A novel integer linear programming model for routing and spectrum assignment in optical networks

Youssouf Hadhbi, Hervé Kerivin, Annegret Wagler
Université Clermont Auvergne,
LIMOS (UMR 6158 CNRS),
Clermont-Ferrand, France
Email: {youssouf.hadhbi,herve.kerivin,annegret.wagler}@uca.fr

Abstract—The routing and spectrum assignment problem is an NP-hard problem that receives increasing attention during the last years. Existing integer linear programming models for the problem are either very complex and suffer from tractability issues or are simplified and incomplete so that they can optimize only some objective functions. The majority of models uses edge-path formulations where variables are associated with all possible routing paths so that the number of variables grows exponentially with the size of the instance. An alternative is to use edge-node formulations that allow to devise compact models where the number of variables grows only polynomially with the size of the instance. However, all known edge-node formulations are incomplete as their feasible region is a superset of all feasible solutions of the problem and can, thus, handle only some objective functions.

Our contribution is to provide the first complete edge-node formulation for the routing and spectrum assignment problem which leads to a tractable integer linear programming model. Indeed, computational results show that our complete model is competitive with incomplete models as we can solve instances of the RSA problem larger than instances known in the literature to optimality within reasonable time and w.r.t. several objective functions. We further devise some directions of future research.

I. INTRODUCTION

Today’s communication networks are optical networks where light is used as communication medium between sender and receiver nodes. For over two decades, the Wavelength-Division Multiplexing (WDM) has been the most popular technology used in fiber-optic communication. WDM combines multiple wavelengths to simultaneously transport signals over a single optical fiber, but must select the wavelengths from a rather coarse fixed grid of frequencies specified by the United Nations agency ITU (International Telecommunication Union) and leads to inefficient use of spectral resources and bans allocating more than a single wavelength to a traffic demand.

In response to the sustained growth of data traffic volumes in communication networks, a new generation of optical networks, called flexgrid Elastic Optical Networks (EONs), has been introduced in the last few years to enhance the spectrum efficiency and enlarge the network capacity [7].

In EONs, the frequency spectrum of an optical fiber is divided into many narrow frequency slots of fixed spectrum width. Any sequence of consecutive slots can form a channel that can be switched in the network nodes to create a lightpath (i.e., an optical connection represented by a route and a channel). EONs enable capacity gain by allocating minimum required bandwidth thanks to a finer spectrum granularity than in the traditional WDM networks.

However, the spectrum assignment in EONs leads to the Routing and Spectrum Assignment (RSA) problem that is much harder to handle in practice than its counterpart using Wavelength-Division Multiplexing. In fact, the RSA problem consists of two parts: the routing (to select for each traffic demand a path through the communication network) and the spectrum assignment (to assign for each demand an interval of consecutive frequency slots within the optical spectrum such that the intervals of lightpaths using a same edge in the network are disjoint), see e.g. [15] and Section II for details. Thereby, the following constraints need to be respected when dealing with the RSA problem:

1) spectrum continuity: the frequency slots allocated to a demand remain the same on all the links of a route;
2) spectrum contiguity: the frequency slots allocated to a demand must be contiguous;
3) non-overlapping spectrum: a frequency slot can be allocated to at most one demand.

The RSA problem is a generalization of the well-studied Routing and Wavelength Assignment (RWA) problem that is associated with a fixed grid of frequencies [3].

The former problem has started to receive a lot of attention over the last few years. It has been shown to be NP-hard [2], [18]. In fact, if for each demand the route is already known, the RSA problem reduces to the so-called Spectrum Assignment (SA) problem and only consists of determining the demands’ channels. The SA problem has been shown to be NP-hard on paths [14] which makes the SA problem (and thus also the RSA problem) much harder than the RWA problem which is well-known to be polynomially solvable on paths, see e.g. [3].

To solve the RSA problem, various approaches have been studied in the literature, based on different Integer Linear Programming (ILP) models. Hereby, detailed models aiming at precisely describing all technological aspects of EONs and being able to handle various criteria for optimization typically suffer from tractability issues resulting from their...
The majority of the existing models uses an edge-path formulation where for each demand, variables are associated either with all possible routing paths or with all possible lightpaths for this demand. One characteristic of this formulation is, therefore, an exponential number of variables issued from the total number of all feasible paths between origin-destination pairs in the network, which grows exponentially with the size of the network.

To bypass the exponential number of variables, edge-path formulations with a precomputed subset of all possible paths per demand have been studied e.g. in [8], [9], [16], [19], see [19] for an overview. However, such formulations cannot guarantee optimality of the solutions in general (as only a precomputed subset of paths is considered and, thus, a restricted problem solved). In order to be able to find optimal solutions of the RSA problem w.r.t. any objective function with the help of an edge-path formulation, all possible paths have to be taken into account. As the explicit models are far too big for computation, it is in order to apply column-generation methods. However, computational results from e.g. [10], [11], [13] show that the size of the instances that can be solved that way is rather limited.

An alternative to edge-path formulations is to use edge-node formulations that lead to less intuitive models for the routing, but have the advantage that the number of variables grows only polynomially with the size of the instance. Despite this advantage, edge-node formulations are not yet well-studied. Only few authors made use of this type of model, as Cai et al. [1], Velasco et al. [16], Zotkiewicz et al. [19], and Jia et al. who used in [6] an edge-node formulation to treat a more general problem.

All three models from [1], [16], [19] are compact models as both the number of variables and constraints is polynomial in terms of the size of the instance. However, all three models are incomplete as their feasible region is a superset of all feasible solutions of the RSA problem and can, thus, handle only some objective functions (see Section IV for details).

Our contribution is to provide the first complete edge-node formulation for the RSA problem that precisely encodes the set of all feasible solutions and can, therefore, be used to optimize any chosen objective function. For that, we propose an appropriate combination of variables and constraints (partly using new variables and constraints), see Section III for details. Our model uses, as in [1], [16], [19], a polynomial number of variables, but an exponential number of constraints to ensure the exact encoding of feasible solutions. As we are able to separate the exponentially-sized families of constraints in polynomial time, our model is computationally tractable and, therefore, competitive with the compact but incomplete models from [1], [16], [19].

While Zotkiewicz et al. [19] do not give computational results, Velasco et al. [16] tested their formulation on a network topology of Spain with 35 edges (64 slots per edge) and 21 nodes with a very small number of 12 demands and requested numbers of slots in \{1, 2, 4\}. The results show that Cplex version 12.1 could optimally solve the problem after 6 hours by minimizing the number of edges activated for the routing (which can be looked at as a network design problem).

Cai et al. [1] tested their formulation on two small network topologies, one with 6 nodes and 9 links and the other with 10 nodes and 22 links, one demand between each pair of nodes in the network and requested numbers of slots in \{1, \ldots, 3\}, \ldots, \{1, \ldots, 9\}. The results show that Gurobi 5.0 could optimally solve the problem after 1 hour by minimizing the max-slot position for the 6 nodes and 9 links topology (but did not report on time limits to solve the instances on the other network).

Our model allows us to solve instances of the RSA problem larger than the instances in [1], [16] to optimality within reasonable time w.r.t. several objective functions (see Section V for details).

The paper is organized as follows. In Section II, we describe in detail the input and the desired output of the RSA problem together with the studied objective functions. In Section III, we present our new edge-node formulation and compare it in Section IV with existing models from the literature [1], [16], [19]. In Section V, we report on computational results achieved with the help of our formulation. We close with some concluding remarks and future research.

II. THE RSA PROBLEM

In this section, we formally define the RSA problem by describing in detail the input and the desired output of the RSA problem together with the studied objective functions.

As input of the RSA problem, we are given
- an optical spectrum \( S = \{1, \ldots, s\} \) of available frequency slots;
- an optical network, represented as an undirected, loopless, connected graph \( G = (V, E) \) that may have parallel edges (if parallel optical fibers are installed between two nodes), and for each edge \( e \in E \) its length \( \ell_e \in \mathbb{R}_+ \) (in kms),
- a multiset \( K \) of demands where each demand \( k \in K \) is specified by
  - an origin node \( o_k \in V \) and a destination node \( d_k \in V \setminus \{o_k\} \),
  - a requested number \( w_k \in \mathbb{N}_+ \) of slots, and
  - a transmission reach \( \bar{\ell}_k \in \mathbb{R}_+ \) (in kms).

The task is to determine for each demand \( k \in K \) a lightpath composed of an \((o_k,d_k)\)-path \( P_k \) in \( G \) respecting the transmission reach \( \bar{\ell}_k \) and a subset \( S_k \subset S \) of \( w_k \) consecutive frequency slots that is available on all edges of \( P_k \) and disjoint from the subsets \( S_{k'} \) of all other demands \( k' \) routed along an edge of \( P_k \), thereby minimizing some objective function.

Hence, the desired output of the RSA problem is, for each demand \( k \in K \), a lightpath composed of
- an \((o_k,d_k)\)-path \( P_k \) in \( G \) with \( \sum_{e \in E(P_k)} l_e \leq \bar{\ell}_k \),
\textbf{III. A NOVEL EDGE-NODE FORMULATION}\textbf{\textit{}}

In this section we introduce our novel edge-node ILP model for the RSA problem in the general variant where demands may be rejected.

\textit{a) Variables:} For the routing, demand-edge variables

\[
z_e^k = \begin{cases} 
1 & \text{if demand } k \text{ is routed through edge } e, \\
0 & \text{otherwise,}
\end{cases}
\]

are used for all \( k \in K \) and all \( e \in E \) as in [17], [19].

For the spectrum assignment, several different variables are necessary. As in [2], [16], demand-slot variables

\[
z_s^k = \begin{cases} 
1 & \text{if slot } s \text{ is the last slot allocated for demand } k, \\
0 & \text{otherwise,}
\end{cases}
\]

are used which indicate that \( s \) is the last of the \( w_k \) consecutive slots allocated for the demand \( k \in K \), with \( s \in S \). The consecutive slots \( s' \in \{s-w_k+1, \ldots, s\} \) shall form the channel assigned to this demand \( k \) whenever \( z_s^k = 1 \).

We newly propose demand-edge-slot variables

\[
t_{e,s}^k = \begin{cases} 
1 & \text{if slot } s \text{ is assigned to demand } k \text{ on edge } e, \\
0 & \text{otherwise},
\end{cases}
\]

for all demands \( k \in K \), all edges \( e \in E \) and all slots \( s \in S \).

When we optimize objective functions involving max-used slot positions, we newly propose edge-max-slot-position variables \( p_e \in Z^+ \) for all edges \( e \in E \) (which indicate the position
of the last slot allocated on the edge \( e \in E \), as well as a max-slot-position variable \( p \in \mathbb{Z}^+ \) (which represents the position of the highest slot used over all the edges \( e \in E \) as in [2]).

When we optimize the number of edges used for the routing, we newly propose edge-activation variables

\[
a_e = \begin{cases} 1 & \text{if some demand } k \text{ is routed through edge } e, \\ 0 & \text{otherwise}, \end{cases}
\]

for all edges \( e \in E \).

b) Constraints: To formulate the constraints, we employ the following notations. For any non-empty subset \( X \subset V \), let \( \delta(X) \) denote the set of edges having one endnode in \( X \) and the other endnode in \( V \setminus X \). The pair \( (X, V \setminus X) \) is called a cut of \( G \), the edges in \( \delta(X) \) are said to cross this cut. In the special case \( X = \{v\} \), we write \( \delta(v) \) instead of \( \delta(\{v\}) \).

For the routing, we use demand-edge variables \( x_{ek}^k \) and have to ensure by appropriate constraints that the subset

\[
E(k) = \{ e \in E : x_{ek}^k = 1 \}
\]

of edges selected for the routing of demand \( k \) indeed forms an \((o_k, d_k)\)-path \( P_k \) in \( G \), for each demand \( k \in K \). For that, we use the following constraints. The origin constraints

\[
\sum_{e \in \delta(o_k)} x_{ek}^k \leq 1, \text{ for all } k \in K
\]

ensure that at most one path \( P_k \) can leave the origin \( o_k \) as at most one of the edges \( e \in \delta(o_k) \) incident to \( o_k \) can be selected for \( E(k) \). Similarly, destination constraints

\[
\sum_{e \in \delta(d_k)} x_{ek}^k - \sum_{e \in \delta(o_k)} x_{ek}^k = 0, \text{ for all } k \in K
\]

force that the path \( P_k \) enters its destination \( d_k \), provided that there is a path \( P_k \) leaving \( o_k \). (Note that if no path is selected for demand \( k \), then \( \sum_{e \in \delta(o_k)} x_{ek}^k = 0 \) holds and ensures that no edge from \( \delta(d_k) \) can be selected either for \( E(k) \).) Origin and destination constraints are used in [1], [16], [19] in a slightly different manner.

In addition, we newly propose path-continuity constraints

\[
\sum_{e \in \delta(X)} x_{ek}^k - \sum_{e \in \delta(o_k)} x_{ek}^k \geq 0, \forall k \in K, \forall X, o_k \in X, d_k \in V \setminus X.
\]

These constraints are important whenever a path \( P_k \) is selected for demand \( k \) (and, thus, \( \sum_{e \in \delta(o_k)} x_{ek}^k = 1 \) holds): they guarantee that there is an edge \( e \in \delta(X) \cap E(k) \) such that the path \( P_k \) indeed crosses the cut \((X, V \setminus X)\) for each \( X \) with \( o_k \in X \) and \( d_k \in V \setminus X \).

Hence, origin, destination and path-continuity constraints together imply that \( E(k) \) contains an \((o_k, d_k)\)-path \( P_k \). It is left to prevent \( E(k) \) from having more edges than needed for \( P_k \) and \( P_k \) from having a length exceeding the transmission reach of demand \( k \).

For that, we use as in [6], [16] degree constraints

\[
\sum_{e \in \delta(v)} x_{ek}^k \leq 2, \text{ for all } k \in K, \text{ and all } v \in V \setminus \{o_k, d_k\}
\]

to prevent that more than two edges from \( E(k) \) are incident to any node. Furthermore, we newly propose cycle-elimination constraints

\[
\sum_{e \in \delta(X)} x_{ek}^k \geq \begin{cases} 2x_{ek}^k & \text{if } |X \cap \{o_k, d_k\}| = 0 \\ x_{ek}^k & \text{if } |X \cap \{o_k, d_k\}| = 1 \end{cases}
\]

\[
\forall k \in K, \forall e \in E, \forall X \subset V
\]

where \( X \subset V \) denotes a subset of nodes containing both endnodes of edge \( e \), to avoid cycles isolated from \( P_k \) (note that isolated edges also fall into this case).

Moreover, we newly propose a transmission-reach constraint

\[
\sum_{e \in E} l_e x_{ek}^k - \ell_k \sum_{e \in \delta(o_k)} x_{ek}^k \leq 0, \text{ for all } k \in K
\]

\[
\text{to ensure that the length of } P_k \text{ does not exceed the transmission reach of } k \text{ if the demand } k \text{ is accepted, otherwise all the variables } x_{ek}^k \text{ are forced to equal zero.}
\]

When we optimize the number of edges used for the routing, we need in addition the following constraints

\[
a_e - x_{ek}^k \geq 0, \text{ for all } k \in K, \text{ and all } e \in E
\]

\[
to force } a_e = 1 \text{ when } x_{ek}^k = 1 \text{ for some } k \in K, \text{ and}
\]

\[
a_e \leq \sum_{k \in K} x_{ek}^k, \text{ for all } e \in E
\]

\[
to guarantee } a_e = 0 \text{ if edge } e \text{ is not used in any routing.}
\]

For the spectrum assignment, we have to guarantee that, whenever demand \( k \) is accepted and an \((o_k, d_k)\)-path \( P_k \) has been selected,

- a channel \( S_0 \subset S \) of \( w_k \) consecutive frequency slots is assigned to \( k \),
- this channel is the same on all edges of \( P_k \) and disjoint from the channels \( S_{k'} \) of all other demands \( k' \) routed along an edge of \( P_k \).

We newly propose channel selection constraints

\[
\sum_{s = w_k} z_{sk}^k - \sum_{e \in \delta(o_k)} x_{ek}^k = 0, \text{ for all } k \in K
\]

\[
\text{that do not allow to assign a channel to demand } k \text{ when no path } P_k \text{ is selected (by not allowing to assign a slot } s \text{ as last slot in the channel), but force to select such a last slot in the channel whenever a path is leaving } o_k \). In addition, we specify the available last slots for the channel of demand \( k \) by forbidden-slot constraints

\[
\sum_{s = 1}^{w_k - 1} z_{sk}^k = 0, \text{ for all } k \in K,
\]

\[
to prevent } demand k \text{ to occupy a slot } s \text{ as last slot in the channel whenever } s < w_k \). Klinkowski et al. [9] proposed a similar idea using demand-edge-first-slot variables.
We newly propose edge-slot constraints
\[ \sum_{s \in S} t_{e,s}^k - w_k x_e^k = 0, \text{ for all } k \in K \text{ and all } e \in E \] (11)

to ensure that precisely \( w_k \) slots are allocated on edge \( e \) to demand \( k \) if and only if demand \( k \) is routed through edge \( e \).

Spectrum contiguity and continuity are handled by the following new demand-edge-slot constraints
\[ \min(s + w_k - 1, s) \leq x_e^s + \sum_{s' \in S} z_{e,s'}^k - t_{e,s}^k \leq 1, \forall k \in K, \forall e \in E, \forall s \in S \] (12)
to force that slot \( s \) on edge \( e \) is allocated to demand \( k \) if and only if demand \( k \) passes through edge \( e \) and slot \( s \) belongs to the channel assigned to demand \( k \) (which is the case if one slot \( s' \in \{s, \ldots, s + w_k - 1\} \) is the last slot of the channel).

We newly propose non-overlapping constraints
\[ \sum_{k \in K} t_{e,s}^k \leq 1, \text{ for all } e \in E \text{ and all } s \in S \] (13)
to ensure that a slot \( s \) on edge \( e \) can be allocated to at most one demand.

When we optimize objective functions involving max-used slot positions, we newly propose two additional constraints
\[ s t_{e,s}^k - p_e \leq 0, \text{ for all } k \in K, \text{ all } e \in E \text{ and all } s \in S \] (14)
to guarantee that no slot \( s \) above \( p_e \) is used on edge \( e \) and
\[ p_e - \sum_{k \in K} \sum_{s \in S} s t_{e,s}^k \leq 0, \text{ for all } e \in E \] (15)
to force the max used slot position on edge \( e \) to equal 0 if no demand is routed through edge \( e \), set the bounds \( p_e \leq p \leq s \), and force \( p_e \in \mathbb{N} \) for all \( e \in E \) and \( p \in \mathbb{N} \) to be integral. Finally, we force all other variables to be binary and require non-negativity for all variables.

c) Objective functions: With the help of these variables, the considered objective functions read as follows:
- \( \min \sum_{s \in E, k \in K} x_e^k \) to minimize the sum of number of hops in the paths,
- \( \min \sum_{e \in E} a_e \) to minimize the number of edges used for the routing, and
- \( \min p \) to minimize the max-used slot position.

Recall that our model encodes the general variant of the RSA problem when demands may be rejected. This situation does not comply with the objective functions studied in [1], [16], [19] (as for all three objective functions, rejecting all demands would yield the optimal solution, with objective function value equal to 0). Our model can be easily adapted to the special case where all demands have to be served, by requiring equality in the origin constraint (1) and simplifying the constraints (2), (3), (6) and (9) by replacing the term \( \sum_{e \in E} \delta(\alpha_k) x_e^k \) by 1.

IV. COMPARISON OF EDGE-NODE FORMULATIONS

All three edge-node formulations from [1], [16], [19] for the RSA problem are compact models as both the numbers of variables and constraints grow only polynomially in the size of the instance, i.e., in the size of the network \( G = (V,E) \) (measured by \( |V| \) and \( |E| \)), the width of the optical spectrum \( S \) (measured by \( |S| \)), and the number of demands (measured by \( |K| \)). Table I summarizes the order of the number of variables and constraints for the three models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Number of Variables</th>
<th>Number of Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model in [16]</td>
<td>( O(</td>
<td>K</td>
</tr>
<tr>
<td>Model in [19]</td>
<td>( O(</td>
<td>K</td>
</tr>
<tr>
<td>Model in [1]</td>
<td>( O(</td>
<td>K</td>
</tr>
</tbody>
</table>

Our model uses also a polynomial number of variables, namely \( O(|K||E||S|) \), but an exponential number of constraints due to
- path-continuity constraints (3) for all subsets \( X \subset V \) with \( o_k \in X, d_k \in V \setminus X \), for all demands \( k \in K \),
- cycle-elimination constraints (5) for all subsets \( X \subset V \) containing both endnodes of edge \( e \), for all edges \( e \in E \) and all demands \( k \in K \).

Recall that path-continuity constraints (3) are used to force that the set \( E(k) \) of edges selected for the routing of demand \( k \) contains an \((o_k, d_k)\)-path \( P_k \), whereas cycle-elimination constraints (5) are used to prevent \( E(k) \) from containing cycles isolated from \( P_k \), see Figure 2 for illustration. None of the models from [1], [16], [19] can exclude the occurrence of cycles isolated from \( P_k \), the model presented in [19] can even not exclude cycles attached to \( P_k \), see Figure 3 for illustration.

Fig. 2. A set \( E(k) \) containing an \((o_k, d_k)\)-path \( P_k \) together with a cycle isolated from \( P_k \).

Fig. 3. A set \( E(k) \) containing an \((o_k, d_k)\)-path \( P_k \) together with a cycle attached to \( P_k \).
In addition, none of the three models checks whether the transmission reach of routing paths is respected. Hence, all three models from [1], [16], [19] are incomplete as their feasible region is a superset of all feasible solutions of the RSA problem and can, thus, handle only some objective functions (where the optimal solution does neither contain cycles isolated from $P_k$ nor cycles attached to $P_k$).

Our model is the first complete edge-node formulation for the RSA problem as it precisely encodes the set of all feasible solutions, i.e., any integral vector satisfying all constraints from our model indeed corresponds to a feasible solution of the RSA problem. Therefore, our model can be used to optimize any objective function chosen as quality measure by the network operator.

In addition, our model is not only complete, but still tractable as we are able to separate the two exponentially-sized families of constraints (3) and (5) in polynomial time.

In fact by the polynomial equivalence between separation and optimization over rational polyhedra [5], the linear relaxations of our model can be solved in polynomial time if and only if the separation problem associated with inequalities (3) and (5) can be solved in polynomial time. The separation problem for the path-continuity constraints (3) reduces to $O(|K|)$ minimum-cut problems in $G$ and the separation problem for the cycle-elimination constraints (5) to $O(|K||E|)$ minimum-cut problems in an auxiliary graph.

Therefore the separation problem associated with (3) and (5) is polynomially solvable using any polynomial-time maximum-flow algorithm (e.g., the preflow-push algorithm of Goldberg and Tarjan [4] running in $O(|V|^3)$ time). Note that this separation approach provides the most-violated inequality if any w.r.t. a demand or a pair of a demand and an edge.

V. Computational results

In this section we present some preliminary computational results that mainly aim at assessing the empirical performances of a branch-and-cut framework based on our model for the three objectives functions presented in Section II and at comparing them with the results obtained by Velasco et al. [16] for $O_2$ and by Cai et al. [1] for objective $O_3$.

In our experiments we therefore consider the Spanish Telefónica network represented in Figure 4 from [16] and three networks represented in Figure 5 from [1]. The characteristics of the topology of these four networks are given in Table II together with the available numbers of slots per link.

As none of the instances considered in [1], [16] were available, we randomly generated multisets of traffic demands, some of them using Net2Plan [12], while guaranteeing that some of those multisets share the properties described in [1], [16], that is, the same number of traffic demands (12 for Spanish Telefónica and 30 for n6s9) and the same range of values for the requested numbers of slots (in $\{1,2,4\}$ for Spanish Telefónica and in $\{1,\ldots,3\},\ldots,\{1,\ldots,9\}$ for n6s9). Table III summarizes the different types of traffic-demand multisets we considered for each network.

All our results were obtained on a laptop, running Microsoft Windows 10 Pro (64-bit), equipped with a 2.5GHz Intel Core i5-7300 HQ processor and 16-GB RAM. The branch-and-cut framework was implemented using IBM ILOG CPLEX Optimization Studio 12.8 C++ library. Note that using user-cut callbacks (needed for the separation of constraints (3) and (5)) in CPLEX 12.8 automatically deactivates the multithreading. To balance some struggles that the default heuristic of CPLEX has to generate good feasible solutions, we implemented a heuristic callback based on

- first decomposing for each demand $k \in K$ its flow (given by the $x_{lk}$-variables) into $(a_k, d_k)$-paths and
- second using a first-fit greedy approach to assign the best possible channels to the demands,

The first objective function $O_1$ was considered in neither [1] nor [16]. Within a one-hour time limit, our branch-and-cut framework was able to solve to optimality all our instances
TABLE II
Characteristics of the network topologies

<table>
<thead>
<tr>
<th>Network’s name</th>
<th>number of nodes</th>
<th>number of links</th>
<th>number of slots per link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spanish Telefónica</td>
<td>21</td>
<td>65</td>
<td>3</td>
</tr>
<tr>
<td>n6s9</td>
<td>6</td>
<td>9</td>
<td>80</td>
</tr>
<tr>
<td>SmallNet</td>
<td>10</td>
<td>22</td>
<td>[80, 100, 140, 180]</td>
</tr>
<tr>
<td>NSFNET</td>
<td>14</td>
<td>21</td>
<td>[120, 160, 210, 265]</td>
</tr>
</tbody>
</table>

TABLE III
Characteristics of the traffic demands

<table>
<thead>
<tr>
<th>Network’s name</th>
<th>number of demands</th>
<th>number of requested slots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spanish Telefónica</td>
<td>12, 15</td>
<td>1, 2, 4</td>
</tr>
<tr>
<td>n6s9</td>
<td>30, 50</td>
<td>1, …, 9</td>
</tr>
<tr>
<td>SmallNet</td>
<td>100, 150, …, 500</td>
<td>1, …, 4</td>
</tr>
<tr>
<td>NSFNET</td>
<td>100, 150, …, 250</td>
<td>2, …, 6</td>
</tr>
</tbody>
</table>

but the ones with 500 demands for which the optimality gap was under 0.5%. Over the course of the solution process, both the lower and upper bounds kept improving and only towards the end, optimal solutions were found.

For the second objective function $O_2$, Velasco et al. [16] were able to solve to optimality a single instance of Spanish Telefónica with 12 demands in over 6 hours. It took less than 3 hours for our branch-and-cut framework to solve to optimality the Spanish Telefónica instances with 12 demands and less than 6 hours for the Spanish Telefónica instances with 15 demands. We also ran our branch-and-cut framework on all the instances associated with n6s9 and were able to get optimal solutions within at most 15 minutes. Very early in the solution process, optimal solutions were found meaning that most of the solution time is dedicated to proving the optimality of those solutions (e.g., for the Spanish Telefónica with 12 demands, an optimal solution is found after about 15 minutes but proved optimal after about 2 hours and 40 minutes).

Cai et al. [1] only considered the third objective function $O_3$ in their experiments with the additional property that given any two distinct nodes $o$ and $d$ of $G$, the multiset $K$ of traffic demands contains either both demands having nodes $o$ and $d$ as their extremities (with the same requested number of slots) or none of them, and for the former case one assigned route is the reverse of the other one. Some of our generated instances for n6s9 fulfilled that property and were all solved to optimality within 20 minutes while Cai et al. [1] needed up to one hour to solve their similar instances (with CPLEX multithreading being active). We also ran our branch-and-cut framework on n6s9 instances without the reverse-demand property and for most of the instances were able to find optimal solutions within two hours and an optimality gap lower than 5% for the others. We noticed a similar behavior of the lower and upper bounds as for objective function $O_1$.

VI. CONCLUDING REMARKS

The RSA problem in flexgrid elastic optical networks is an NP-hard problem for which various ILP models have been proposed in the literature. Hereby, detailed models aiming at precisely describing all technological aspects and being able to handle different criteria for optimization typically suffer from tractability issues resulting from their greater complexity such that the tendency is to use simplified models.

The majority of the existing models uses edge-path formulations where the numbers of variables and constraints grow exponentially with the size of the instance, due to the huge number of feasible paths between all origin-destination pairs in the network. Hence, models based on edge-path formulations are often simplified by considering only subsets of precomputed paths (which cannot guarantee optimality, except for few objective functions) or require column-generation techniques (which limits the size of the instances that can be solved to optimality).

An alternative to edge-path formulations is to use edge-node formulations that have the advantage that the number of variables grows only polynomially with the size of the instance. Three compact edge-node formulations are presented in [1], [16], [19] where both the number of variables and constraints is polynomial in terms of the size of the instance. However, all three models are incomplete as their feasible region is a superset of all feasible solutions of the RSA problem and can, thus, handle only some objective functions.

Our contribution is to provide the first complete edge-node formulation for the RSA problem that precisely encodes the set of all feasible solutions and can, therefore, be used to optimize any chosen objective function. For that, we propose an appropriate combination of variables and constraints (partly using new variables and constraints) which results in a model having, as in [1], [16], [19], a polynomial number of variables, but an exponential number of constraints to ensure the exact encoding of feasible solutions.

As we are able to separate the exponentially-sized families of constraints in polynomial time, our model is computationally competitive with the compact but incomplete models from [1], [16], [19]. The computational results support this as our branch-and-cut solver was able, on the one hand, to efficiently handle larger instances and, on the other hand, to find optimal solutions for instances similar to those in [1], [16] in shorter time.

Hereby, we noticed by analyzing the computational results for objective function $O_2$ that for most instances the optimal solution was found early in the computation process, but that most of the computation time was needed to certify its optimality. Hence, our future research also includes to strengthen lower bounds for the value of different objective functions in order to shorten the time during the computation needed for certifying optimality of a solution.

Therefore, we plan as future research, on the one hand, to strengthen our model further by devising new inequalities, e.g. derived as Chvátal-Gomory cuts from the initial constraints, and, on the other hand, to further improve the separation procedure for the exponentially-sized families of constraints.

Finally, recall that many different objective functions may be considered, depending on the network operator’s choice. Besides $O_1$, $O_2$, $O_3$, the following objective functions may be of interest:
Hence, it is also in order to develop strategies to cope with these problems. Hereby, the optimal solutions w.r.t. different objective functions may significantly differ such that an optimal solution for one objective may provide rather bad values according to other optimality criteria. For instance, the three optimal solutions presented in Example 2.1 (for $O_1$, $O_2$ for $O_3$, $O_4$ for $O_5$) also optimal for $O_3$ minimizing the maximum edge load of 3) differ from each other and from the optimal solution for $O_4$ and $O_5$ presented in $M_4$ (with minimum total length 13 of paths and minimum total cost 22).

\[
M_4 = \begin{pmatrix}
ab & 1 & 2 \\
af & 1 & 1 & 2 \\
b & 1 & 1 & 3 & 2 & 4 \\
c & 3 & 3 \\
d & 5 & 5 & 5 \\
e & 5 & 5 & 5 \\
\end{pmatrix}
\]

We notice that the objective functions

- $O_1$, $O_2$, $O_4$ for the routing may lead to solutions where some edges are highly loaded (with 6 slots in $M_1$, $M_2$ and $M_4$) where 3 slots suffice as in $M_3$) which also forces a large used spectrum width (6 slots in $M_1$, $M_2$ and $M_4$) where 4 slots suffice as in $M_3$).

- $O_3$ and $O_5$ for the spectrum assignment may lead to routings along longer paths (total length of 17 in $M_3$) which also increase the total cost of the solution (29 for $M_3$ where 22 suffice as in $M_4$).

Hence, it is also in order to develop strategies to cope simultaneously with different quality measures of solutions.

ACKNOWLEDGMENT

This work was supported by the French National Research Agency grant ANR-17-CE25-0006, project FLEXOPTIM.

REFERENCES


