section VI-B), starting with the quality of the solutions (Section VI-B.1), exact vs. heuristic solutions (Section VI-B.3), and then the impact of the path selection on the GoS (Section VI-B.4), as well as the usage of the bandwidth spectrum from one link to the next (Section VI-B.5). The remaining results refer to the dynamic case (Section VI-C) where we investigate the bandwidth spectrum waste when no rearrangements are performed.

All computational results have been obtained with running the programs on a server with the help of CPLEX [26] (Version V12.6.2) for solving the (integer) linear programs¹. Programs never used more than 2Gb memory and 2 CPUs.

A. Data Sets

We run experiments on six different networks: NSFNET [27], USANET [28], GERMANY [27], NTT [29], ATT [5], and BRAZIL [18], whose characteristics are reported in Table II. Column entitled “deg.” is the average nodal degree, in order to measure the network connectivities. Last two columns report on the traffic distribution thanks to the mean and the variance on the number of requests per node pair with traffic, as identified by μ and σ , respectively. Topologies are reproduced in Figure 4 where undirected lines represent bidirectional links.

For each network topology, we consider various traffic instances with up to 150 wavelengths. For the first traffic

¹Several fine tuning of cplex parameters are required for solving the data sets as efficiently as possible. These include switching off the presolve operations, restricting the number of threads to 1, solving the problems using Barrier algorithms, setting the emphasis of the pricing problem on feasibility rather than optimality, and switching back the solver of pricing problems to traditional branch-and-bound, rather than dynamic search.

TABLE II
CHARACTERISTICS OF THE DATASETS

Data instances	$ V $	$ L $	deg.	W	$ \mathcal{SD} $	$\sum D_{sd}$	Traffic distribution μ	σ
NSF ₃₀				30	141	436	3.1	1.4
NSF ₇₅	14	40	3.0	75	182	1,371	7.5	2.4
NSF ₁₁₅				115	182	2,194	12.1	2.7
USA ₇₅				75	455	1,336	2.9	2.4
USA ₁₂₅	24	88	3.7	125	541	2,422	4.5	1.9
USA ₁₅₀				150	552	3,509	6.4	2.2
GER ₁₀₀				100		2,365	3.6	6.2
GER ₁₃₀	50	176	3.5	130	660	3,041	6.2	4.6
GER ₁₅₀				150		4,989	8.6	6.3
NTT ₄₂				42	338	1,038	3.1	1.4
NTT ₅₀	55	144	2.6	50	452	1,362	3.0	1.4
NTT ₁₅₀				150	452	5,684	12.6	7.4
ATT ₂₀	90	274	3.0	20	272	359	1.3	0.7
ATT ₁₁₃	71	350	4.9	113	2,869	2,918	1.0	0.7
BRAZIL ₄₈	27	140	5.2	48	549	1,370	2.5	1.1

instances of the NSF and USA topologies, the directed traffic demand matrix $T = [T_{sd}]$ is generated by drawing integer traffic demands (with one unit being the transport capacity of one wavelength) uniformly at random in $\{0, 1, 2, 3, 4, 5\}$. For the GERMANY topology, the first traffic instance (i.e., GER₁₀₀) comes from the snd-lib library [27]. For the NTT topology, the first two traffic instances (i.e., NTT₄₂ and NTT₅₀) are from [29]. The next augmented traffic instances correspond to incremental traffic: NSF/USA/GER/NTT _{λ} \subseteq NSF/USA/GER/NTT _{λ'} where NSF/USA/GER _{λ'} are built upon NSF/USA/GER _{λ} by REPEAT times randomly adding ALEA more requests for each pair of nodes. ALEA is taken uniformly at random from $\{1, 2, 3, 4, 5\}$ for NSF/USA and from $\{0, 1, 2, 3\}$ for GERMANY. For NSF, REPEAT = 10, 19, for USA, REPEAT = 4, 8 and for GERMANY, REPEAT = 1, 5. NTT₁₅₀ is built directly upon NTT₅₀, considering a random additional number of requests from $\{1, \dots, 25\}$, which leads to a less uniform traffic instance, as shown by the variance indicator in the last column of Table II. Traffic matrix of BRAZIL comes from [18]. We selected a limited number of traffic instances when reporting on the numerical experiments, their characteristics are described in Table II, the index of the instance names refer to the number of wavelengths.

B. Static Case: Algorithm Comparative Performances

1) *Solution Accuracies and Algorithm Efficiencies:* In Tables III and IV, we compare the performances of the four new proposed algorithms and the algorithm of [9], using Strategy 3 for selecting the paths.

In Table III, we focus on the comparison of the maximum GoS, and the number of generated wavelength configurations, using either PP_{PATH} or PP_{LINK}. While there is no dominance in terms of best generated ε -optimal solutions between CG and CG+, algorithm CG++ dominates both CG and CG+ for all data sets, increasing the number of granted demand from 2 (NSF₃₀) to 71 (NTT₁₅₀) lightpaths.

In Table IV, we report the number of columns in the last iteration of the column generation algorithms, and the computational times (in seconds). Reported computational times (in columns entitled CPU) do not include the preprocessing

operations, which is reported in the columns entitled PRE_CPU. While in most cases CG+ is much faster than CG, it requires more iterations. However, as the pricing problem restricts the search of wavelength configurations using pre-computed shortest paths most of the time (indeed, 97% of the time on average), then computational times can be significantly reduced.

CG++ requires more computational times than CG+, in all smaller data sets even if there are less wavelength configurations generated. This is due to the fact that PP_{PATH} contains more paths and therefore requires longer computational times for its solution. However, the number of calls to PP_{LINK} is less for CG++ than for CG+. This is as expected since PP_{PATH} is richer in terms of the number of paths that are considered. The effect of this smaller number of calls to PP_{LINK} is revealed in the largest data sets, i.e., GER network, NTT₁₅₀, ATT and BRAZIL. In Table III, it can be seen that in the latter data sets, the number of calls to PP_{LINK} is much smaller for CG++ than for CG+ ; it results in significantly smaller computational times.

2) *Comparison of CG, CG+, and CG++:* We observe in Table III that the number of wavelength configuration generations with the link formulation, i.e., PP_{LINK}, which is more computationally expensive than PP_{PATH}, decreases, as we move from CG to CG+, and then to CG++. Except for NTT₁₅₀, the number of PP_{LINK} solutions is very small for CG++, contributing significantly to the overall reduction of the computational effort for CG++.

As the size of the network topology increases, i.e., moving from NSF to USA topology, while CG performs better than CG+ in terms of GoS, the differences between the computational times widen: on average, computational times for CG are 43 times longer than for CG+. It is interesting to observe that except for USA₇₅, we recourse to PP_{LINK} only one time, meaning that the solution provided by the wavelength configurations of PP_{PATH} account for most of the computational times, and explain the reduction of it in comparison with algorithm CG. CG++ improves on CG+, while requiring to generate less wavelength configurations for USA₇₅ and USA₁₃₀, but more wavelength configurations for the largest USA data set USA₁₅₀. This can be explained by the larger number of considered k -shortest paths.

In GERMANY network, except for one instance, CG+ gets better solutions than CG, with an increased number of lightpaths ranging from 12 (in addition to 2,185 in GER₇₅) to 49 (in addition to 2,791 in GER₁₃₀). Computational times of CG in most cases are higher than those of CG+, i.e., on average 5 times larger. As for previous networks, CG++ always returns better GoS results than both CG and CG+. It is interesting to see that CG++ succeeds in obtaining better results in considerably shorter time than CG+, which itself performs dramatically faster than CG. In fact, the computational times of CG are on average 13 times longer than that of CG++, which is a significant improvement. This can be explained by the low number of PP_{LINK} solutions in CG++ (from 3 to at most 21), that can make up for the larger computing times of its PP_{PATH} problem in comparison to that of CG+.

NTT network has an interesting feature and that is the degree of its nodes, which in most cases is equal to 2.

TABLE III
STATIC CASE: COMPARATIVE PERFORMANCE OF THE ALGORITHMS

Data instances	z^* _{LP}	ε -optimal algorithms								Heuristic algorithms								
		CG [9]			CG+			CG++		CG ^{H+}			CG ^{H++}					
		\tilde{z} _{ILP}	ε	#PP _{LINK}	\tilde{z} _{ILP}	ε	#PP _{iter}	\tilde{z} _{ILP}	ε	PATH	LINK	\tilde{z} _{ILP}	ε	#PP _{PATH}	\tilde{z} _{ILP}	ε	#PP _{PATH}	
NSF30	430	412	4.2	85	419	2.6	158	6	421	2.1	140	3	410	4.7	88	411	4.4	100
NSF75	1,242	1,218	1.9	109	1,221	1.7	189	7	1,230	1.0	217	3	1,198	3.5	126	1,222	1.6	108
NSF115	1,924	1,898	1.4	92	1,900	1.2	167	6	1,909	0.8	217	2	1,854	3.6	104	1,904	1.0	108
USA75	1,281	1,229	4.1	249	1,227	4.2	378	2	1,241	3.1	402	2	1,227	4.2	345	1,234	3.7	253
USA125	2,255	2,190	2.9	324	2,160	4.2	315	1	2,201	2.4	458	2	2,160	4.2	315	2,160	4.2	299
USA150	3,029	2,914	3.8	252	2,885	4.8	255	1	2,975	1.8	508	2	2,885	4.8	255	2,885	4.8	258
GER100	2,306	2,185	5.2	349	2,197	4.7	2,850	42	2,245	2.7	537	3	2,099	9.0	236	2,206	4.3	336
GER130	2,960	2,791	5.7	437	2,840	4.1	3,543	104	2,889	2.4	639	3	2,658	10.2	192	2,798	5.5	414
GER150	4,663	4,472	4.1	731	4,502	3.5	6,253	119	4,519	3.1	937	6	4,219	9.5	336	4,466	4.2	711
NTT42	1,038	1,031	0.7	16	1,024	2.3	75	2	1,038	0.0	31	2	1,022	1.5	57	1,025	1.3	20
NTT50	1,362	1,356	0.4	32	1,310	3.8	287	5	1,362	0.0	59	2	1,308	4.0	112	1,318	3.2	55
NTT150	5,553	5,431	2.2	314	5,372	3.3	2,292	55	5,502	0.9	867	21	5,204	3.6	277	5,431	2.2	311
ATT20	359	328	8.6	118	328	8.6	2,571	62	354	1.4	380	7	272	24.2	113	321	10.6	71
ATT113	2,918	2,856	2.1	319	2,884	1.2	7,303	112	2,902	0.5	397	6	2,740	6.1	1,182	2,841	2.6	211
BRAZIL ₄₈	1,370	1,297	5.3	168	1,295	5.5	895	17	1,317	3.9	313	2	1,273	7.0	195	1,301	5.0	156

TABLE IV
STATIC CASE: COMPUTATIONAL TIMES (SECONDS) OF THE ALGORITHMS

Data instances	CG				ε -optimal algorithms						Heuristic algorithms					
	\tilde{z} _{ILP}	GoS	CPU	# cols.	CG+			CG++			CG ^{H+}			CG ^{H++}		
					PRE_CPU	CPU	# cols.	PRE_CPU	CPU	# cols.	CPU	# cols.	CPU	# cols.	CPU	# cols.
NSF30	412	94.5	20	84	0	3	157	2	7	139	1	87	6	99		
NSF75	1,218	88.7	26	108	0	3	188	3	7	216	2	125	5	107		
NSF115	1,898	86.5	24	91	0	3	166	3	7	216	1	103	5	107		
USA75	1,229	92.0	663	248	1	34	377	12	129	401	27	344	70	252		
USA125	2,190	90.4	1,211	323	1	21	314	17	138	457	21	314	106	298		
USA150	2,914	83.0	890	251	1	17	254	21	155	507	15	254	87	257		
GER100	2,185	92.4	6,045	348	2	885	2,849	47	427	536	13	235	161	335		
GER130	2,791	91.8	7,050	436	2	2,111	3,542	54	467	638	12	191	108	413		
GER150	4,472	78.9	16,863	730	2	3,448	6,252	70	1,747	936	23	335	283	710		
NTT42	1,031	99.3	49	15	0	4	74	25	5	30	1	56	5	19		
NTT50	1,356	99.6	161	31	1	22	286	30	7	58	2	111	8	54		
NTT150	5,431	95.5	2,403	313	1	686	2,291	48	183	866	10	276	81	310		
ATT20	328	91.3	1,644	117	1	1,917	2,570	76	1,473	379	8	112	131	70		
ATT113	2,877	97.9	26,035	318	9	3,807	7,302	714	893	396	1,018	1,181	273	210		
BRAZIL ₄₈	1,297	94.7	1,145	167	1	1,191	894	23	563	312	195	181	44	155		

TABLE V
COMPARISON OF VARIOUS STRATEGIES FOR SELECTING THE SET OF PATHS IN CG++

Data Instances	CG algorithm		CG++ algorithm											
	z^* _{LP}	\tilde{z} _{ILP}	Strategy 1				Strategy 2				Strategy 3			
			\tilde{z} _{ILP}	PP _{PATH}	PP _{LINK}	CPU	\tilde{z} _{ILP}	PP _{PATH}	PP _{LINK}	CPU	\tilde{z} _{ILP}	PP _{PATH}	PP _{LINK}	CPU
NSF30	430	412	415	127	2	29	415	110	1	14	421	140	3	7
NSF75	1,242	1,218	1,222	108	1	3	1,222	127	1	55	1,230	217	3	7
NSF115	1,924	1,898	1,907	145	2	2	1,907	114	1	61	1,909	217	2	7
USA75	1,281	1,229	1,234	253	1	67	1,237	238	1	259	1,241	402	2	129
USA125	2,255	2,190	2,191	361	3	92	2,195	311	1	737	2,201	458	2	138
USA150	3,029	2,914	2,948	406	3	80	2,961	292	1	797	2,975	508	2	155
GER100	2,306	2,185	2,206	336	1	82	2,226	317	1	358	2,245	537	3	427
GER130	2,960	2,791	2,861	432	2	107	2,874	399	1	802	2,889	639	3	467
GER150	4,663	4,472	4,507	1,443	11	808	4,507	692	2	3,354	4,519	937	6	1,747
NTT42	1,038	1,031	1,038	28	2	6	1,038	11	1	19	1,038	31	2	5
NTT50	1,362	1,356	1,362	103	3	5	1,362	29	1	19	1,362	59	2	7
NTT150	5,553	5,431	5,431	311	1	75	5,491	273	1	1,672	5,502	867	21	183
ATT20	359	330	330	544	3	1,906	335	512	1	3,493	354	380	7	1,473
ATT113	2,918	2,877	2,884	419	9	2,806	2,889	286	2	4,688	2,902	397	6	893
BRAZIL ₄₈	1,370	1,297	1,303	157	1	1,235	1,300	142	1	1,709	1,317	313	2	563

In two out of three data sets, CG++ reaches 100% GoS. Using Strategy 3 for NTT₁₅₀, the number of pre-computed paths is 1,568. Table IV shows that this set of paths is rich enough to reduce the number of calls to PP_{LINK}, while it is not too big, which otherwise would increase the computing times of PP_{PATH}.

3) *Exact vs. Heuristic*: When comparing with the exact methods, the heuristic algorithms fulfil their purpose: they provide very good solutions with small computational times. Indeed, differences between the accuracies of the exact and heuristic algorithms is, in most cases, quite low. For CG^{H+} the difference between returned GoS and the corresponding ε -optimal GoS from CG+ varies between 0 lightpaths (0% in all USA instances) to 283 (6.3% in GER₁₃₀). The performance of the algorithm CG^{H++} is even better, with the difference

between its solution and that of CG++ lying in the range of 5 lightpaths (0.3% in NSF₁₁₅) to 91 (3.1% GER₁₃₀).

Heuristic CG^{H++} always obtains better heuristic solutions than CG^{H+}. Improvement can lead to up to 247 more granted lightpaths, see, e.g., data set GER₁₅₀, 4,219 granted lightpaths with CG^{H+} vs. 4,466 with CG^{H++}, while the lower bound is equal to 4,663. The resulting accuracy is $\varepsilon = 4\%$, which is excellent for this large data set with 50 nodes, 176 links, 5666 lightpath requests, and 150 wavelengths. Computational time of CG^{H++} is usually larger than CG^{H+}, due to the larger number of considered paths in the wavelength generator problems (i.e., the pricing problems).

4) *Path Selection*: It is of interest to notice that, while the number of shortest paths for a given node pair is usually very small, although it increases with the length of the shortest path,

TABLE VI
CHARACTERISTICS OF THE SELECTED PATHS IN THE ε -OPTIMAL SOLUTIONS OF CG++

Data instances	SP ¹		SP ²		SP ³		SP ⁴		SP ^{≥ 5}	
	#	%	#	%	#	%	#	%	#	%
NSF ₃₀	391	83.2	66	14.0	11	2.3	1	0.2	1	0.2
NSF ₇₅	1,116	87.9	136	10.7	18	1.4	0	0.0	0	0.0
NSF ₁₁₅	1,808	89.2	193	9.5	24	1.2	2	0.1	0	0.0
USA ₇₅	914	61.0	418	27.9	154	10.3	10	0.7	2	0.1
USA ₁₂₅	1,599	62.0	670	26.0	278	10.8	26	1.0	7	0.3
USA ₁₅₀	2,108	64.5	831	25.4	299	9.1	24	0.7	6	0.2
GER ₁₀₀	1,321	49.8	841	31.7	383	14.5	78	2.9	27	1.0
GER ₁₃₀	1,642	49.5	923	27.8	559	16.8	153	4.6	42	1.3
GER ₁₅₀	3,221	64.9	1,002	20.2	742	14.9	0	0.0	0	0.0
NTT ₄₂	1,304	81.2	173	10.8	86	5.4	25	1.6	17	1.1
NTT ₅₀	1,409	79.9	202	11.5	94	5.3	37	2.1	22	1.2
NTT ₁₅₀	4,633	77.8	900	15.1	298	5.0	84	1.4	37	0.6
ATT ₂₀	205	51.5	49	12.3	58	14.6	26	6.5	60	15.1
ATT ₁₁₃	1,195	40.2	1,264	42.5	500	16.8	7	0.2	5	0.2
BRAZIL ₄₈	739	46.7	640	40.5	201	12.7	1	0.1	0	0.0

the number of k -shortest paths in P² can increase sharply. For instance, the number of 2nd shortest paths can reach 52 and 70 for some node pairs in USA and GER networks, respectively.

In Table V, we compare the three strategies for selecting the paths in CG++ and this table is a summary of the extensive computational experiments we conducted. For Strategy 1, we found out that a good value was $k^{\text{SP}} = 15$, and in Strategy 2, a good constant value is $\rho = 0.5$. Both correspond to a very good compromise between the computational times and the maximum GoS value. Computational times in columns entitled CPU are in seconds and do not include the time needed for preprocessing.

Results of Table V show that CG++ using Strategy 1 mostly has the smallest computational times, while Strategy 3 results in improved GoS. In all data sets, Strategy 2 is at least as good as Strategy 1, and, in many cases, improves it. However, the computational time is increased drastically, due to a significantly larger number of considered paths, while not all of them are very useful. Strategy 3 improves the GoS of almost all the data sets, while reducing the computational time compared to Strategy 2. For example, employing Strategy 3 in solving NTT₁₅₀ grants 71 and 11 more requests than Strategy 1 and 2 respectively, while the computational times are still in a reasonable range. In conclusion, algorithm CG++ with Strategy 3 provides the best GoS within reasonable computational times.

Characteristics of the selected paths in CG++ are described in Table VI. We observe that, while shortest paths play an important role in the ε -optimal solution of CG++, 2nd and 3rd shortest paths also have a significant share in the final set of chosen paths. It is interesting to see that, for ATT₁₁₃, the share of the 2nd shortest paths is even more than the 1st shortest paths. This justifies the enlargement of the initial set of pre-computed paths to take into account the paths beyond shortest paths.

5) *Spectrum Utilization:* Lastly, we investigate the bandwidth spectrum usage for 3 instances, in the ε -optimal solutions of CG++. In Table VII, we report the average, min and max link spectrum usage for the ε -optimal solutions of Algorithm CG++, while in Figure 6, we display the spectrum link usage for 3 different instances.

For both GER₁₅₀ and the ATT₁₁₃ instances, the variance in the spectrum usage of links is quite high, while in USA₁₅₀,

TABLE VII
AVERAGE, MIN AND MAX LINK SPECTRUM (%)

Data instances	Min	Average	σ	Max
NSF ₃₀	50.0	84.4	13.0	100.0
NSF ₇₅	44.0	62.0	7.6	69.3
NSF ₁₁₅	50.4	71.7	7.7	80.0
USA ₇₅	29.3	71.8	16.4	96.0
USA ₁₂₅	32.8	63.8	14.3	84.8
USA ₁₅₀	36.7	63.2	14.7	86.7
GER ₁₀₀	0.0	53.5	28.6	97.0
GER ₁₃₀	0.0	51.7	28.0	93.8
GER ₁₅₀	0.0	55.3	29.2	90.0
NTT ₄₂	0.0	19.2	9.1	40.8
NTT ₅₀	0.0	31.7	14.2	58.0
NTT ₁₅₀	0.0	62.8	20.8	88.7
ATT ₂₀	0.0	41.0	31.0	100.0
ATT ₁₁₃	0.0	24.9	23.6	97.3
BRAZIL ₄₈	37.5	72.0	15.0	100.0

it is smaller. It can be explained by the characteristics of the topologies: USA is closer to a grid like topology in which nearly all links participate equally in the overall set of k -shortest paths, while GER and ATT topologies are more irregular. It may mean that if traffic patterns deviate further from rather homogeneously distributed traffic patterns (see the mean and the variance of the current traffic patterns in Table II), then, we might observe even larger variances in the spectrum usage of the links.

In Figure 7, we have indicated in red the links with the highest number of lightpath traversals, and in green the links with the small number. Blue links represent the most loaded links obtained by solving the max-flow formulation (30)–(35). Comparison between the sets of blue and red links shows that the max-flow model is successful in identifying the potentially bottleneck links. This leads to a smarter approach in selecting the paths used in the wavelength generator problem PP_{PATH} of the CG++ algorithm with Strategy 3. It also results in a more load balanced solution that has less highly loaded links. In NTT₁₅₀, link $\ell_{(6,4)}$ are the least loaded links after solving the problem with CG++ and have a 0% spectrum usage. Since the number of traffic requests in NTT₁₅₀ between nodes v_6 and v_4 is 0 in both directions, the result is justified. GER₁₅₀ has three edges (i.e., 6 links) with 0% spectrum usage: $\ell_{(28,29)}, \ell_{(46,71)}, \ell_{(49,18)}$, meaning that those links do not belong to any path of interest for provisioning the demand.

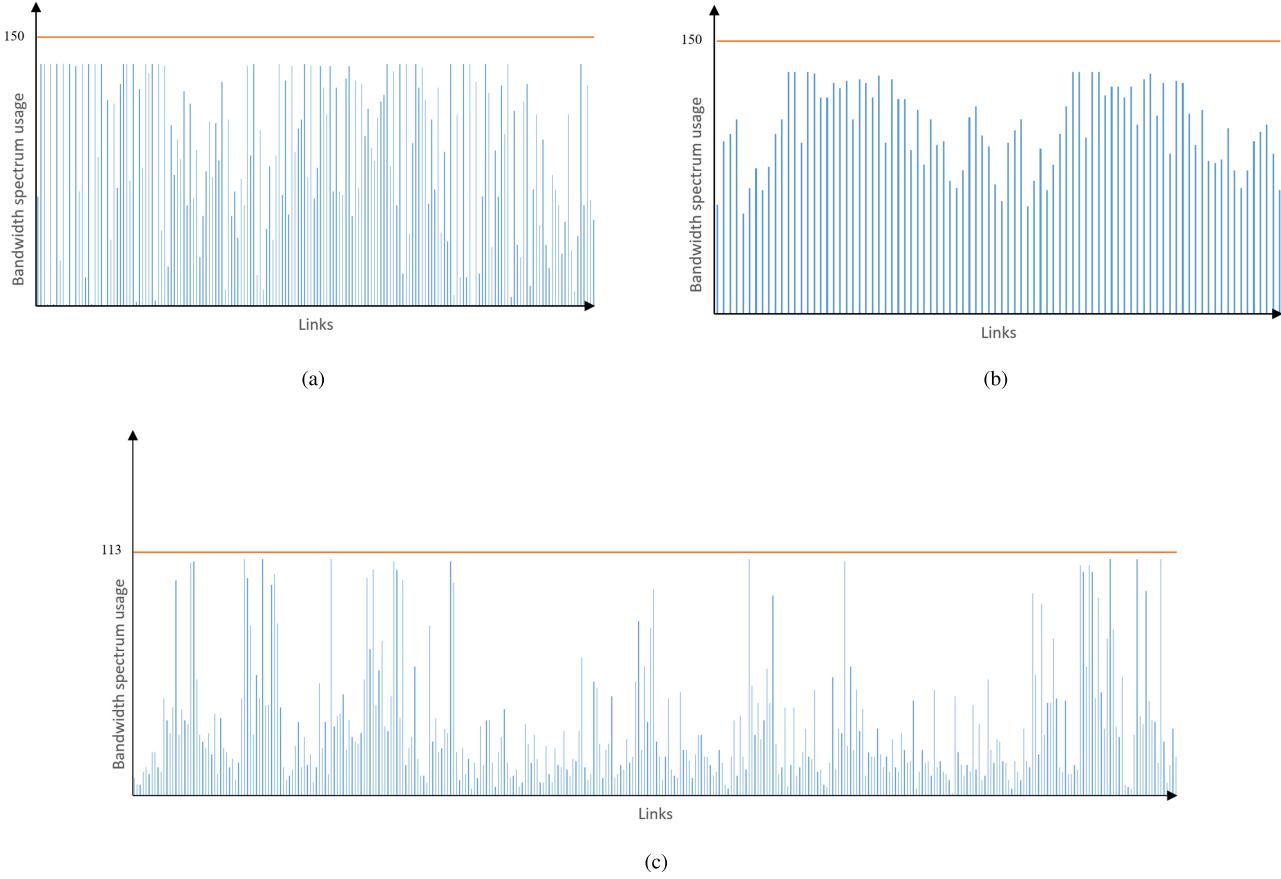


Fig. 6. Bandwidth Spectrum Usage with CG++. (a) GER₁₅₀. (b) USA₁₅₀. (c) ATT₁₁₃.

C. Dynamic Case: Algorithm Performances

We investigate dynamic traffic within the framework of incremental traffic, following the steady increase of, e.g., the video traffic that has been observed during the last 5 years [1]. The purpose of the experiments is to evaluate how much bandwidth is wasted if no lightpath rearrangements are performed when increasing the number of lightpaths.

In order to generate incremental traffic, the overall set of connection requests of every data set is divided into smaller sets, such that $\mathcal{SD}^{t-1} \subseteq \mathcal{SD}^t$ and for every pair $(v_s, v_d) \in \mathcal{SD}^t$, $D_{sd}^{t-1} \leq D_{sd}^t$, where t represents a time parameter. At each time period, δ traffic requests are selected randomly, from the original demand set \mathcal{SD} at time $t = 0$, and from the remaining demand set $(\mathcal{SD} \setminus \bigcup_{t': t' < t} \mathcal{SD}^{t'})$ of time $t > 0$. The requests are randomly selected, one unit at a time.

Results are summarized in Table VIII, with the use of different values for the δ step size. We report the maximum GoS for each strategy and dataset. The results for the static case are the ones obtained by using CG++ with Strategy 3.

We observe that, in every dataset, up to a turning point, the greater step size affects the most the set of available lightpaths and causes the GoS to drop. From that turning point on, the incoming sets of new traffic requests become big enough to be more similar to the static traffic. This results in a provisioning that is closer to the optimal granting in

TABLE VIII
GRADE OF SERVICE - NO LIGHTPATH REARRANGEMENT

Data instances	GoS				
	Static	$\delta = 10$	$\delta = 50$	$\delta = 200$	$\delta = 500$
NSF ₃₀	96.6	83.7	71.6	74.1	96.6
NSF ₇₅	89.6	71.3	62.3	56.6	63.0
NSF ₁₁₅	87.0	70.6	61.3	52.8	59.4
USA ₇₅	92.9	87.0	78.6	71.2	78.5
USA ₁₂₅	90.9	80.6	76.5	66.3	68.7
USA ₁₅₀	84.8	73.7	65.8	57.6	59.3
GER ₁₀₀	94.9	82.7	75.7	68.9	76.3
GER ₁₃₀	95.0	81.8	71.6	67.9	71.7
GER ₁₅₀	79.8	60.7	55.0	44.7	48.5
NTT ₄₂	100.0	98.1	96.3	93.3	96.6
NTT ₅₀	100.0	90.1	89.4	85.2	89.7
NTT ₁₅₀	96.8	76.4	73.9	59.9	63.5
ATT ₂₀	98.6	81.3	75.5	86.1	98.6
ATT ₁₁₃	99.5	84.5	80.0	85.5	95.5
BRAZIL ₄₈	96.1	88.4	83.9	82.0	88.2

the static case, and ultimately a higher GoS. For instance, in GER₁₃₀, GoS declines as the step size grows from 50 to 200. After this point, with increasing the step size, GoS starts to improve. In practice, network upgrades are commonly carried out in scheduled intervals (i.e., every three to six months) [30], using traffic forecasts [31], leading to some small δ increment in percent with respect to the existing traffic, rather than large ones. Consequently, it means potentially significant bandwidth spectrum misuse if no lightpath rearrangement are conducted.

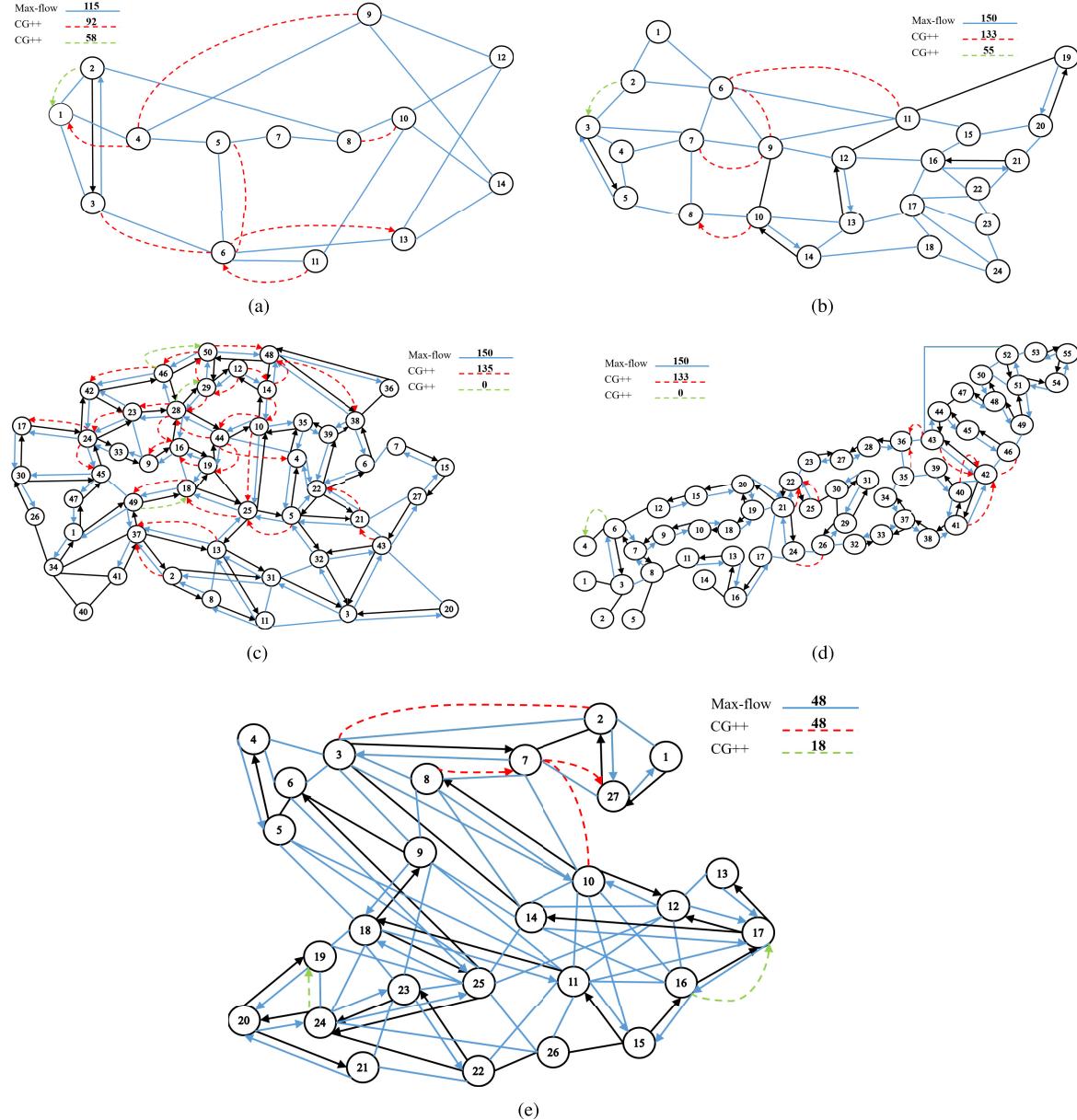


Fig. 7. Compare the highly loaded links in the solutions of model (30)–(36) vs. CG++ model. (a) NSF₁₁₅. (b) USA₁₅₀. (c) GER₁₅₀. (d) NTT₁₅₀. (e) BRAZIL₄₈.

While technology did not allow such lightpath rearrangements in the past, this is now possible in DWDM networks that are equipped with directionless and colorless ROADM [32].

VII. CONCLUSION

We obtained very efficient algorithms for solving large scale RWA problems, which allow the exact solutions of data sets with up to 90 nodes and 150 wavelengths. In addition, we also provide two heuristics that provide very good solutions, in very short computing times, while their accuracy can be assessed. It appears that the concept of limiting the search of wavelength configurations using shortest paths, or a limited number of k -shortest paths is very promising in terms of enhancing the performance of the ε -optimal or heuristic algorithms.

Future work should account for multi-rate lightpaths, and the minimum number of required regenerators.

APPENDIX EXISTING ILP MAX-RWA MODELS

We provide here a review on the ILP max-RWA formulations of the literature. There are three general classes of ILP formulations for max-RWA problem. The first one corresponds to *link formulations* and uses variables indexed with links. The second class of formulations relies on paths, resulting in *path formulations*. The third one uses configurations and decomposition schemes and is reviewed in the core part of the paper. General definitions and notations are the same as the ones in Section II.

A. Link Formulations

There are mainly three types of link formulations, characterized by the way the connection requests are considered: individually (model RWA _{k}), grouped with respect to

their source/destination nodes (model RWA_{sd}), grouped with respect to their source nodes (model RWA_s).

1) *RWA_k Request Indexed Model:* Many authors used that formulation, it is the most used exact ILP one for the max RWA problem, see, e.g., [33]. Every connection request $k \in K$ is characterized by its source v_s^k and its destination v_d^k . Variables are defined as follows:

$x_k^\lambda = 1$ if request k is granted by wavelength λ , 0 otherwise.
 $x_{k\ell}^\lambda = 1$ if request k is granted by wavelength λ on link ℓ , 0 otherwise.

The mathematical formulation is as follows:

$$\max z_{\text{RWA}_k}(x) = \sum_{k \in K} \sum_{\lambda \in \Lambda} x_k^\lambda \quad (38)$$

subject to:

$$\sum_{\ell \in \omega^+(v)} x_{k\ell}^\lambda = \sum_{\ell \in \omega^-(v)} x_{k\ell}^\lambda \quad k \in K, \lambda \in \Lambda, \quad v \in V \setminus \{v_s^k, v_d^k\} \quad (39)$$

$$\sum_{\ell \in \omega^+(v_s^k)} x_{k\ell}^\lambda = \sum_{\ell \in \omega^-(v_d^k)} x_{k\ell}^\lambda = x_k^\lambda \quad k \in K, \lambda \in \Lambda \quad (40)$$

$$\sum_{\ell \in \omega^-(v_s^k)} x_{k\ell}^\lambda = \sum_{\ell \in \omega^+(v_d^k)} x_{k\ell}^\lambda = 0 \quad k \in K, \lambda \in \Lambda \quad (41)$$

$$\sum_{k \in K} x_{k\ell}^\lambda \leq 1 \quad \ell \in L, \lambda \in \Lambda \quad (42)$$

$$\sum_{\lambda \in \Lambda} x_{k\ell}^\lambda \leq 1 \quad k \in K \quad (43)$$

$$x_{k\ell}^\lambda \leq x_k^\lambda \quad k \in K, \ell \in L, \lambda \in \Lambda \quad (44)$$

$$x_k^\lambda, x_{k\ell}^\lambda \in \{0, 1\} \quad k \in K, \ell \in L, \lambda \in \Lambda. \quad (45)$$

The objective function (38) maximizes the number of granted requests. Constraints (39) and (40) establish the routes while respecting the wavelength continuity constraints. Loops at the source and destination nodes are eliminated by Constraints (41). Constraints (42) prevent the wavelength conflict: on each link, each wavelength belongs to at most one lightpath. Constraints (43) make sure at most one wavelength is used for every request. Consistency between variables is guaranteed thanks to Constraints (44). Constraints (45) define the domain of the variables.

2) *RWA_s Source Indexed Model:* The third model is a source indexed formulation in which requests are grouped based on their source nodes. This type of modelling was studied in [34]–[36] and improved in [10]. Let K_s denote the set of connection requests starting from node v_s , while V^{D_s} contains their destination nodes. Define by $D_s = \sum_{v \in V^{D_s}} D_{sv}$, the aggregated traffic request from node v_s . Consider the following variables:

$y_{s\ell}^\lambda = 1$ if a connection request from v_s is granted by wavelength λ on link ℓ , 0 otherwise.

The mathematical formulation is as follows:

$$\max z_{\text{RWA}_s}(y) = \sum_{(v_s, v_d) \in \mathcal{SD}} \left(\sum_{\lambda \in \Lambda} \sum_{\ell \in \omega^-(v_d)} y_{s\ell}^\lambda - \sum_{\lambda \in \Lambda} \sum_{\ell \in \omega^+(v_d)} y_{s\ell}^\lambda \right) \quad (46)$$

subject to:

$$\sum_{\ell \in \omega^+(v)} y_{s\ell}^\lambda - \sum_{\ell \in \omega^-(v)} y_{s\ell}^\lambda = 0 \quad \lambda \in \Lambda, v_s \in V : D_s > 0, \quad v \in V \setminus (V^{D_s} \cup \{v_s\}) \quad (47)$$

$$\sum_{\lambda \in \Lambda} \sum_{\ell \in \omega^-(v_d)} y_{s\ell}^\lambda - \sum_{\lambda \in \Lambda} \sum_{\ell \in \omega^+(v_d)} y_{s\ell}^\lambda \leq D_{sd} \quad v_s \in V : D_s > 0, \quad v_d \in V^{D_s} \quad (48)$$

$$\sum_{\ell \in \omega^+(v_d)} y_{s\ell}^\lambda - \sum_{\ell \in \omega^-(v_d)} y_{s\ell}^\lambda \leq 0 \quad \lambda \in \Lambda, v_s \in V : D_s > 0, \quad v_d \in V^{D_s} \quad (49)$$

$$\sum_{v_s \in V : D_s > 0} y_{s\ell}^\lambda \leq 1 \quad \ell \in L, \lambda \in \Lambda \quad (50)$$

$$y_{s\ell}^\lambda \in \{0, 1\} \quad \ell \in L, \lambda \in \Lambda, \quad v_s \in V : D_s > 0. \quad (51)$$

Again, the objective (46) maximizes the grade of service. Constraints (47)–(49) are flow conservation constraints and take care of wavelength continuity requirements. Constraints (50) guarantee that there is at most one connection request per wavelength on each link. Constraints (51) define the domains of the variables.

3) *RWA_{sd} Node Pair Indexed Model:* Link model RWA_{sd} aggregates the requests according to their source and destination nodes. It uses the following variables:

$y_{sdl}^\lambda = 1$ if a connection request from v_s to v_d is granted by wavelength λ on link ℓ , 0 otherwise.

The mathematical formulation is as follows:

$$\max z_{\text{RWA}_s}(y) = \sum_{\lambda \in \Lambda} \sum_{(v_s, v_d) \in \mathcal{SD}} \sum_{\ell \in \omega^+(v_s)} y_{sdl}^\lambda \quad (52)$$

subject to:

$$\sum_{\ell \in \omega^+(v)} y_{sdl}^\lambda = \sum_{\ell \in \omega^-(v)} y_{sdl}^\lambda \quad \lambda \in \Lambda, (v_s, v_d) \in \mathcal{SD}, \quad v \in V \setminus \{v_s, v_d\} \quad (53)$$

$$\sum_{\lambda \in \Lambda} \sum_{\ell \in \omega^+(v_s)} y_{sdl}^\lambda = \sum_{\lambda \in \Lambda} \sum_{\ell \in \omega^-(v_d)} y_{sdl}^\lambda \leq D_{sd} \quad (v_s, v_d) \in \mathcal{SD} \quad (54)$$

$$\sum_{\lambda \in \Lambda} \sum_{\ell \in \omega^-(v_s)} y_{sdl}^\lambda = \sum_{\lambda \in \Lambda} \sum_{\ell \in \omega^+(v_d)} y_{sdl}^\lambda = 0 \quad (v_s, v_d) \in \mathcal{SD} \quad (55)$$

$$\sum_{(v_s, v_d) \in \mathcal{SD}} y_{sdl}^\lambda \leq 1 \quad \ell \in L, \lambda \in \Lambda \quad (56)$$

$$y_{sdl}^\lambda \in \{0, 1\} \quad \ell \in L, \lambda \in \Lambda, \quad (v_s, v_d) \in \mathcal{SD}. \quad (57)$$

Again, the objective (52) maximizes the grade of service. Constraints (53) and (54) are the continuity constraints. Constraints (55) prevent the loops around source and destination nodes. Wavelength conflict is avoided thanks to Constraints (56). Constraints (57) define the domain of the variables.

B. Path Formulation

This section discusses the ILP model that uses path variables. Such a model was studied in e.g., [37], [38]. Note that it is an exact model only if all paths are considered, and therefore not scalable as soon as we consider topologies with more than 10 nodes.

Let $\mathcal{P} = \bigcup_{sd \in \mathcal{SD}} P_{sd}$ be the overall set of simple (i.e., loopless) paths indexed by p , and δ_ℓ^p a parameter that is equal to 1 if link ℓ is in path p , 0 otherwise. Consider the following variables:

$x_p^\lambda = 1$ if a lightpath is established using path p and wavelength λ , 0 otherwise.

The mathematical formulation is as follows:

$$\max z_{\text{PATH}}(x) = \sum_{\lambda \in \Lambda} \sum_{p \in \mathcal{P}} x_p^\lambda \quad (58)$$

subject to:

$$\sum_{p \in \mathcal{P}} \delta_\ell^p x_p^\lambda \leq 1 \quad \ell \in L, \lambda \in \Lambda \quad (59)$$

$$\sum_{\lambda \in \Lambda} \sum_{p \in P_{sd}} x_p^\lambda \leq D_{sd} \quad (v_s, v_d) \in \mathcal{SD} \quad (60)$$

$$x_p^\lambda \in \{0, 1\} \quad p \in \mathcal{P}, \lambda \in \Lambda. \quad (61)$$

Again, the objective (58) maximizes the grade of service. Constraints (59) make sure that only one lightpath is going through every pair of wavelengths and links. Constraints (60) force the model to grant at most D_{sd} lightpaths to every pair (v_s, v_d) . Constraints (61) define the domains of the variables.

REFERENCES

- [1] *Cisco Visual Networking Index: Forecast and Methodology, 2014–2019*, CISCO, San Jose, CA, USA, May 2015.
- [2] R. Durairajan, P. Barford, J. Sommers, and W. Willinger, “InterTubes: A study of the US long-haul fiber-optic infrastructure,” in *Proc. Conf. ACM Special Interest Group Data Commun. (SIGCOMM)*, London, U.K., Aug. 2015, pp. 1–14.
- [3] N. Nagatsu, S. Okamoto, and K. Sato, “Large scale photonic transport network design based on optical paths,” in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov. 1996, pp. 321–327.
- [4] T. F. Noronha, M. G. C. Resende, and C. C. Ribeiro, “A biased random-key genetic algorithm for routing and wavelength assignment,” *J. Global Optim.*, vol. 50, no. 3, pp. 503–518, Jul. 2006.
- [5] A. Martins, C. Duhamel, P. Mahey, R. Saldanha, and M. C. de Souza, “Variable neighborhood descent with iterated local search for routing and wavelength assignment,” *Comput. Oper. Res.*, vol. 39, no. 9, pp. 2133–2141, Sep. 2012.
- [6] R. Koganti and D. Sidhu, “Analysis of routing and wavelength assignment in large WDM networks,” *Procedia Comput. Sci.*, vol. 34, pp. 71–78, 2014.
- [7] C. Duhamel, P. Mahey, A. X. Martins, R. R. Saldanha, and M. de Souza, “Model-hierarchical column generation and heuristic for the routing and wavelength assignment problem,” *4OR*, vol. 14, no. 2, pp. 201–220, Jun. 2016.
- [8] T. Lee, K. Lee, and S. Park, “Optimal routing and wavelength assignment in WDM ring networks,” *IEEE J. Sel. Areas Commun.*, vol. 18, no. 10, pp. 2146–2154, Oct. 2000.
- [9] B. Jaumard, C. Meyer, and B. Thiongane, “On column generation formulations for the RWA problem,” *Discrete Appl. Math.*, vol. 157, no. 6, pp. 1291–1308, Mar. 2009.
- [10] B. Jaumard, C. Meyer, and B. Thiongane, “Comparison of ILP formulations for the RWA problem,” *Opt. Switching Netw.*, vol. 4, nos. 3–4, pp. 157–172, Nov. 2007.
- [11] B. Jaumard, C. Meyer, and B. Thiongane, “ILP formulations for the RWA problem for symmetrical systems,” in *Handbook of Optimization in Telecommunications*, P. M. Pardalos and M. G. C. Resende, Eds. Norwell, MA, USA: Kluwer, 2006, ch. 23, pp. 637–678.
- [12] K. Miliotis, G. I. Papadimitriou, and A. S. Pomportsis, “Design alternatives for wavelength routing networks,” *Opt. Laser Technol.*, vol. 35, no. 2, pp. 137–154, Mar. 2003.
- [13] H. Zang, J. P. Jue, and B. Mukherjee, “A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks,” *Opt. Netw. Mag.*, vol. 1, pp. 47–60, Jan. 2000.
- [14] V. Chvatal, *Linear Programming*. San Francisco, CA, USA: Freeman, 1983.
- [15] T. F. Noronha and C. C. Ribeiro, “Routing and wavelength assignment by partition colouring,” *Eur. J. Oper. Res.*, vol. 171, no. 3, pp. 797–810, Jun. 2006.
- [16] D. Banerjee and B. Mukherjee, “A practical approach for routing and wavelength assignment in large wavelength-routed optical networks,” *IEEE J. Sel. Areas Commun.*, vol. 14, no. 5, pp. 903–908, Jun. 1996.
- [17] B. Jaumard, C. Meyer, and X. Yu, “When is wavelength conversion contributing to reducing the blocking rate?” in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 4, Nov./Dec. 2005, pp. 2078–2083.
- [18] B. Jaumard, C. Meyer, and X. Yu, “How much wavelength conversion allows a reduction in the blocking rate?” *J. Opt. Netw.*, vol. 5, no. 12, pp. 881–900, Dec. 2006.
- [19] L. Velasco *et al.*, “On-demand incremental capacity planning in optical transport networks,” *J. Opt. Commun. Netw.*, vol. 8, no. 1, pp. 11–22, Jan. 2016.
- [20] Alcatel-Lucent Bell Labs. *Metro Network Traffic Growth: An Architecture Impact Study*, accessed on Aug. 2016. [Online]. Available: <http://www.tmcnet.com/tmc/whitepapers/documents/whitepapers/2013/9378-bell-labs-metro-network-traffic-growth-an-architecture.pdf>
- [21] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. Vance, “Branch-and-price: Column generation for solving huge integer programs,” *Oper. Res.*, vol. 46, no. 3, pp. 316–329, Mar. 1998.
- [22] J. Y. Yen, “Finding the K shortest loopless paths in a network,” *Manage. Sci.*, vol. 17, no. 11, pp. 712–716, Jul. 1971.
- [23] D. Eppstein, “Finding the k shortest paths,” *SIAM J. Comput.*, vol. 28, no. 2, pp. 652–673, Apr. 1998.
- [24] P. Pavon-Marino and J.-L. Izquierdo-Zaragoza, “Net2plan: An open source network planning tool for bridging the gap between academia and industry,” *IEEE Netw.*, vol. 29, no. 5, pp. 90–96, Sep./Oct. 2015. [Online]. Available: <http://www.net2plan.com/>
- [25] M. Daryalal, “Efficient spectrum utilization in large-scale RWA and RSA problems,” M.S. thesis, Dept. Comput. Sci. Softw. Eng., Concordia Univ., Montreal, QC, Canada, 2016.
- [26] *Cplex IBM ILOG CPLEX 12.6 Optimization Studio*, IBM, New York, NY, USA, 2014.
- [27] S. Orlowski, M. Piöro, A. Tomaszewski, and R. Wessäly, “SNDlib 1.0—Survivable network design library,” in *Proc. 3rd Int. Netw. Optim. Conf. (INOC)*, Spa, Belgium, Apr. 2007, pp. 276–286.
- [28] M. Batayneh, D. A. Schupke, M. Hoffmann, A. Kirstaedter, and B. Mukherjee, “On routing and transmission-range determination of multi-bit-rate signals over mixed-line-rate WDM optical networks for carrier Ethernet,” *IEEE/ACM Trans. Netw.*, vol. 19, no. 5, pp. 1304–1316, Oct. 2011.
- [29] M. Vega-Rodriguez and A. Rubio-Largo. *NTT Network*, accessed on Jul. 2016, [Online]. Available: http://mstar.unex.es/mstar_documentos/RWA/RWA-Instances.html
- [30] R. Aparicio-Pardo, P. Pavon-Marino, and B. Mukherjee, “Robust upgrade in optical networks under traffic uncertainty,” in *Proc. Conf. Opt. Netw. Design Modeling (ONDM)*, Colchester, U.K., Apr. 2012, pp. 1–6.
- [31] S. Yang and F. A. Kuipers, “Traffic uncertainty models in network planning,” *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 172–177, Feb. 2014.
- [32] Y. Li, L. Gao, G. Shen, and L. Peng, “Impact of ROADM colorless, directionless, and contentionless (CDC) features on optical network performance [Invited],” *J. Opt. Commun. Netw.*, vol. 4, no. 11, pp. B58–B67, Nov. 2012.
- [33] R. Krishnaswamy, “Algorithms for routing, wavelength assignment and topology design in optical networks,” Ph.D. dissertation, Dept. Elect. Commun. Eng., Indian Inst. Sci., Bengaluru, India, 1998.
- [34] D. Coudert and H. Rivano, “Lightpath assignment for multifibers WDM networks with wavelength translators,” in *Proc. IEEE Globecom*, Nov. 2002, pp. 2686–2690.
- [35] M. Tomatore, G. Maier, and A. Pattavina, “WDM network optimization by ILP based on source formulation,” in *Proc. IEEE Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, vol. 3, Jun. 2002, pp. 1813–1821.

- [36] R. M. Krishnaswamy and K. N. Sivarajan, "Design of logical topologies: A linear formulation for wavelength-routed optical networks with no wavelength changers," *IEEE/ACM Trans. Netw.*, vol. 9, no. 2, pp. 186–198, Apr. 2001.
- [37] K. Lee, K. C. Kang, T. Lee, and S. Park, "An optimization approach to routing and wavelength assignment in WDM all-optical mesh networks without wavelength conversion," *ETRI J.*, vol. 24, no. 2, pp. 131–141, Apr. 2002.
- [38] M. E. M. Saad and Z.-Q. Luo, "A Lagrangean decomposition approach for the routing and wavelength assignment in multifiber WDM networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov. 2002, pp. 2818–2822.

Maryam Daryalal received the M.Sc. degree in industrial engineering from the Amirkabir University of Technology, Tehran, Iran, in 2013, and the M.Comp.Sc. degree in computer science from Concordia University, Montreal, QC, Canada, in 2016. Since 2017, she is with Morgan Stanley, Montreal, as a Technology Analyst of Wealth Management.



Brigitte Jaumard (A'97–M'04–SM'07) holds a Concordia University Research Chair, Tier 1, on the Optimization of Communication Networks with the Computer Science and Software Engineering Department, Concordia University. Recent studies include the design of the most efficient algorithms for the design and dimensioning of logical survivable topologies against single or multiple failures (e.g., survivable VPN topology over a service provider network). Other recent studies deal with dimensioning, provisioning, and scheduling algorithms in optical grids or clouds, in broadband wireless networks, and in passive optical networks. In artificial intelligence, contributions include the development of efficient optimization algorithms for probabilistic logic (reasoning under uncertainty) and for automated mechanical design in social networks (design of trust estimator tools). In transportation, her recent contributions include new algorithms for freight train scheduling and locomotive assignment subject to energy minimization. She has published over 300 papers in international journals in operations research and in telecommunications. Her current research interests include mathematical modeling and algorithm design for large-scale optimization problems arising in communication networks, transportation and logistics networks, and artificial intelligence.