Journal of Combinatorial Optimization (2024) 47:56
https://doi.org/10.1007/s10878-024-01125-x

Branch-and-cut-and-price algorithm for the constrained-routing and spectrum assignment problem

Ibrahima Diarrassouba[1] · **Youssouf Hadhbi**[2] · **A. Ridha Mahjoub**[3,4]

Accepted: 21 February 2024 / Published online: 14 April 2024
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

The Constrained-Routing and Spectrum Assignment (C-RSA) problem arises in the design of $5G$ telecommunication optical networks. Given an undirected, loopless, and connected graph G, an optical spectrum of available contiguous frequency slots $\mathbb{S}$, and a set of traffic demands K, the C-RSA consists of assigning, to each traffic demand $k \in K$, a path in G between its origin and destination, and a subset of contiguous frequency slots in $\mathbb{S}$ subject to certain technological constraints while optimizing some linear objective function. In this paper, we devise an exact algorithm to solve the C-RSA. We first introduce an extended integer programming formulation for the problem. Then we investigate the associated polytope and introduce several classes of valid inequalities. Based on these results, we devise a Branch-and-Cut-and-Price algorithm for the problem and present an extensive computational study. This is also be compared with a Branch-and-Cut algorithm of the state-of-the-art.

Keywords $5G$ Network design · Branch-and-Cut-and-Price · Valid inequalities

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

✉ Youssouf Hadhbi
youssouf.hadhbi@uca.fr

Ibrahima Diarrassouba
diarrasi@univ-lehavre.fr

A. Ridha Mahjoub
ali.mahjoub@ku.edu.kw; ridha.mahjoub@lamsade.dauphine.fr

1 LMAH, FR CNRS 3335, Le Havre Normandie University, 76600 Le Havre, France

2 LIMOS, UMR CNRS 6158, Clermont Auvergne INP, 63178 Clermont Ferrand, France

3 Departement of Statistics and Operations Research, Kuwait University, Kuwait City, Kuwait

4 LAMSADE, UMR CNRS 7243, Paris-Dauphine PSL University, 75775 Paris Cedex 16, France

1 Introduction

The last decade has seen deep changes in optical transport networks with continuous growth in bandwidth capacity due to the growth of global communication services and networking like the mobile internet network. Therefore, a new generation of optical transport network architecture called Spectrally Flexible Optical Networks (SFONs) has been introduced as a promising technology because of its flexibility and efficiency compared with the traditional Fixed-Grid Optical Wavelength Division Multiplexing (WDM) (Ramaswami 2009; Ramaswami et al. 1993). In SFONs, the optical spectrum is divided into slots having the same frequency of 12.5 GHz where WDM uses 50 GHz as recommended by ITU-T (Amar 2016). We refer the reader to Lopez and Velasco (2016) for more details about the architectures, technologies, and control of SFONs. With SFONS, Telecommunications operators are faced to several optimization problems. In particular some variants of routing and resource assignment problems arise when dimensioning and planning optical networks. The classical Routing and Wavelength Assignment (RWA) problem is the key issue for the design of Fixed-Grid WDM networks. Here, we are given an optical network and a set of demands where each demand has a pair of origin–destination nodes. The task is to find a path for each demand and a wavelength such that a single 50 GHz wavelength is assigned to each demand, while minimizing the total number of assigned wavelength. It was first considered by Bal et al. (1991), and extended by Chlamtac et al. (1992). The RWA is known to be NP-Hard (Chlamtac et al. 1992). It has been shown to be equivalent to the n-graph-coloring problem where the number n of the colors corresponds to the number of wavelengths, and finding the minimum number of wavelengths to route all the traffic demands is equivalent to finding the chromatic number of a conflict graph (where the demands correspond to the nodes and two nodes are linked by an edge if the path of the associated demands share an edge) when the paths are already established. It has also been shown to be equivalent to a special case of the integer multicommodity flow (MCF) problem where some specific constraints are added (Brun and Baraketi 2014). Several models and algorithms have been proposed to solve the RWA problem. These are based on some integer linear programming formulations (Brun and Baraketi 2014; Christodoulopoulos et al. 2010; Liu and Rouskas 2013), decomposition-based methods (Simonis 2011; Zang et al. 2000), and heuristics (Shiva-Kumar and Sreenivasa-Kumar 2002; Siregar et al. 2003; Skorin-Kapov 2007). However, in SFONs, RWA cannot handle the changes from wavelength to contiguous slots. As a consequence, a new problem has appeared to deal with this situation, called *Routing and Spectrum Assignment* (RSA) problem. It consists of assigning for each traffic demand k a physical optical path, and a set of contiguous slots (called also channels), while optimizing some linear objective function and satisfying the following constraints (Hadhbi et al. 2019):

1. *spectrum contiguity*: a set of contiguous slots should be assigned to each demand k with a width equal to the number of slots requested by demand k;
2. *spectrum continuity*: the set of contiguous slots assigned to each demand stills the same along the chosen path;

3. *non-overlapping spectrum*: the sets of slots of demands whose paths are not edge-disjoint cannot share any slot over the shared edges.

The RSA plays a key role when dimensioning and planning SFONs. In recent years, the routing and resources assignment in communication networks have received increasing attention. In particular, numerous works have been conducted on the RSA problem since its first appearance. From a complexity point of view, the RSA is known to be NP-Hard (Shirazipourazad et al. 2013; Talebi et al. 2014). Various integer linear programming formulations have been proposed. In particular, a big amount of work is related to an edge-path based formulation. Heuristics are proposed in Christodoulopoulos et al. (2011), Klinkowski et al. (2010), Klinkowski and Walkowiak (2011), Velasco et al. (2012), Zotkiewicz et al. (2013). These are based on the edge-path formulation by taking into account some precomputed-paths for each traffic demand. Hai and Hoang (2017) propose some hybrid approaches combining heuristics with exact algorithms to solve the RSA problem. On the other hand, column generation based techniques have been used by Ruiz et al. (2013) and Jaumard and Daryalal (2016) to properly manage the set of paths for the edge-path formulation and solve the linear relaxation of the RSA problem. Nguyen et al. (2019) devise also a column generation algorithm to solve the RSA problem. In order to strengthen the LP bounds of the RSA relaxation, Klinkowski et al. (2014) propose a class of valid inequalities related to the non-overlapping constraint. A Branch-and-Cut-and-Price algorithm has been developed by Klinkowski et al. (2015).

Moreover, edge-node based formulations have been considered in Bertero et al. (2018), Cai et al. (2013), Velasco et al. (2012), Zotkiewicz et al. (2013). In Bertero et al. (2018), present a comparative study of these formulations.

In addition to exact approaches, several heuristics (Ding et al. 2014; Mesquita et al. 2018; He et al. 2020), and metaheuristic algorithms (Goscien et al. 2015; Gong et al. 2012; Hai and Hoang 2017; Hai et al. 2017; Klinkowski et al. 2015; Lezama et al. 2021), have been proposed for the RSA problem. A hybrid meta-heuristic approach has been proposed by Ruiz et al. (2020) to solve large-sized instances of the RSA problem. Also artificial intelligence based algorithms have been devised for the RSA problem (Liu et al. 2019; Lohani et al. 2019). Some techniques based on machine learning are also proposed (Chen et al. 2018; Gu et al. 2020; Salani et al. 2019; Zhang et al. 2020).

In this paper, we consider the *Constrained-Routing and Spectrum Assignment* (C-RSA) problem. Here we impose that the network should also satisfy the transmission-reach constraint that is the route for each demand should not exceed a certain length. Today, the transmission-reach constraint is very required by the network operators due to the usage of some equipment called transponder configurations. They are necessary to send an optical signal between the origin node and the destination node of each traffic demand. For this reason, we have studied a real variant of the RSA problem called C-RSA while considering additional constraints related to the transmission-reach constraints.

The C-RSA is more complex and more challenging than the RSA. It can also be seen as a new variant of the well known Routing, Modulation and Spectrum Assignment

(RMSA) problem where the modulation formats (called also transponder configurations) are preselected for each traffic demand.

Recently, Hadhbi et al. (2019) introduced a linear integer programming formulation for the C-RSA problem. This has been used by Chouman et al. (2021) and Chouman et al. (2021) to show the impact of some objective functions on the optical network state, and by Colares et al. (2021) to propose a compact formulation for the problem. Note that the transmission-reach constraint has not been taken into account by Velasco et al. (2012), Cai et al. (2013), and Bertero et al. (2018). Their formulations can easily be adapated to the C-RSA problem. Computational results presented by Hadhbi et al. (2019), show that their formulation solves larger instances compared with those of Velasco et al. (2012) and Cai et al. (2013). Recently, Pedro et al. (2022) propose a new framework for the C-RSA based on an edge-path formulation combined with a Multi-Commodity Flow approach.

Our aim in this paper is to devise an exact algorithm to solve the C-RSA. We propose an extended formulation based on the so-called path formulation. We investigate also the polytope associated with this formulation and introduce several classes of valid inequalities. Using these results, we develop a Branch-and-Cut-and-Price (B&C&P) algorithm by combining a Branch-and-Bound (B&B) algorithm with cutting plane and column generation algorithms. We present an extensive computational study for showing the efficiency of our algorithm.

The rest of this paper is organized as follows. In Sect. 2, we present the C-RSA problem. In Sect. 3, we introduce the extended formulation. In Sect. 5, we describe the associated polytope and introduce some valid inequalities. Our Branch-and-Cut-and-Price algorithm is presented in Sect. 6. In Sect. 7, we discuss our computational results. Finally, Sect. 8 is devoted to some concluding remarks and future work.

In the rest of this section, we give more definitions and notations that are used in the sequel.

A graph is denoted by $G = (V, E)$ where V represents the set of nodes and E the set of edges. G is said to be connected if for each pair of nodes $u, v \in V$ there exists at least one path between u and v in G. A path p in a graph $G = (V, E)$ is a sequence of edges in G. For a path p of G, $E(p)$ will denote the set of edges of p. Also, for an edge $e \in E$, $p(e)$ denotes the set of paths going through e in G. A subset of nodes C in a graph $G = (V, E)$, is said to be clique if every two different nodes in C are linked by an edge in G which is equivalent to say that the induced subgraph is complete. Given two nodes s, t of G, an (s, t)-path is a path between s and t. A graph $G = (V, E)$ is said to be conflict graph if there exists a logical relation between each node and its complement in G such that each node in G represents a binary variable for an integer programming problem. An edge is formed between each node v and its complement in a conflict graph G, which is equivalent to say that atmost one of the variables represented by node v and its complement can equal to one in a feasible solution of the problem. Two distinct nodes u and v are said to be adjacent in a graph $G = (V, E)$ if u and v form an edge in G. Given a graph $G = (V, E)$, a subet of nodes C form a cycle in G if C can be represented as a sequence of adjacent and distinct nodes in G such that we can start from each node v in C and ends at the same node v. An optical spectrum $\mathbb{S}$ can be seen as a set of contiguous frequency slots numbered from 1 to $\bar{s}$. An interval $I = [s_i, s_j]$ of slots is a set of contiguous frequency slots

numbered from s_i to s_j with $j \geq i+1$ and $s_j \leq \bar{s}$. The cardinality of any given set of elements S is denoted by $|S|$.

2 The constrained-routing and spectrum assignment problem

The Constrained-Routing and Spectrum Assignment Problem can be stated as follows. Consider a spectrally flexible optical network that can be seen as an undirected, loopless, and connected graph $G = (V, E)$ with lengths $\ell_e \in \mathbb{R}_+$ (in kms) and costs $c_e \in \mathbb{R}_+$ on the edges $e \in E$. Each fiber-link $e \in E$ is associated with $\bar{s} \in \mathbb{N}_+$ contiguous slots numbered from 1 to $\bar{s}$. Let $\mathbb{S} = \{1, \ldots, \bar{s}\}$ be the set of available frequency slots. Also, let K be a set of demands. Each demand $k \in K$ is given by its origin node $o_k \in V$, a destination node $d_k \in V \backslash \{o_k\}$, a slot-width requirement $w_k \in \mathbb{Z}_+$, and a transmission-reach $\bar{\ell}_k \in \mathbb{R}_+$ (in kms). The C-RSA problem consists in determining for each demand $k \in K$, an (o_k,d_k)-path p_k in G whose total length does not exceed $\bar{\ell}_k$ (i.e., $\sum_{e \in E(p_k)} \ell_e \leq \bar{\ell}_k$). Also, each demand $k \in K$ must be assigned a subset S_k of w_k contiguous slots, in such a way that the slot assignments satisfy the so-called non-overlapping constraints. Non-overlapping constraints here mean that if two distinct demands k and k' are assigned to paths p_k and $p_{k'}$, which share an edge, then the slots subsets S_k and $S_{k'}$, assigned to k and k' must be disjoint. Finally, the total routing cost, i.e. $\sum_{k \in K} \sum_{e \in E(p_k)} c_e$ must be minimized.

Figure 1, below, gives an illustration of a feasible solution of the C-RSA problem for a graph with 7 nodes, 10 edges and 4 demands. Here $\bar{s} = 9$, and each link e has a triplet $(\ell_e, c_e, \bar{s})$. We can remark that the slots assigned to each demand are contiguous and each pair of demands whose paths intersect (like demands 1 and 2, and 2 and 4) must have disjoint slot sets.

In what follows, we introduce an extended formulation for the C-RSA problem.

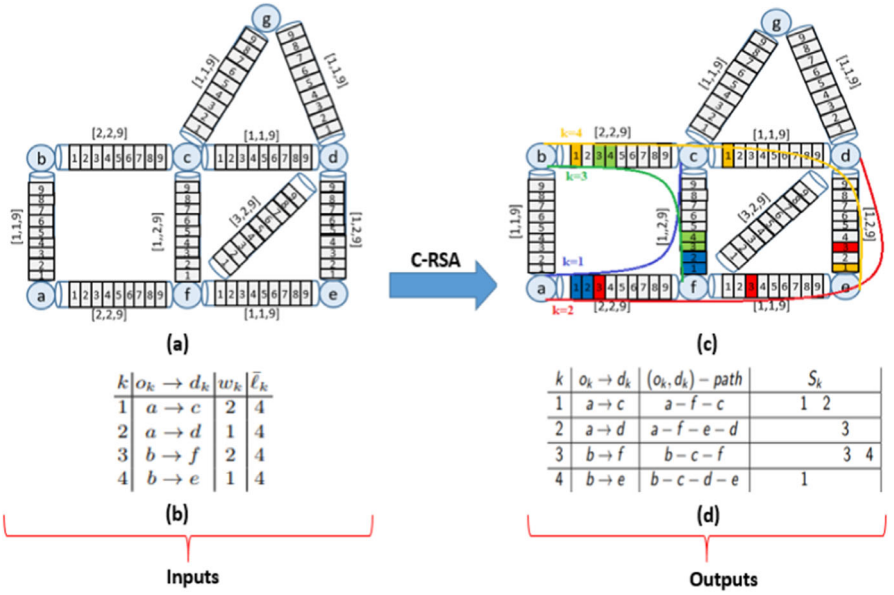
3 Integer linear programming formulation

For every $k \in K$, let P^k denote the set of (o_k,d_k)-paths p_k in G such that $\sum_{e \in E(p_k)} \ell_e \leq \bar{\ell}_k$. For $k \in K$, $p \in P^k$ and $s \in \mathbb{S}$, let $y^k_{p,s}$ be a variable which takes value 1 if slot s is the last slot considered on path p for routing demand k, and 0 if not. For $k \in K$ and $e \in E$, let x^k_e be a variable which takes value 1 if demand k goes through edge e and 0 if not. Also, for $k \in K$ and $s \in \mathbb{S}$, let z^k_s be a variable which takes value 1 if slot s is the last slot assigned for the routing of demand k and 0 if not. The C-RSA problem is equivalent to the following integer linear program

$$\min \sum_{k \in K} \sum_{e \in E} c_e x^k_e, \tag{1}$$

subject to

$$\sum_{p \in P^k} \sum_{s=1}^{w_k - 1} y^k_{p,s} = 0, \forall k \in K, \tag{2}$$



k	$o_k \to d_k$	w_k	$\bar{\ell}_k$
1	$a \to c$	2	4
2	$a \to d$	1	4
3	$b \to f$	2	4
4	$b \to e$	1	4

k	$o_k \to d_k$	$(o_k, d_k) - path$	S_k
1	$a \to c$	$a - f - c$	1 2
2	$a \to d$	$a - f - e - d$	3
3	$b \to f$	$b - c - f$	3 4
4	$b \to e$	$b - c - d - e$	1

Fig. 1 Instance of the C-RSA and a feasible solution

$$\sum_{p \in P^k} \sum_{s=w_k}^{\bar{s}} y_{p,s}^k = 1, \forall k \in K, \tag{3}$$

$$x_e^k - \sum_{p \in P^k(e)} \sum_{s=w_k}^{\bar{s}} y_{p,s}^k = 0, \forall k \in K, \forall e \in E, \tag{4}$$

$$z_s^k - \sum_{p \in P^k} y_{p,s}^k = 0, \forall k \in K, \forall s \in \mathbb{S}, \tag{5}$$

$$\sum_{k \in K} \sum_{p \in P^k(e)} \sum_{s'=s}^{\min(s+w_k-1,\bar{s})} y_{p,s'}^k \leq 1, \forall e \in E, \forall s \in \mathbb{S}, \tag{6}$$

$$y_{p,s}^k \geq 0, \forall k \in K, \forall p \in P^k, \forall s \in \mathbb{S}, \tag{7}$$

$$x_e^k \geq 0, \forall k \in K, \forall e \in E, \tag{8}$$

$$z_s^k \geq 0, \forall k \in K, \forall s \in \mathbb{S}, \tag{9}$$

$$y_{p,s}^k \in \{0, 1\}, \forall k \in K, \forall p \in P^k, \forall s \in \mathbb{S}. \tag{10}$$

Constraints (2) express the fact that a demand $k \in K$ cannot have a slot s as the last slot before its slot-width w_k. Constraints (3) ensure that exactly one slot $s \in \{w_k, \ldots, \bar{s}\}$ is assigned as last slot for the routing of demand k, and exactly one single path from P^k can be used by each demand $k \in K$. Equations (4) and (5) express the use of an edge $e \in E$ and the assignment of slot s as last-slot by demand k, respectively. Constraints (6) impose that a slot s on an edge e can be used by at most one demand $k \in K$ going

through e. Constraints (7)–(9) are the trivial inequalities, and constraints (10) are the integrality constraints. Remark that in this formulation, we do not explicitly mention that variables x_e^k and z_s^k are binary. Indeed, it is not hard to see that if variables $y_{p,s}^k$ satisfy constraints (2), (3) and (10), then equations (4), (5) and (6) imply that variables x_e^k and z_s^k are binary.

Note that variables x and z are not strictly necessary for the formulation. The main role of these variables is to introduce some valid inequalities to strengthen the linear relaxation of our formulation.

On the other hand, we ensure that all the demands K should be routed. For this, the spectrum width $\bar{s}$ must be sufficient to route these demands. In what follows, we describe how the feasibility of an instance is examined before solving the problem.

4 Instance feasibility

In practice, network operators consider the values of $\bar{s}$ and $\bar{\ell}_k$ such that all the demands can be routed in the network. In this section, we propose a heuristic to check if an instance $(G, K, \mathbb{S})$ is feasible.

First, we use a shortest-path algorithm to check if the transmission-reach $\bar{\ell}_k$ of each demand k is feasible that is to say if there exists at least one feasible (o_k, d_k) path for demand k. If not, the problem is infeasible. Otherwise, we then check if $\bar{s}$ is sufficient to route the demands. If not we will keep increasing $\bar{s}$ until we obtain a feasible one that is $\bar{s}$ which enables the routing of all the demands. This is based on the following procedure.

We first enumerate a set of feasible paths $B^k \subseteq P^k$ for each demand k (30 feasible (o_k, d_k)-paths for each demand k) using a brute force procedure. This aims at creating a search tree that contains these paths. It can also be seen as a dynamic programming algorithm.

Using the precomputed paths B^k for each demand $k \in K$, we check if $\bar{s}$ is sufficient to route all the demands while using one path p from the set of paths B^k and an interval of slots S_k in $\mathbb{S}$ for each demand $k \in K$. For this, we use a Branch-and-Bound algorithm based on a new formulation. This formulation shares the same variables and constraints with our previous MILP while taking $P^k = B^k$, modifying equations (3) as follows

$$\sum_{p \in P^k} \sum_{s=w_k}^{\bar{s}} y_{p,s}^k \leq 1, \forall k \in K.$$

Then, we check if there exist some demands that have been rejected due to a lack of resources that is to say $\bar{s}$ is not sufficient to route all the demands. If this is the case, then we increase the spectrum width $\bar{s}$ by 40. If $\bar{s} = 320$, and some rejected demands still exist, then we duplicate the most occupied edges (e.g., 80% of their capacities are used). We solve the problem again while minimizing the number of rejected demands, and keep adding slots or / and edges until no demand is rejected.

Using this procedure, we guarantee the feasibility of the instance, and therefore the existence of at least one feasible solution for the problem.

In the next section we describe some classes of valid inequalities for the C-RSA.

5 Valid Inequalities

Given an instance $(G, K, \mathbb{S})$ of the C-RSA, we let $P(G, K, \mathbb{S})$ denote the polytope, convex hull of all the solutions of program (1)–(10). For a demand $k \in K$, let E_0^k (resp. E_1^k) denote the set of all *forbidden* (resp. *essential*) edges for demand k that is to say the set of edges e such that the length of each (o_k,d_k)-path in G going through e exceeds $\bar{\ell}_k$ (resp. each (o_k, d_k)-path of length smaller than $\bar{\ell}_k$ uses e). Note that sets E_0^k and E_1^k can be determined in polynomial time using Dijkstra's algorithm (1959) and network flows. For an edge $e \in E$, let K_e denote the subset of demands of K having e as essential edge.
More details concerning the computation of E_0^k and E_1^k will be provided in Sect. 6 within the framework of the Branch-and-Cut-and-Price algorithm.

5.1 Cover-based Inequalities

The first class of inequalities can be seen as cover inequalities related to the contiguity and non-overlapping constraints.

5.1.1 Edge-Interval-Capacity-Cover Inequalities

Given an interval of contiguous slots $I = [s_i, s_j]$, a subset of demands $\tilde{K} \subseteq K$ is said to be a *cover* for I if and only if $\sum_{k\in\tilde{K}} w_k > |I|$ and $w_k < |I|$ for each $k \in \tilde{K}$. And, it is said to be *minimal* if in addition $\sum_{k'\in\tilde{K}\setminus\{k\}} w_{k'} \leq |I|$ for each demand $k \in \tilde{K}$.
Given an edge $e \in E$, a subset $\tilde{K}$ of demands is said to be a minimal cover for I over edge e if and only if $\tilde{K}$ is a minimal cover for I and $e \notin E_0^k$ for each demand $k \in \tilde{K}$.

Proposition 1 (Hadhbi 2022) *Let $e \in E$, and $I = [s_i, s_j]$ be an interval in $[1, \bar{s}]$. Let $\tilde{K}$ be a minimal cover for I. Then, the inequality*

$$\sum_{k\in\tilde{K}} x_e^k + \sum_{k\in\tilde{K}} \sum_{s=s_i+w_k-1}^{s_j} z_s^k \leq 2|\tilde{K}| - 1, \tag{11}$$

is valid for $P(G, K, \mathbb{S})$.

5.1.2 Edge-capacity-cover inequalities

Next, we give some inequalities related to the capacity constraint on the edges.

Proposition 2 (Hadhbi 2022) *Consider an edge e in E. Then, the inequality*

$$\sum_{k\in K\setminus K_e} w_k x_e^k \leq \bar{s} - \sum_{k'\in K_e} w_{k'}, \tag{12}$$

is valid for $P(G, K, \mathbb{S})$.

For an edge $e \in E$, a subset of demands $C \subseteq K$ with $e \notin E_0^k \cap E_1^k$ for each demand $k \in C$, is said to be a cover if $\sum_{k \in C} w_k > \bar{s} - \sum_{k' \in K_e} w_{k'}$. Moreover, C is said to be minimal if in addition $\sum_{k' \in C \setminus \{k\}} w_{k'} \leq \bar{s} - \sum_{k" \in K_e} w_{k"}$ for each $k \in K$.

Proposition 3 (Hadhbi 2022) *Consider an edge e in E, and C a minimal cover in K. Then, the inequality*

$$\sum_{k \in C} x_e^k \leq |C| - 1, \tag{13}$$

is valid for $P(G, K, \mathbb{S})$.

5.2 Clique-based inequalities

5.2.1 Edge-interval-clique inequalities

Let $e \in E$, and $I = [s_i, s_j]$ be an interval in $[1, \bar{s}]$ with $s_i \leq s_j - 1$. The *conflict graph* H_I^e with respect to e and I is the graph defined as follows. For each demand $k \in K$ with $w_k \leq |I|$ and $e \notin E_0^k$ consider a node v_k in H_I^e. Two nodes v_k and $v_{k'}$ are linked by an edge in H_I^e if $w_k + w_{k'} > |I|$. That is to say v_k and $v_{k'}$ form an edge in H_I^e if the two correponding demands k, k' define a minimal cover for I over e.
In the rest of this section, we introduce valid inequalities that come from conflict graphs.

Proposition 4 (Hadhbi 2022) *Consider an edge $e \in E$. Let $I = [s_i, s_j]$ be an interval in $[1, \bar{s}]$. Let C be a clique in a conflict graph H_I^e with $|C| \geq 3$, and $\sum_{v_k \in C} w_k \leq \bar{s} - \sum_{k' \in K_e \setminus C} w_{k'}$. Then, the inequality*

$$\sum_{v_k \in C} x_e^k + \sum_{s=s_i+w_k-1}^{s_j} z_s^k \leq |C| + 1, \tag{14}$$

is valid for $P(G, K, \mathbb{S})$.

5.2.2 Interval-clique inequalities

Proposition 5 (Hadhbi 2022) *Consider an interval $I = [s_i, s_j]$ in $[1, \bar{s}]$. Let k, k' be a pair of demands in K with $E_1^k \cap E_1^{k'} \neq \emptyset$, $w_k \leq |I|$, $w_{k'} \leq |I|$, and $w_k + w_{k'} > |I|$. Then, the inequality*

$$\sum_{s=s_i+w_k-1}^{s_j} z_s^k + \sum_{s'=s_i+w_{k'}-1}^{s_j} z_{s'}^{k'} \leq 1, \tag{15}$$

is valid for $P(G, K, \mathbb{S})$.

Using inequalities (15), we introduce the following conflict graph.
Let $I = [s_i, s_j]$ be an interval in $[1, \bar{s}]$. Consider a conflict graph H_I^E defined as follows. For each demand $k \in K$ with $w_k \leq |I|$ consider a node v_k in H_I^E. Two nodes v_k and $v_{k'}$ are linked by an edge in H_I^E if $w_k + w_{k'} > |I|$ and the two demands k and k' share an essential edge in G, i.e., $E_1^k \cap E_1^{k'} \neq \emptyset$. We have the following.

Proposition 6 (Hadhbi 2022) *If C is a clique in a conflict graph H_I^E with $|C| \geq 3$, then, the inequality*

$$\sum_{v_k \in C} \sum_{s=s_i+w_k-1}^{s_j} z_s^k \leq 1, \tag{16}$$

is valid for $P(G, K, \mathbb{S})$.

5.2.3 Edge-slot-assignment-clique inequalities

Let H_S^e be a conflict graph defined as follows. For each slot $s \in \{w_k, \ldots, \bar{s}\}$ and demand $k \in K$ with $e \notin E_0^k$ consider a node $v_{k,s}$ in H_S^e. Two nodes $v_{k,s}$ and $v_{k',s'}$ are linked by an edge in H_S^e if either

- $k = k'$ or
- or $\{s - w_k + 1, \ldots, s\} \cap \{s' - w_{k'} + 1, \ldots, s'\} \neq \emptyset$.

Proposition 7 (Hadhbi 2022) *If C is a clique in a conflict graph H_S^e with $|C| \geq 3$, and $\sum_{k \in C} w_k \leq \bar{s} - \sum_{k' \in K_e \setminus C} w_{k'}$, then, the inequality*

$$\sum_{v_{k,s} \in C} (x_e^k + z_s^k) \leq |C| + 1, \tag{17}$$

is valid for $P(G, K, \mathbb{S})$.

5.2.4 Slot-assignment-clique inequalities

Let H_S^E be a conflict graph defined as follows. For every slot $s \in \{w_k, \ldots, \bar{s}\}$ and demand $k \in K$ consider a node $v_{k,s}$ in H_S^E. Two nodes $v_{k,s}$ and $v_{k',s'}$ are linked by an edge in H_S^E if either $k = k'$ or $E_1^k \cap E_1^{k'} \neq \emptyset$ and $\{s - w_k + 1, \ldots, s\} \cap \{s' - w_{k'} + 1, \ldots, s'\} \neq \emptyset$ when $k \neq k'$ (due to the constraints (6)). We have the following.

Proposition 8 (Hadhbi 2022) *If C is a clique in H_S^E with $|C| \geq 3$, then, the inequality*

$$\sum_{v_{k,s} \in C} z_s^k \leq 1, \tag{18}$$

is valid for $P(G, K, \mathbb{S})$.

5.3 Odd-cycle-based Inequalities

In what follows we will describe valid inequalities induced by odd cycles in a conflict graphs H_I^E and H_S^E.

5.3.1 Interval-odd-cycle inequalities

Proposition 9 (Hadhbi 2022) *Let $I = [s_i, s_j]$ be an interval in $[1, \bar{s}]$, and H an odd-cycle in a conflict graph H_I^E. Then, the inequality*

$$\sum_{v_k \in H} \sum_{s=s_i+w_k-1}^{s_j} z_s^k \leq \frac{|H|-1}{2}, \tag{19}$$

is valid for $P(G, K, \mathbb{S})$.

5.3.2 Slot-assignment-odd-cycle inequalities

Proposition 10 (Hadhbi 2022) *Let H be an odd-cycle in a conflict graph H_S^E. Then, the inequality*

$$\sum_{v_{k,s} \in H} z_s^k \leq \frac{|H|-1}{2}, \tag{20}$$

is valid for $P(G, K, \mathbb{S})$.

6 Branch-and-cut-and-price algorithm

As mentioned before, the path formulation (1)–(10) has an exponential number of variables. We thus devise a Branch-and-Cut-and-Price algorithm to solve the problem. Recall that a Branch-and-Cut-and-Price algorithm consists in solving a sequence of linear programs using a column generation procedure at each node of a Branch-and-Cut tree. Moreover, when the solution obtained by the column generation procedure is fractional, we use, in a separation phase, the valid inequalities presented in Sect. 5 to strengthen the linear relaxation of the path formulation.
In what follows, we present our column generation procedure and the separation algorithms for the valid inequalities. First, we present a preprocessing algorithm whose aim is to compute and handle sets of forbidden edges and essential edges.

6.1 Preprocessing algorithm

Recall that, for a demand k, the set of forbidden edges, denoted by E_0^k, is composed by edges that cannot be used in a feasible (o_k, d_k)-path due to the transmission-reach

constraint. Also, the set of essential edges, denoted by E_1^k, is composed by edges which belong to all the feasible (o_k, d_k)-paths.

Our preprocessing algorithm starts by computing E_0^k. The idea is to choose an edge $e = ij$ and check if it belongs to E_0^k or not. For this, we first use Dijkstra's algorithm to compute a shortest path between the pair of nodes (o_k, i), (j, d_k), (o_k, j), and (i, d_k), denoted respectively by $P_{o_k,i}$, P_{j,d_k}, $P_{o_k,j}$ and P_{i,d_k}. Then, we consider the two (o_k, d_k)-paths $P_1 = (P_{o_k,i}, ij, P_{j,d_k})$ and $P_2 = (P_{o_k,j}, ij, P_{i,d_k})$. If the length of P_1 and P_2 is greater than $\bar{\ell}_k$, then edge e is a forbidden edge for demand k and hence is added to E_0^k.

Next, we compute the edge set E_1^k. First, we remove from G all the edges of E_0^k. We denote by G^k the remaining graph. This works as follows. For each demand $k \in K$, we give weight 1 to each edge of G^k. Then, we compute a minimum weight $(o_k - d_k)$-cut in G^k. If the weight of this cut equals to 1, then this cut is formed by a single edge e which is an essential edge for demand k. This edge is hence added to E_1^k and its weight becomes 2. We compute again a minimum weight $(o_k - d_k)$-cut and check the weight as before and stop when this weight is greater than or equal to 2. The minimum weight cut is computed using Goldberg and Tarjan's (1986) maximum flow algorithm.

Finally, for every demand k, we add the following constraints in the path formulation

$$x_e^k = 0, \text{ for every } e \in E_0^k,$$
$$x_e^k = 1, \text{ for every } e \in E_1^k.$$

As a consequence, the Branch-and-Cut-and-Price algorithm is now based on this new formulation.

6.2 The column generation algorithm

The column generation algorithm starts with a restricted linear program containing a small subset of variables which provides a feasible basis for the master problem. For this, as it will be detailed below (Sect. 6.2.2), we compute some initial columns which define a feasible solution for the master problem.

At each iteration, the algorithm checks if there exists a variable $y_{p,s}^k$ not in the current RMP having a negative reduced cost. If so, such a variable is added to the current RMP which is solved again. The procedure stops when all the path variables y have non negative reduced cost. The problem of finding variables having negative reduced cost is the so-called pricing problem.

6.2.1 The pricing problem

The pricing problem consists in finding a feasible path p for a demand k and a slot s such that variable $y_{p,s}^k$ has a negative reduced cost if one exists. Let $\alpha_k \in \mathbb{R}, k \in K$; $\beta^k \in \mathbb{R}, k \in K$; $\lambda_e^k \in \mathbb{R}, k \in K, e \in E$; $\rho_s^k \in \mathbb{R}, k \in K, s \in \mathbb{S}$; $\mu_s^e \leq 0, e \in E, s \in \mathbb{S}$, be the dual variables for constraints (2), (3), (4), (5), (6), respectively. The dual of the

linear relaxation (1)–(7) is

$$\max \sum_{e\in E}\sum_{s\in\mathbb{S}} \mu_s^e - \sum_{k\in K} \beta^k, \tag{21}$$

subject to

$$\beta^k - \sum_{e\in E(p)} (\lambda_e^k + \sum_{s'=s-w_k+1}^{s} \mu_{s'}^e) - \rho_s^k \geq 0, \forall k \in K, \forall p \in P^k, \forall s \in \{w_k, \ldots, \bar{s}\}, \tag{22}$$

$$\alpha^k - \sum_{e\in E(p)} \sum_{s'=\max(1,s-w_k+1)}^{s} \mu_{s'}^e \geq 0, \forall k \in K, \forall p \in P^k, \forall s \in \{1, \ldots, w_k - 1\}, \tag{23}$$

$$c_e + \lambda_e^k \geq 0, \forall k \in K, \forall e \in E, \tag{24}$$

$$\alpha^k + \rho_s^k \geq 0, \forall k \in K, \forall s \in \mathbb{S}, \tag{25}$$

$$\mu_s^e \leq 0, \forall e \in E, \forall s \in \mathbb{S}. \tag{26}$$

As a consequence, the so-called reduced-cost $rc_s^k(p)$, related to demand $k \in K$, path $p \in P^k$ and slot $s \in \{w_k, \ldots, \bar{s}\}$, is given by

$$rc_s^k(p) = \beta^k - \rho_s^k + \sum_{e\in E(p)} \left[-\lambda_e^k - \sum_{s'=s-w_k+1}^{s} \mu_{s'}^e \right]. \tag{27}$$

Therefore, for each demand $k \in K$ and slot $s \in \{w_k, \ldots, \bar{s}\}$, the pricing problem aims at finding a path p^* of P^k such that

$$rc_s^k(p^*) = \beta^k - \rho_s^k + \min_{p\in P^k} \left[\sum_{e\in E(p)} -\lambda_e^k - \sum_{s'=s-w_k+1}^{s} \mu_{s'}^e \right]. \tag{28}$$

Finding a such path p^* can be seen as a separation procedure for the dual constraints (22), which consists in identifying a path p^* for a demand $k \in K$ and a slot $s \in \{w_k, \ldots, \bar{s}\}$ such that $\beta^k - \rho_s^k + \sum_{e\in E(p)}[-\lambda_e^k - \sum_{s'=s-w_k+1}^{s} \mu_{s'}^e] < 0$ and $\sum_{e\in E(p^*)} \ell_e \leq \bar{\ell}_k$. Therefore, the pricing problem is equivalent to the Resource Constrained Shortest Path Problem (RCSP) which is well known to be weakly NP-Hard (Dror 1994). It consists in finding a RCP which does not belong to a set of paths called *forbidden-paths*. To solve this problem, we propose a pseudo-polynomial time dynamic programming based algorithm. It is an adaptation of the dynamic programming algorithm proposed by Dumitrescu and Boland (2001) for the RCSP problem

while taking into account some additional contraints related to the so-called forbidden-paths. This is based on the following procedure.
First, for each demand $k \in K$ and slot s, we consider a graph $G_s^k = (V, E \setminus E_0^k)$, and a set of forbidden-paths F^k. Each edge e in G_s^k is specified by a cost $\tilde{c}_e$ equal to $-\lambda_e^k - \sum_{s'=s-w_k+1}^{s} \mu_{s'}^e$, and a length $\tilde{\ell}_e$ equal to ℓ_e. The dynamic programming based algorithm consists in finding, for each demand k and slot s, the minimum-cost path $p*$ in G_s^k which does not belong to set F^k while satisfying the transmission-reach constraint. For this, we associate with each node v in G_s^k a set of labels L_v where each label corresponds to a sub-path from the origin o_k to node v in G_s^k. For each node $v \in V$, we use $T_v \subseteq L_v$ to denote the set of labels of node v which have been treated. That is the corresponding sub-paths have been used to generate a new label for each neighbor node $u \in V \setminus \{v\}$ with $uv \in E$ or $vu \in E$. Note that the edges of a forbidden path $p \in F^k$ can be used by the minimum-cost path $p*$ for demand k and slot s.
The complexity of the algorithm is bounded by $\mathcal{O}(|E| * \bar{\ell}_k)$ (Dumitrescu and Boland 2001) for each demand k.
Algorithm 1 summarizes the different steps of the dynamic programming algorithm.

.

6.2.2 The Initial Columns

We consider the set of precomputed paths B_k for each demand k and $\bar{s}$ obtained in Sect. 4. Using this, we introduce a light path variable $y_{p,s}^k$ for each demand k, path $p \in B_k$ and slot $s \in \{1, \ldots, \bar{s}\}$. By adding the variables x_e^k for each demand $k \in K$ and edge $e \in E$, and variables z_s^k for each demand $k \in K$ and slot $s \in \{1, \ldots, \bar{s}\}$, we obtain a feasible basis for the RMP.

6.3 Separation Procedures

In what follows, we present our separation procedures for the valid inequalities presented in Sect. 5.
As noticed in Sect. 5, our valid inequalities are based on three well known classes of inequalities: cover, clique and odd-cycle inequalities. The separation problem for cover inequalities was first introduced by Crowder et al. (1983) for the knapsack problem. The cover separation problem is NP-Hard (Klabjan et al. 1998; Nemhauser and Wolsey 1988). Nemhauser and Sigismondi (1992) have proposed a greedy algorithm to obtain an approximate solution for this problem. On the other hand, the separation problem for clique inequalities is well known to be NP-Hard (Karp 1972; Padberg 1973). The greedy algorithm proposed by Nemhauser and Sigismondi (1992) for the cover separation problem can be adapted to obtain an approximate solution for the clique separation problem. However, the separation problem for odd-cycle inequalities can be solved exactly in polynomial time as shown by Grötschel et al. (1988). These results are still valid for the valid inequalities presented in Sect. 5.
In the following, we adapt the greedy algorithm proposed by Nemhauser and Sigismondi (1992) for the separation of the cover-based inequalities (11) and (13). This algorithm will also be used for the separation of the clique-based inequalities (14),

Algorithm 1 Dynamic Programming Algorithm

Data: Graph $G = (V, E)$, a demand k, a slot s, a set of forbidden-paths F^k, and the optimal values of the duals variables $(\alpha^*, \beta^*, \lambda^*, \rho^*, \mu^*)$
. **Result**: The minimum-cost path $p*$ for demand k and slot s
Step 0: Initialization
Set $L_{o_k} = \{(0, 0)\}$ and $L_v = \emptyset$ for each node $v \in V \setminus \{o_k\}$;
Set $T_v = \emptyset$ for each node $v \in V$;
Set STOP=FALSE;
Set $p^* = NULL$;
while *STOP==FALSE* **do**
  if $\cup_{v \in V}(L_v \setminus T_v) = \emptyset$ **then**
    Set STOP= TRUE;
    Set $p^* = \emptyset$;
    We select one label p from the labels L_{d_k} of destination node d_k having

$$\beta^k - \rho_s^k + \sum_{e \in E(p)} [-\lambda_e^k - \sum_{s'=s-w_k+1}^{s} \mu_{s'}^e] < 0, \text{ and } p \notin F^k.$$

    if *such label exists* **then**
      Set $p^* = p$;
    end
  end
  if $\cup_{v \in V}(L_v \setminus T_v) \neq \emptyset$ **then**
    Select a node $i \in V$ and a label $p \in L_i \setminus T_i$ having the smallest value of $\sum_{e \in E(p)} l_e$;
    for *each* $e = ij \in \delta(i)$ *such that* $\sum_{e' \in E(p)} l_{e'} + l_e \leq \bar{l}_k$ **do**
      if $j \notin V(p)$ *and* $p \cup \{e\} \notin F^k$ **then**
        Set $p' = p \cup \{e\}$;
        Update the set of label $L_j = L_i \cup \{p'\}$;
      end
    end
    Set $T_i = T_i \cup \{p\}$;
  end
end
return the minimum-cost path p^* for demand k and slot s;

(16), (17), (18). The polynomial algorithm proposed by Grötschel et al. in Grötschel et al. (1988) for the separation of odd-cycle-based inequalities will be also adapted for the separation of odd-cycle inequalities (19) and (20).

6.3.1 Separation of edge-interval-capacity-cover inequalities

We discuss in this section the separation problem of inequalities (11). Consider a fractional solution $(\bar{y}, \bar{z}, \bar{x})$, an edge $e \in E$ and an interval of contiguous slots $I = [s_i, s_j]$ in $\mathbb{S}$. The separation problem associated with inequality (11) consists in finding

a minimal cover $\tilde{K}^*$ for the interval I over the edge e, such that

$$\sum_{k\in\tilde{K}^*} \bar{x}_e^k + \sum_{s'=s_i+w_k-1}^{s_j} \bar{z}_{s'}^k > 2|\tilde{K}^*| - 1.$$

For this, we first select a demand $k \in K$ having the largest number of requested slots w_k with $\bar{x}_e^k > 0$ and $\sum_{s'=s_i+w_k-1}^{s_j} \bar{z}_{s'}^k > 0$. Then set $\tilde{K}^* = \{k\}$. After that, we iteratively add each demand $k' \in K \setminus \tilde{K}^*$ to $\tilde{K}^*$ if $\bar{x}_e^{k'} > 0$ and $\sum_{s'=s_i+w_{k'}-1}^{s_j} \bar{z}_{s'}^{k'} > 0$ while $\sum_{k\in\tilde{K}^*} w_k \leq |I|$, i.e., until $\tilde{K}^*$ becomes a cover for the interval I over edge e, that is $\sum_{k\in\tilde{K}^*} w_k > |I|$. We then derive a minimal cover from $\tilde{K}^*$ by deleting each demand $k \in \tilde{K}^*$ if $\sum_{k'\in\tilde{K}^*\setminus\{k\}} w_{k'} > |I|$. If inequality (11) induced by $\tilde{K}^*$ is violated, then it is added to the current LP.

6.3.2 Separation of edge-capacity-cover inequalities

Next, we discuss the separation of inequalities (13). Consider a fractional solution $(\bar{y}, \bar{z}, \bar{x})$, and an edge $e \in E$. The separation problem aims at identifying a minimal cover $\tilde{K}^*$ for the edge e, such that

$$\sum_{k\in\tilde{K}^*} \bar{x}_e^k > |\tilde{K}^*| - 1.$$

First, we select a demand $k \in K \setminus K_e$ having the largest number of requested slot w_k with $\bar{x}_e^k > 0$, and set $\tilde{K}^* = \{k\}$. Then, we iteratively add each demand $k' \in K\setminus(K_e \cup \tilde{K}^*)$ to $\tilde{K}^*$ if $x_e^{k'} > 0$ while $\sum_{k\in\tilde{K}^*} w_k \leq \bar{s} - \sum_{\tilde{k}\in K_e} w_{\tilde{k}}$ until $\tilde{K}^*$ becomes a cover for edge e. After that, we derive a minimal cover from $\tilde{K}^*$ by deleting each demand $k \in \tilde{K}^*$ if $\sum_{k'\in\tilde{K}^*\setminus\{k\}} w_{k'} > \bar{s} - \sum_{\tilde{k}\in K_e} w_{\tilde{k}}$. We then add inequality (13) induced by $\tilde{K}^*$ to the current LP if it is violated.

6.3.3 Separation of edge-interval-clique inequalities

Consider a fractional solution $(\bar{y}, \bar{z}, \bar{x})$. The separation problem of inequalities (14) consists in identifying a maximal clique C^* in a conflict graph $\tilde{H}_I^e$ for a given edge e and an interval of contiguous slots $I = [s_i, s_j]$ such that

$$\sum_{k\in C^*} \bar{x}_e^k + \sum_{s'=s_i+w_k-1}^{s_j} \bar{z}_{s'}^k > |C^*| + 1,$$

For this we need to identify a maximal clique in a conflict graph $\tilde{H}_I^e$. This works as follows. We first associate a positive weight for each node v_k in $\tilde{H}_I^e$ which equal to $\bar{x}_e^k * \sum_{s'=s_i+w_k-1}^{s_j} \bar{z}_{s'}^k$. Then we select a demand $k \in K$ having the largest weight $\bar{x}_e^k * \sum_{s'=s_i+w_k-1}^{s_j} \bar{z}_{s'}^k$. After that, we iteratively add each demand k' having $\bar{x}_e^{k'} > 0$ and

$\sum_{s'=s_i+w_{k'}-1}^{s_j} \bar{z}_{s'}^{k'} > 0$ if its corresponding node $v_{k'}$ is linked with all corresponding nodes of demands that are already added in C^*. At the end, C^* defines a clique in a conflict graph $\tilde{H}_I^e$. If inequality (14) induced by C^* is violated, then it is added to the current LP.

6.3.4 Separation of interval-clique inequalities

Here we discuss the separation of inequalities (16). Consider an interval of contiguous slots $I = [s_i, s_j]$. The separation problem for inequality (16) consists in identifying a maximal clique C^* in a conflict graph $\tilde{H}_I^E$ such that

$$\sum_{k \in C^*} \sum_{s'=s_i+w_k-1}^{s_j} \bar{z}_{s'}^{k} > 1.$$

For this, we use also the greedy algorithm introduced by Nemhauser and Sigismondi (1992) to identify a maximal clique in a conflict graph $\tilde{H}_I^E$. This works as follows. We first associate a positive weight $\sum_{s'=s_i+w_k-1}^{s_j} \bar{z}_{s'}^{k}$ for each node v_k in $\tilde{H}_I^E$. We then select a demand $k \in K$ having the largest weight $\sum_{s'=s_i+w_k-1}^{s_j} \bar{z}_{s'}^{k}$, and set $C^* = \{k\}$. After that, we iteratively add each demand k' having $\sum_{s'=s_i+w_{k'}-1}^{s_j} \bar{z}_{s'}^{k'} > 0$ if its corresponding node $v_{k'}$ is linked with each node v_k in $\tilde{H}_I^E$ with $k \in C^*$. At the end, we add inequality (16) induced by the maximal clique C^* to the current LP if it is violated.

6.3.5 Separation of interval-odd-cycle inequalities

Let's now discuss the separation problem of inequalities (19). It consists in identifying an odd-cycle H^* in a conflict graph $\tilde{H}_I^E$ for an interval I of contiguous slots and a fractional solution $(\bar{y}, \bar{z}, \bar{x})$ such that

$$\sum_{k \in H^*} \sum_{s'=s_i+w_k-1}^{s_j} \bar{z}_{s'}^{k} > \frac{|H^*| - 1}{2},$$

if there is some. The algorithm aims at finding a minimum weighted odd-cycle in an auxiliary graph with non-negative weight. For this, we first construct an auxiliary graph $\tilde{H}_I'^E$ from $\tilde{H}_I^E$ which defines a bipartite graph by creating a copy v_k' for each node v_k in $\tilde{H}_I^E$. Two nodes are linked in $\tilde{H}_I'^E$ if their original nodes are linked in $\tilde{H}_I^E$. We assign to each link (v_a, v_b) in $\tilde{H}_I^E$ a non-negative weight which equal to $\frac{1-\sum_{s'=s_i+w_a-1}^{s_j} \bar{z}_{s'}^{a} - \sum_{s'=s_i+w_b-1}^{s_j} \bar{z}_{s'}^{b}}{2}$. We then compute for each node v_k in $\tilde{H}_I^E$, the shortest path between v_k and its copy v_k' in the auxiliary conflict graph $\tilde{H}_I'^E$ denoted by $p_{v_k, v_k'}$. After that, we check if the total sum of weights over edges in $p_{v_k, v_k'}$ is smallest than

$\frac{1}{2}$, that is

$$\sum_{(v_a,v_b)\in E(p_{v_k,v'_k})} \frac{1-\sum_{s'=s_i+w_a-1}^{s_j}\bar{z}^a_{s'}-\sum_{s'=s_i+w_b-1}^{s_j}\bar{z}^b_{s'}}{2} < \frac{1}{2}.$$

If so, the odd-cycle H^* is composed of all the original nodes of nodes belonging to the computed shortest path p_{v_k,v'_k}. We then add inequality (19) induced by H^* to the current LP.

6.3.6 Separation of edge-slot-assignment-clique inequalities

Consider an edge $e \in E$. The separation problem for inequalities (17) consists in identifying a maximal clique C^* in a conflict graph $\tilde{H}^e_S$ such that

$$\sum_{v_{k,s}\in C^*} \bar{x}^k_e + \bar{z}^k_s > |C| + 1.$$

For this, we adapt a greedy algorithm introduced by Nemhauser and Sigismondi (1992) to identify a maximal clique C^* in a conflict graph $\tilde{H}^e_S$. Based on this, we first assign a positive weight $\bar{z}^k_s * \bar{x}^k_e$ to each node $v_{k,s}$ in $\tilde{H}^e_S$. We then select a node $v_{k,s}$ having the largest weight. Then set $C^* = \{v_{k,s}\}$. Next, we iteratively add to C^* each node $v_{k',s'}$ having $\bar{z}^{k'}_{s'} > 0$ and $\bar{x}^{k'}_e > 0$ if it is linked with all the nodes $v_{k,s}$ already added in C^*. If inequality (17) induced by C^* is violated, then it should be added to the current LP.

6.3.7 Separation of slot-assignment-clique inequalities

Now, we describe the separation problem for inequalities (18). It consists in identifying a maximal clique C^* in a conflict graph $\tilde{H}^E_S$ such that

$$\sum_{v_{k,s}\in C^*} \bar{z}^k_s > 1.$$

This works as follows. First, we assign a positive weight $\bar{z}^k_s$ to each node $v_{k,s}$ in a conflict graph $\tilde{H}^E_S$. We then select a node $v_{k,s}$ in $\tilde{H}^E_S$ having the largest weight, and set $C^* = \{v_{k,s}\}$. After that, we iteratively add each node $v_{k',s'}$ to the current C^* if it is linked with each node $v_{k,s} \in C^*$ and $\bar{z}^{k'}_{s'} > 0$. Then we add inequality (18) induced by C^* to the current LP if it is violated.

6.3.8 Separation of slot-assignment-odd-cycle inequalities

Finally, let's discuss the separation problem of inequalities (20). The separation problem consists in identifying an odd-cycle H^* in a conflict graph $\tilde{H}_S^E$ such that

$$\sum_{v_{k,s} \in H^*} \bar{z}_s^k > \frac{|H^*| - 1}{2}.$$

This can be done in polynomial time by finding a minimum weighted odd-cycle in an auxiliary graph formed from $\tilde{H}_S^E$. To do so, we first construct an auxiliary conflict graph $\tilde{H}_S'^E$ by duplicating each node $v_{k,s}$ in $\tilde{H}_S'^E$ (i.e., $v_{k,s}$ and its copy $v'_{k,s}$). Each two nodes form an edge in $\tilde{H}_S'^E$ if their original nodes are linked in $\tilde{H}_S^E$. Then we assign to each link $(\tilde{v}_{k,s}, \tilde{v}_{k',s'})$ a non-negative weight which equal to $\frac{1-\bar{z}_s^k-\bar{z}_{s'}^{k'}}{2}$. We then compute for each node $v_{k,s}$ in $\tilde{H}_S^E$, the shortest path $p_{v_{k,s},v'_{k,s}}$ between $v_{k,s}$ and its copy in $\tilde{H}_S'^E$. After that, we check if the total sum of weight over edges belonging to this path is smaller than $\frac{1}{2}$, that is

$$\sum_{(v_{a,s'},v_{b,s''}) \in E(p_{v_{k,s},v'_{k,s}})} \frac{1 - \bar{z}_{s'}^a - \bar{z}_{s''}^b}{2} < \frac{1}{2}.$$

If so, the odd-cycle H^* is composed of all the original nodes of nodes belonging to the computed shortest path $p_{v_{k,s},v'_{k,s}}$. As a consequence, inequality (20) induced by H^* is added to the current LP.

6.4 Overview of the branch-and-cut-and-price algorithm

In this section, we summarize the main steps of the Branch-and-Cut-and-Price algorithm.

The first step of the algorithm is to apply the preprocessing algorithm described above to find the sets of forbidden and essential edges. Recall that at the end of the preprocessing phase, the path formulation is updated with additional constraints related to forbidden and essential edges.

As mentioned before, we use the column generation algorithm to solve a sequence of linear programs at each node of the Branch-and-Bound tree. In addition, if the obtained solution is fractional we try to find one or more valid inequalities that are violated by that solution using the separation procedure presented in Sect. 6.3. Note that the structure of the pricing problem remains the same given that all the valid inequalities added to the problem are not based on the path variables $y_{p,s}^k$.

We repeat this procedure at each iteration of the algorithm until no new columns or violated inequalities are found. The final solution is optimal for the linear relaxation of the path formulation. Furthermore, if it is integral, then it is optimal for the problem. Otherwise, we create two subproblems by branching on a fractional $y_{p,s}^k$ variable for

a demand $k \in K$, a path $p \in P^k$ and a slot $s \in \{w_k, \ldots, \bar{s}\}$ with $0 < \bar{y}^k_{p,s} < 1$. Recall that variables x^k_e and z^k_s are binary as soon as variables $y^k_{p,s}$ are binary. Note that the pricing can be affected by the branching rule. For this, we don't need to generate an additional light path for a demand k for which $y^k_{p,s}$ is fixed to 1 for some path p and slot s. On the other hand, for each light path variable $y^k_{p,s}$ fixed to 0, we should not regenerate again the corresponding path p for demand k and slot s. However, the edges of path p can be used by another path p' for the same demand k and slot s. This has been taken into account in our dynamic programming algorithm.

7 Computational results

Our B&C&P algorithm has been implemented in C++ under Linux using the SCIP 6.0.2 framework (Gamrath et al. 2020) and CPLEX 12.9 (Cplex 2020) as LP solver. The program has been run on HPC with 64 Gb of memory and 8 threads running in parallel. The maximum CPU time has been set up to 5 h (18,000 s).
For the test instances, we use two types of graphs. The first ones are called "Real" as they correspond to real 5G network topology. The second ones, called "Realistic", are graphs from SND-Lib (Orlowski et al. 2007) having up to 166 nodes and 166 edges. Table 1 below gives the details for each graph.
For each instance, the demand set K is randomly generated with $|K| \in \{10, 20, 30, 40, 50, 100, 150\}$.
Our aim in this section is to show the efficiency of our B&C&P algorithm for solving the C-RSA, and also the impact of the valid inequalities.
Table 2 below reports the results we have obtained. For each instance we give

- $|K|$: the number of demands,
- $|\mathbb{S}|$: the number of slots,
- Nodes: the number of nodes in the Branch-and-Bound tree,
- Gap: the optimality gap,
- Cols: the number of columns added in the column generation phase,
- Cuts: the number of inequalities added in the constraints separation phase,
- TT: the total CPU time.

For each triplet $(G, |K|, |\mathbb{S}|)$ we generate four instances. Note that for our purpose, we have deactivated SCIP internal cut generation procedures. Based on some preliminary results, the separation of our valid inequalities is performed along a complete B&C&P in the following order:

- Edge-Capacity-Cover inequalities (13),
- Edge-Interval-Capacity-Cover inequalities (11),
- Edge-Slot-Assignment-Clique inequalities (17),
- Edge-Interval-Clique inequalities (14),
- Slot-Assignment-Clique inequalities (18).

In what follows, we report in Table 2 the results obtained by the complete B&C&P algorithm (c.f. columns B&C&P in Table 2) compared with the B&P algorithm.

Table 1 Characteristics of different topologies used for the experiments

Graphs		Number of nodes	Number of links	Max node degree	Min node degree	Average node degree
Real topology	German	17	25	5	2	2.94
	Nsfnet	14	21	4	2	3
	Spain	30	56	6	2	3.73
	Coronet100	100	136	5	2	2.72
Realistic topology	Brain161	161	166	37	1	2.06
	India35	35	80	9	2	4.57
	Ta65	65	108	10	1	3.32
	Zib54	54	80	10	1	2.96

Each line of Table 2 corresponds to the average results over 4 instances. For this, we distinguish two cases when the average CPU time is smaller than the time limit of 18000 s and optimality gap greater than 0, 00:

- we exceed the memory limit for one of the 4 instances before the time limit is reached,
- or there exists at least one instance of the 4 instances which is solved to optimality in less than 18000 seconds. So the average gap is positive, and the average CPU time is less than 18000 seconds.

It can be seen from Table 2 that 37 instances over 56 (i.e.. 66% of the instances) are solved to optimality. Out of these 37 instances, 64% (i.e., 24 instances) are solved to optimality at the root of the B&B tree. Moreover, 44% (i.e., 25 instances) are solved to optimality in less than 1 min. For the remaining instances, 16 instances are solved with an average optimality gap less than 1% in more than 1h. There exist only 3 triplets (Spain,50,35), (Brain161,50,40) and (Brain161,100,80) for which we reach the time limit of 18000 seconds when using the complete B&C&P algorithm.

We have also analyzed the efficiency of the valid inequalities presented in this paper. For this, we have run the algorithm on the same test instances as before, but without using the valid inequalities. The Branch-and-Price (B&P) algorithm thus obtained is then compared to the B&C&P algorithm.

Table 2 shows the results obtained by the B&C&P vs the B&P algorithm. As it appears, the number of instances solved to optimality decreases when the inequalities are not used (25 against 37 instances). Also, the number of nodes in the Branch-and-Bound tree is significantly less when using the B&C&P algorithm (12 against 24 instances solved at the root of the Branch-and-Bound tree).

Finally, the average CPU time over all the instances increases when the valid inequalities are not used (6461, 75 sec against 4748, 58 sec). Also, 21 instances are solved to optimality by the B&P in less than 1 min (against 25). For the remaining instances, 23 instances are solved with an average optimality gap less than 1% in more than 1h. Fig. 2 below summarizes the comparison of the results between the B&C&P and B&P algorithms over five indicators.

These observations clearly show the effectiveness of the valid inequalities and the whole B&C&P algorithm to solve large-sized instances of the C-RSA problem. We can clearly see that the B&C&P algorithm outperforms the B&P for all the five indicators.

Now, we analyse the efficiency of each class of valid inequalities we have introduced in this paper using three real graphs and three realistic graphs from SND-Lib Orlowski et al. (2007). For this, we have solved the problem using only one class of valid inequalities at once. Then, we have compared the results against the B&P algorithm and the complete B&C&P algorithm. Here we compare the optimality gap for each instance and each class of valid inequalities. These results are given in Table 3. For each instance, Table 3 reports the optimality gap for:

- The B&P algorithm (column B&P),
- The B&C&P with only Edge-Interval-Capacity-Cover inequalities (11) (column Ineq_11),
- The B&C&P with only Edge-Capacity-Cover inequalities (13) (column Ineq_13),
- The B&C&P with only Edge-Interval-Clique inequalities (14) (column Ineq_14),

Table 2 Influence of the valid inequalities: B&P vs B&C&P

Instance			B&P_SCIP				B&C&P_SCIP				
G	$\|K\|$	$\|\mathbb{S}\|$	Nodes	Gap	Cols	TT	Nodes	Gap	Cols	Cuts	TT
German	10	15	28	**0.00**	13.50	0.88	1	**0.00**	3.25	6.25	0.07
	20	45	39	**0.00**	0	6.31	1	**0.00**	0	7.75	0.25
	30	45	1	**0.00**	0	0.20	1	**0.00**	0	0	0.31
	40	45	1489.67	0.33	324.67	6000.12	1557.67	0.13	309.67	339	5998.03
	50	55	3550.75	0.18	412.50	13506.57	1513	0.14	371	385	9020.19
	100	140	1	**0.00**	0	9.86	2	**0.00**	0	6.25	64.73
	150	210	34	**0.00**	0	417.78	51	**0.00**	0	24.75	932.25
Nsfnet	10	15	11	**0.00**	41.50	0.37	1	**0.00**	0	0	0.02
	20	20	190.50	**0.00**	168.50	34.67	1	**0.00**	165267.25	26.75	4487.61
	30	30	3286	1.17	295.25	9032.68	1	**0.00**	74970.75	7	8926.96
	40	35	4673.50	0.47	342	18000	1	**0.00**	0	16	3.72
	50	50	2218	0.54	566	13506.61	1	**0.00**	108442.50	18.25	8932.48
	100	120	2029	2.01	1849	18000	1	**0.00**	0	0	6.36
	150	160	321.50	11.66	1847.25	18000	1	**0.00**	0	0	32.48
Coronet100	10	320	1	**0.00**	867.50	51.36	1	**0.00**	867.50	0	54.11
	20	320	1	**0.00**	663.50	31.06	1	**0.00**	663.50	0	37.09
	30	40	1504	2.94	2577.50	18000	1545	1	11460.50	861.75	18000
	40	40	2969	0.46	1196	9008.08	12	**0.00**	57205.50	152	8983.29
	50	80	771.50	0.01	1282.75	4508.32	28.50	**0.00**	690.25	14.75	80.28
	100	120	333.50	3.69	1931	6836.74	130	4.62	1919	206.25	5182.07
	150	200	124	17.25	4272	14131.79	111.50	22.90	2629.50	296	16231.28

Table 2 continued

Instance			B&P_SCIP				B&C&P_SCIP				
G	$\|K\|$	$\|\mathbb{S}\|$	Nodes	Gap	Cols	TT	Nodes	Gap	Cols	Cuts	TT
Spain	10	15	5	**0.00**	79.75	0.58	1	**0.00**	78.75	2.50	0.19
	20	20	1645	**0.00**	871.25	3033.88	724.5	**0.00**	782.50	91	1860.90
	30	25	2422	0.77	1586.75	10943.42	3427	0.54	1896	471.75	12332.07
	40	30	1545	0.56	1205.50	9000.70	2339.5	0.61	1162.75	619	9001.09
	50	35	3117.5	0.79	3172.50	17996.72	2163	0.79	3241.25	700	18000
	100	120	977.5	0.15	4870.50	14439.73	389.5	0.08	4885.75	143.25	13897.07
	150	160	127.5	16.84	5562.75	15163.74	122.5	19.77	5539	282.25	17242.74
India35	10	40	2	**0.00**	0	0.56	1	**0.00**	0	8.50	0.28
	20	40	1	**0.00**	36	0.66	1	**0.00**	36	0	0.57
	30	40	71.50	**0.00**	109	49.93	9.50	**0.00**	34.50	43	9.51
	40	40	3975.50	0.38	2046.25	18000	2754.50	0.11	17896.50	737.50	13542.45
	50	80	1	**0.00**	69.50	3.87	1	**0.00**	69.50	24	6.37
	100	120	496	0.01	50.50	9072.59	353.50	**0.00**	98	356.25	4820.46
	150	200	292	0.01	96.50	9831.30	100.50	0.10	96.50	389.25	8155.83
Ta65	10	40	1	**0.00**	54.50	0.58	1	**0.00**	54.50	0	0.42
	20	40	1	**0.00**	188.50	1.41	1	**0.00**	188.50	2	1.21
	30	40	1109.50	0.08	539.75	4499.45	1383.50	0.08	509.50	125.50	4500
	40	40	529	0.03	562	4517.43	1.50	**0.00**	380.25	17.75	9.95
	50	40	1591.50	0.08	1131.50	9002.35	596	0.03	991.75	337.25	4540.26
	100	80	463	0.14	2179	6683.60	362	**0.00**	2148.25	248	5566.88
	150	160	123	2.29	3821.75	5870.91	147.50	**0.00**	3743.25	232.25	5225.13

Table 2 continued

Instance			B&P_SCIP				B&C&P_SCIP				
G	$\|K\|$	$\|\mathbb{S}\|$	Nodes	Gap	Cols	TT	Nodes	Gap	Cols	Cuts	TT
Brain161	10	40	1	**0.00**	0	0.35	1	**0.00**	0	0	0.17
	20	40	1	**0.00**	0	0.72	1	**0.00**	0	0	0.33
	30	40	8	**0.00**	0	8.51	1	**0.00**	0	1	0.62
	40	40	14	**0.00**	54.75	35.18	2.50	**0.00**	45.50	9.50	6.77
	50	40	5370	0.50	264.33	18000	10326.33	0.40	367	2087.67	18000
	100	80	1318	0.81	314	18000	2234.50	0.15	420.25	654.25	18000
	150	160	113.50	**0.00**	0	2257.49	85.50	**0.00**	0	30.25	1645.30
Zib54	10	40	1	**0.00**	27	0.40	1	**0.00**	27	0	0.27
	20	40	1	**0.00**	137.50	1.23	1	**0.00**	137.50	0	0.70
	30	40	670.5	0.09	283.50	4501.18	3	**0.00**	132.25	10	5.35
	40	40	99	**0.00**	378.75	102.17	1	**0.00**	275.75	18.75	3.15
	50	40	864.5	0.42	761.50	9063.67	840.5	0.13	649	280.75	9015.80
	100	80	1104	0.18	1600.50	16751.83	508	0.03	1787.50	343.50	8004.65
	150	160	101.5	2.28	2911.50	5938.67	125	0.26	2860.75	156.75	5550.35

0.00 written in bold means that the instance is solved to optimality

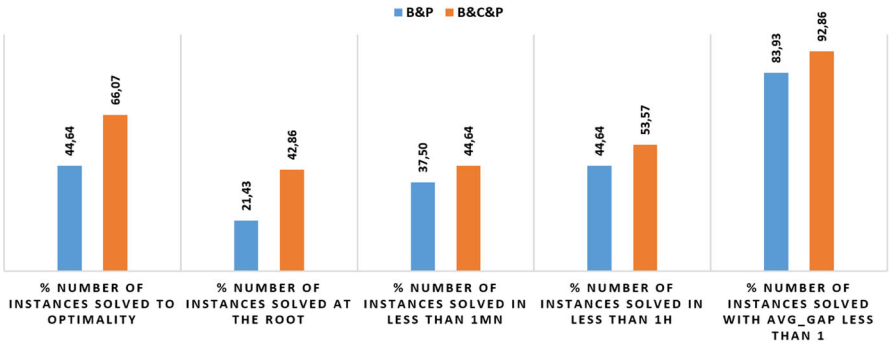


Fig. 2 Comparison of the results: B&P vs B&C&P

- The B&C&P with only Interval-Clique inequalities (16) (column Ineq_16),
- The B&C&P with only Edge-Slot-Assignment-Clique inequalities (17) (column Ineq_17),
- The B&C&P with only Slot-Assignment-Clique inequalities (18) (column Ineq_18),
- The B&C&P with only Interval-Odd-Cycle inequalities (19) (column Ineq_19),
- The B&C&P with only Slot-Assignment-Odd-Cycle inequalities (20) (column Ineq_20),
- The full B&C&P algorithm (column B&C&P).

The results show that the complete B&C&P algorithm is able to solve more instances to optimality compared with when using only one class of valid inequalities. Moreover, when the optimality is not guaranteed, the optimality gap achieved by the complete B&C&P algorithm, in general, is better than when using only one class of valid inequalities.

We have also noticed that for some instances (eg. lines (Coronet100,150,200), (India35,150,200), (Ta65, 50, 40)), the algorithm with only Edge-Capacity-Cover inequalities (13) provides a better optimality gap than the complete B&C&P algorithm.

We compare also the number of nodes in the branching tree for each instance and each class of valid inequalities. These results are given in Table 4. Observe that fewer nodes are generated in the branching tree when using a complete B&C&P compared with the case when only one class of valid inequalities is used. The results show also that the number of nodes generated in the branching tree is smaller when using the Edge-Capacity-Cover inequalities (13) compared with the one obtained for the other classes of valid inequalities.

Next, we will compare the complete B&C&P algorithm with an existing Branch-and-Cut (B&C) algorithm (Diarrassouba and Hadhbi 2022) using the same instances and computational environment. This comparison is shown in Table 5.

The results show that the complete B&C&P algorithm solves several instances to optimality that are not solved to optimality when using the B&C algorithm (66 % vs 43 %). Also, we notice that 43 % of instances are solved to optimality by the B&C&P algorithm at the root node of the branching tree, while 82% of instances require branching when using the B&C algorithm. Furthermore, the number of nodes

Table 3 Influence of each class of valid inequalities based on the optimality gap

Instances			Optimality Gaps									
G	$\|K\|$	$\|\mathbb{S}\|$	B&P	Ineq_11	Ineq_13	Ineq_14	Ineq_16	Ineq_17	Ineq_18	Ineq_19	Ineq_20	All_Ineq
German	10	15	**0,00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	20	45	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	30	45	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	40	45	0.33	0.28	0.12	0.32	0.28	0.28	0.28	0.33	0.28	0.13
	50	55	0.18	0.26	0.14	0.18	0.25	0.18	0.14	0.18	0.18	0.14
	100	140	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	150	210	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
Nsfnet	10	15	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	20	20	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	30	30	1.17	1.28	1.07	1.17	1.17	1.16	1.17	1.17	1.17	0.00
	40	35	0.47	0.79	0.79	0.82	0.71	0.64	0.71	0.47	0.74	0.00
	50	50	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.00
	100	120	2.01	1.98	1.76	1.94	1.90	2.54	2.13	2.01	4.01	0.00
	150	160	11.66	8.93	6.26	11.70	11.06	9.91	11.06	11.66	40.05	0.00
Coronet100	10	320	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	20	320	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	30	40	2.94	1.47	1	2.2	2.47	2.00	2.03	2.94	2.11	1
	40	40	0.46	0.83	0.98	0.83	1.41	0.55	1.41	0.46	0.5	0.00
	50	80	0.01	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.01	**0.00**	0.00
	100	120	3.69	3.69	4.78	1.64	3.69	5.51	3.69	3.69	5.98	4.62
	150	200	17.25	22.89	21.15	28.67	28.68	24.23	9.93	17.25	37.35	22.90

Table 3 continued

Instances			Optimality Gaps									
G	$\|K\|$	$\|\mathbb{S}\|$	B&P	Ineq_11	Ineq_13	Ineq_14	Ineq_16	Ineq_17	Ineq_18	Ineq_19	Ineq_20	All_Ineq
India35	10	40	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	20	40	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	30	40	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	40	40	0.38	0.22	0.10	0.21	0.40	0.26	0.40	0.38	0.40	0.11
	50	80	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	100	120	0.01	0.01	**0.00**	0.01	0.01	0.01	0.01	0.01	0.01	0.00
	150	200	0.01	0.01	0.35	0.22	0.01	0.49	0.01	0.01	0.01	0.10
Ta65	10	40	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	20	40	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	30	40	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
	40	40	0.03	0.03	**0.00**	0.03	**0.00**	0.03	**0.00**	0.03	**0.00**	0.00
	50	40	0.08	0.07	**0.00**	0.06	0.08	0.06	0.07	0.08	0.07	0.03
	100	80	0.14	0.04	0.07	0.02	0.02	0.06	0.02	0.14	0.06	0.00
	150	160	2.29	2.29	2.37	0.59	2.37	**0.00**	0.08	2.29	1.34	0.00
Zib54	10	40	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	20	40	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	30	40	0.09	0.09	0.03	0.09	0.09	0.09	0.09	0.09	0.09	0.00
	40	40	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	50	40	0.42	0.24	0.06	0.24	0.24	0.25	0.24	0.42	0.24	0.13
	100	80	0.18	0.30	0.08	0.08	0.06	0.09	0.06	0.18	0.06	0.03
	150	160	2.28	2.28	2.28	0.78	0.46	1.11	2.28	2.28	0.68	0.26

0.00 written in bold means that the instance is solved to optimality

Table 4 Influence of each class of valid inequalities based on the number of nodes in the branching tree

Instances			Branching Tree Size									
G	$\|K\|$	$\|S\|$	B&P	Ineq_11	Ineq_13	Ineq_14	Ineq_16	Ineq_17	Ineq_18	Ineq_19	Ineq_20	All_Ineq
German	10	15	28	1.5	1	7	19	11.5	19	28	19	1
	20	45	39	21	1	38	39	21.5	39	39	53.5	1
	30	45	1	1	1	1	1	1	1	1	1	1
	40	45	1489.67	811	1378.5	879.5	863	653	1067.5	1489.67	782	1557.67
	50	55	3550.75	1863	2178.5	2349	1723	1909	3324.5	3550.75	2089	1513
	100	140	1	1	2.5	1	1	1	1	1	1	2
	150	210	34	32	30.5	31	32	32	32	34	46.5	51
Nsfnet	10	15	11	8	1	11	10	8.5	11	11	11	1
	20	20	145	158.5	97	107.5	114	96.5	110.5	145	113.5	1
	30	30	4373.67	2513	2529	3252.5	2875	2996	3125.5	4373.67	2398	1
	40	35	4817.67	4530.5	4324	6310	4954	4628.5	5084.5	4817.67	5188.5	1
	50	50	2218	2151.5	2384	2036	2231	2211.5	2347	2218	1801	1
	100	120	2029	1807.5	1580.5	1669.5	1917	1578.5	1828.75	2029	558.5	1
	150	160	321.50	349.5	482.5	423	334	375	323.5	321.50	54	1
Coronet100	10	320	1	1	1	1	1	1	1	1	1	1
	20	320	1	1	1	1	1	1	1	1	1	1
	30	40	1504	4166.33	3659	4000	2297.75	3637.5	2163	1504	4064.5	1545
	40	40	2969	629	2950.5	1146.5	2928	754	2979.5	2969	3114	12
	50	80	771.5	283.5	80	73	80	64	80	771.5	18	28.5
	100	120	333.5	261.5	255.5	200.5	287	179	307	333.5	189	130
	150	200	124	145	124.5	117.5	127.5	148	154	124	22.5	111.5

Table 4 continued

Instances			Branching Tree Size									
G	$\|K\|$	$\|S\|$	B&P	Ineq_11	Ineq_13	Ineq_14	Ineq_16	Ineq_17	Ineq_18	Ineq_19	Ineq_20	All_Ineq
India35	10	40	2	2	1	2	2	2.5	2	2	2	1
	20	40	1	1	1	1	1	1	1	1	1	1
	30	40	71.50	74	11.5	47	71.5	128.5	71.5	71.50	71.5	9.50
	40	40	3975.50	4277	2922	3907.67	4360.5	4266	4510.5	3975.50	4814	2754.50
	50	80	1	1	1	1	1	1	1	1	1	1
	100	120	496	461.5	249.5	767.5	574	412	370	496	413.5	353.50
	150	200	292	195.5	130.5	181.5	273	235	272	292	276.5	100.50
Ta65	10	40	1	1	1	1	1	1	1	1	1	1
	20	40	1	1	1	1	1	1	1	1	1	1
	30	40	1109.50	888	1088.5	1055	1005	800.5	1110	1109.50	1013	1383.50
	40	40	529	711	2	541	912.5	1076	912.5	529	912.5	1.50
	50	40	1591.50	1465.5	165.5	1637.5	1234	1566.5	1274.5	1591.50	823	596.00
	100	80	463	467.75	354.5	382	496.5	532	431.5	463	381.5	362.00
	150	160	123	124	183	194	200.5	210.5	238	123	92	147.50
Zib54	10	40	1	1	1	1	1	1	1	1	1	1
	20	40	1	1	1	1	1	1	1	1	1	1
	30	40	670.50	1076.5	3	1005	663	880	650	670.50	660	3
	40	40	99	36.5	1	732.5	99	747	99	99	94.5	1
	50	40	864.50	1868	527	1893	1778	1816	1678.5	864.50	1637.5	840.50
	100	80	1104	754.5	561	780	1194	759.5	1004	1104	793.5	508.00
	150	160	101.50	110	88	106.5	146.5	82	92.5	101.50	87	125.00

Table 5 Comparison between the B&C algorithm and the B&C&P algorithm

Instances			B&C_SCIP				B&C&P_SCIP				
G	$\|K\|$	$\|\mathbb{S}\|$	Nodes	Gap	Cuts	TT	Nodes	Gap	Cols	Cuts	TT
German	10	15	59	**0.00**	429.75	0.83	1	**0.00**	3.25	6.25	0.07
	20	45	141	**0,00**	2403.5	3.89	1	**0.00**	0	7.75	0.25
	30	45	160376.5	1.46	129867.25	8334.95	1	**0.00**	0	0	0.31
	40	45	383058.66	3.70	224642.33	16624.87	1557.67	0.13	309.67	339	5998.03
	50	55	251152.5	13.73	305309.75	17074.95	1513	0.14	371	385	9020.19
	100	140	3014.25	**0.00**	17668.5	617.8	2	**0.00**	0	6.25	64.73
	150	210	3609	**0.00**	24782.25	3057.79	51	**0.00**	0	24.75	932.25
Nsfnet	10	15	1	**0.00**	95.75	0.15	1	**0.00**	0	0	0.02
	20	20	21586	**0.00**	24587	192.27	1	**0.00**	165267.25	26.75	4487.61
	30	30	281569.66	3.29	340177	11048.71	1	**0.00**	74970.75	7	8926.96
	40	35	119841.66	1.17	163519.33	5673.46	1	**0.00**	0	16	3.72
	50	50	148476.5	5.91	340399.25	17405.09	1	**0.00**	108442.5	18.25	8932.48
	100	120	1	**0.00**	464.25	40.87	1	**0.00**	0	0	6.36
	150	160	1	**0.00**	496.25	136.02	1	**0.00**	0	0	32.48
Coronet100	10	320	1	**0.00**	679.75	462.15	1	**0.00**	867.5	0	54.11
	20	320	15	**0.00**	2550.5	832.22	1	**0.00**	663.5	0	37.09
	30	40	11304.25	6.08	219986	18000	1545	1	11460.5	861.75	18000
	40	40	2127	**0.00**	97838	707.54	12	**0.00**	57205.5	152	8983.29
	50	80	19.75	**0.00**	10159	139.55	28.5	**0.00**	690.25	14.75	80.28
	100	120	8390.25	7.66	72527	10920.7	130	4.62	1919	206.25	5182.07
	150	200	3165.75	29.13	60798.75	15527.1	111.5	22.90	2629.5	296	16231.28

Table 5 continued

Instances			B&C_SCIP				B&C&P_SCIP				
G	$\|K\|$	$\|\mathbb{S}\|$	Nodes	Gap	Cuts	TT	Nodes	Gap	Cols	Cuts	TT
Spain	10	15	37	**0.00**	1017.5	2.33	1	**0.00**	78.75	2.5	0.19
	20	20	277441	3.83	52289.33	13998.05	724.5	**0.00**	782.5	91	1860.9
	30	25	195940.25	40.16	137842	18000	3427	0.54	1896	471.75	12332.07
	40	30	96801	35.80	317990	18000	2339.5	0.61	1162.75	619	9001.09
	50	35	62458	56.42	331186	18000	2163	0.79	3241.25	700	18000
	100	120	17551.5	24.23	122482.25	18000	389.5	0.08	4885.75	143.25	13897.07
	150		5541	58.09	59099.25	18000	122.5	19.77	5539	282.25	17242.74
India35	10	40	1	**0.00**	779.75	1.8	1	**0.00**	0	8.5	0.28
	20	40	7	**0.00**	2437.75	5.92	1	**0.00**	36	0	0.57
	30	40	32156.75	**0.00**	10917.5	2309.66	9.5	**0.00**	34.5	43	9.51
	40	40	191812	0.18	27270	17333.53	2754.5	0.11	17896.5	737.5	13542.45
	50	80	69.25	**0.00**	11105.75	112.19	1	**0.00**	69.5	24	6.37
	100	120	23403.75	0.44	48702.75	9494.52	353.5	**0.00**	98	356.25	4820.46
	150	200	1026	**0.00**	19898	4101.8	100.5	0.10	96.5	389.25	8155.83
Ta65	10	40	1	**0.00**	373.25	5	1	**0.00**	54.5	0	0.42
	20	40	4.5	**0.00**	2391.25	12.53	1	**0.00**	188.5	2	1.21
	30	40	18749.25	0.08	14385.5	4408.51	1383.5	0.08	509.5	125.5	4500
	40	40	40794.5	0.64	48325.5	8858.97	1.5	**0.00**	380.25	17.75	9.95
	50	40	29624	16.33	107237	12064.88	596	0.03	991.75	337.25	4540.26
	100	80	10711	1.89	67741.5	9727.83	362	**0.00**	2148.25	248	5566.88
	150	160	4852	10.40	39939.75	7882.29	147.5	**0.00**	3743.25	232.25	5225.13

Table 5 continued

Instances			B&C_SCIP				B&C&P_SCIP				
G	$\|K\|$	$\|\mathbb{S}\|$	Nodes	Gap	Cuts	TT	Nodes	Gap	Cols	Cuts	TT
Brain161	10	40	1	**0.00**	117.25	2.08	1	**0.00**	0	0	0.17
	20	40	1	**0.00**	891.75	11.53	1	**0.00**	0	0	0.33
	30	40	23659.5	0.05	7614.25	4447.42	1	**0.00**	0	1	0.62
	40	40	37746.75	0.36	50575.5	13633.59	2.5	**0.00**	45.5	9.5	6.77
	50	40	27629.67	6.88	202920	17865.57	10326.33	0.40	367	2087.67	18000
	100	80	12583	7.09	95074	17938.60	2234.5	0.15	420.25	654.25	18000
	150	160	3721.75	3.91	29027.5	10696.73	85.5	**0.00**	0	30.25	1645.3
Zib54	10	40	1	**0.00**	309	2.46	1	**0.00**	27	0	0.27
	20	40	21.50	**0.00**	1642.25	8.39	1	**0.00**	137.50	0	0.70
	30	40	20554.75	0.55	79145.75	4431.22	3	**0.00**	132.25	10	5.35
	40	40	88619.25	0.10	24408.50	8776.35	1	**0.00**	275.75	18.75	3.15
	50	40	92605.50	2.23	188661.50	17658.98	840.50	0.13	649	280.75	9015.80
	100	80	22474.25	8.48	122186	17880.43	508	0.03	1787.50	343.50	8004.65
	150	160	6973.25	0.97	47.018.25	9809.37	125	0.26	2860.75	156.75	5550.35

0.00 written in bold means that the instance is solved to optimality

generated in the branching tree is in most cases very large when using the B&C algorithm (hundreds of nodes). We notice also that the CPU time is, in general, better for the B&C&P algorithm.
As a consequence, the B&C&P algorithm appears to be better than the B&C algorithm presented in Diarrassouba and Hadhbi (2022).

8 Conclusion

In this paper, we have addressed the C-RSA problem using a polyhedral approach and propose an exact resolution algorithm. We have first given an extended formulation for the problem using path variables. Then we have investigated the associated polytope, and introduced several classes of valid inequalities. Using these results, we have devised a Branch-and-Cut-and-Price algorithm to solve the C-RSA. The computational results show that the algorithm is efficient to solve optimality instances with different network topologies and large size demand sets. They show also that the valid inequalities we have introduced play an important role in improving the performance of the algorithm. We have also demonstrated the effectiveness of the B&C&P, which is shown to be better than any existing approach of the state-of-the-art.

Finally, it would be interesting to further address the Survivable Constrained-Routing and Spectrum Assignment (SC-RSA) problem. It first consists in determining for each traffic demand $k \in K$ two edge-disjoint (o_k, d_k)-paths in G namely primary path p_k and backup path p'_k with $\sum_{e \in E(p_k)} \ell_e \leq \bar{\ell}_k$ and $\sum_{e' \in E(p_{k'}) \ell_{e'}} \leq \bar{\ell}_k$. Moreover, an interval of contiguous frequency slots $S_k \subseteq \mathbb{S}$ (resp. $S'_k \subseteq \mathbb{S}$) of width equal to w_k is assigned to each demand k throughout its primary path p_k (resp. backup path p'_k) while satisfying the C-RSA constraints, and optimizing some network design objective functions. For this, we plan to investigate the polyhedron of the SC-RSA problem, and devise exact algorithms based on Branch-and-Cut and Branch-and-Cut-and-Price algorithms to solve the problem.

Funding The authors have not disclosed any funding.

Data Availability Enquiries about data availability should be directed to the authors.

Declarations

Competing interests The authors have not disclosed any competing interests.

References

Amar D (2016) Performance assessment and modeling of flexible optical networks. Thesis, Institut National des Télécommunications

Bertero F, Bianchetti M, Marenco J (2018) Integer programming models for the routing and spectrum allocation problem. In: Official Journal of the Spanish Society of Statistics and Operations Research, pp 465–488

Brun B, Baraketi S (2014) Routing and wavelength assignment in optical networks, pp 1–41. https://hal.archives-ouvertes.fr/hal-01062321
Cai A, Shen G, Peng L, Zukerman M (2013) Novel node-arc model and multiiteration heuristics for static routing and spectrum assignment in elastic optical networks. J Lightw Technol 3402–3413
Chen X, Guo J, Zhu Z, Proietti R, Castro A, Yoo SJB (2018) Deep-RMSA: a deep-reinforcement-learning routing, modulation and spectrum assignment agent for elastic optical networks. In: Optical fiber communications conference and exposition (OFC), pp 1–3
Chlamtac I, Ganz A, Karmi G (1992) Lightpath communications: an approach to high bandwidth optical WAN's. IEEE Trans Commun 1171–1182
Chouman H, Gravey A, Gravey P, Hadhbi Y, Kerivin H, Morvan M, Wagler A (2021) Impact of RSA optimization objectives on optical network state, pp 1–7. https://hal.uca.fr/hal-03155966
Chouman H, Luay A, Colares R, Gravey A Gravey P, Kerivin H, Morvan M, Wagler A (2021) Assessing the health of flexgrid optical networks, pp 1–7. https://hal.archives-ouvertes.fr/hal-03123302
Christodoulopoulos K, Manousakis K, Varvarigos E (2010) Offline routing and wavelength assignment in transparent wdm networks. In: Networking, IEEE/ACM transactions, pp 1557–1570
Christodoulopoulos K, Tomkos I, Varvarigos EA (2011) Elastic bandwidth allocation in flexible OFDM-based optical networks. In: Lightwave technology, pp 1354–1366
Colares R, Kerivin H, Wagler A (2021) An extended formulation for the Constraint Routing and Spectrum Assignment Problem in Elastic Optical Networks, pp 1–16. https://hal.uca.fr/hal-03156189
Cplex II (2020) V12. 9: User's manual for CPLEX. Int Bus Mach Corp 46(53):157
Crowder H, Johnson EL, Padberg MW (1983) Solving large scale zero-one linear programming problems. Oper Res 31:803–834
Diarrassouba I, Hadhbi Y (2022) The constrained-routing and spectrum assignment problem: valid inequalities and branch-and-cut algorithm. In: Ljubić I, Barahona F, Dey SS, Mahjoub AR (eds) Combinatorial optimization, ISCO, Lecture Notes in Computer Science, vol 13526
Dijkstra EW (1959) A note on two problems in connexion with graphs. Numer Math 1(1):269–271
Ding Z, Xu Z, Zeng X, Ma T, Yang F (2014) Hybrid routing and spectrum assignment algorithms based on distance-adaptation combined coevolution and heuristics in elastic optical networks. J Opt Eng 1–10
Dror M (1994) Note on the complexity of the shortest path models for column generation in VRPTW. J Oper Res 977–978
Dumitrescu I, Boland N (2001) Algorithms for the weight constrained shortest path problem. Int Trans Oper Res 15–29
Gamrath G, Anderson D, Bestuzheva K, Chen WK, Eifler L, Gasse M, Gemander P, Gleixner A, Gottwald L, Halbig K, Hendel G, Hojny C, Koch T, Bodic L, Maher PJ, Matter F, Miltenberger M, Mühmer E, Müller B, Pfetsch ME, Schlösser F, Serrano F, Shinano Y, Tawfik C, Vigerske S, Wegscheider F, Weninger D, Witzig J (March 2020) The SCIP Optimization Suite 7.0. http://www.optimization-online.org/DB_HTML/2020/03/7705.html
Goldberg AV, Tarjan RE (1986) A new approach to the maximum flow problem. In: Proceedings of the Eighteenth Annual Association for Computing Machinery Symposium on Theory of Computing, pp 136–146
Gong L, Zhou X, Lu W, Zhu Z (2012) A two-population based evolutionary approach for optimizing routing, modulation and spectrum assignments (RMSA) in O-OFDM networks. IEEE Commun Lett 1520–1523
Goscien R, Walkowiak K, Klinkowski M (2015) Tabu search algorithm, routing, modulation and spectrum allocation, anycast traffic, elastic optical networks. J Comput Netw 148–165
Grötschel M, Lovász L, Schrijver A (1988) Stable sets in graphs. In: Geometric algorithms and combinatorial optimization. algorithms and combinatorics, vol 2. Springer, Berlin, Heidelberg
Gurobi Optimization LLC (2021) Gurobi optimizer reference manual. https://www.gurobi.com
Gu R, Yang Z, Ji Y (2020) Machine learning for intelligent optical networks: a comprehensive survey. J CoRR 1–42
Hadhbi Y (2022) The constrained-routing and spectrum assignment problem: polyhedral analysis and algorithms. PhD thesis at Clermont Auvergne University
Hadhbi Y, Kerivin H, Wagler A (2019) A novel integer linear programming model for routing and spectrum assignment in optical networks. In: Federated conference on computer science and information systems (FedCSIS) , pp 127–134
Hai DH, Hoang KM (2017) An efficient genetic algorithm approach for solving routing and spectrum assignment problem. J Recent Adv Signal Process 187–192

Hai DH, Morvan M, Gravey P (2017) Combining heuristic and exact approaches for solving the routing and spectrum assignment problem. J Iet Optoelectron 65–72
Hai DT, Morvan M, Gravey P (2017) Combining heuristic and exact approaches for solving the routing and spectrum assignment problem. IET Optoelectron 12:65–72
He S, Qiu Y, Xu J (2020) Invalid-resource-aware spectrum assignment for advanced-reservation traffic in elastic optical network. In: Sensors
Jaumard B, Daryalal M (2016) Scalable elastic optical path networking models. In: 18th international conference transparent optical networks (ICTON), pp 1–4
Jiang R, Feng M, Shen J (2017) An defragmentation scheme for extending the maximal unoccupied spectrum block in elastic optical networks. In: 16th international conference on optical communications and networks (ICOCN), pp 1–3
Karp RM (1972) Reducibility among combinatorial problems. In: Complexity of computer computations: proceedings of a symposium on the complexity of computer computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, pp 85–94
Klabjan D, Nemhauser GL, Tovey C (1998) The complexity of cover inequality separation. Oper Res Lett 35–40
Klinkowski M, Pedro J, Careglio D, Pioro M, Pires J, Monteiro P, Sole-Pareta J (2010) An overview of routing methods in optical burst switching networks. Opt Switch Network 41–53
Klinkowski M, Pioro M, Zotkiewicz M, Ruiz M, Velasco L (2014) Valid inequalities for the routing and spectrum allocation problem in elastic optical networks. In: 16th international conference on transparent optical networks (ICTON), pp 1–5
Klinkowski M, Pioro M, Zotkiewicz M, Ruiz M, Velasco L (2015) A simulated annealing heuristic for a branch and price-based routing and spectrum allocation algorithm in elastic optical networks. In: Intelligent data engineering and automated learning – IDEAL, Springer International Publishing, pp 290–299
Klinkowski M, Pióro M, Żotkiewicz M, Walkowiak K, Ruiz M, Velasco L (2015) Spectrum allocation problem in elastic optical networks - A branch-and-price approach. In: 17th international conference on transparent optical networks (ICTON), pp 1–5
Klinkowski M, Walkowiak K (2011) Routing and spectrum assignment in spectrum sliced elastic optical path network. IEEE Commun Lett 884–886
Lezama F, Martinez-Herrera AF, Castanon G, Del-Valle-Soto C, Sarmiento AM, Munoz de Cote A (2021) Solving routing and spectrum allocation problems in flexgrid optical networks using pre-computing strategies. J Photon Netw Commun 41:17–35
Liu Z, Rouskas GN (2013) Link selection algorithms for link-based ilps and applications to rwa in mesh networks. In: Optical network design and modeling (ONDM), 17th international conference. IEEE, pp 59–64
Liu L, Yin S, Zhang Z, Chu Y, Huang S (2019) A Monte Carlo based routing and spectrum assignment agent for elastic optical networks. In: Asia communications and photonics conference (ACP), pp 1–3
Lohani V, Sharma A, Singh YN (2019) Routing, modulation and spectrum assignment using an AI based algorithm. In: 11th international conference on communication systems & networks (COMSNETS), pp 266–271
Lopez V, Velasco L (2016) Elastic optical networks: architectures, technologies, and control. Springer Publishing Company, Incorporated
Mesquita LAJ, Assis K, Santos AF, Alencar M, Almeida RC (2018) A routing and spectrum assignment heuristic for elastic optical networks under incremental traffic. In: SBFoton international optics and photonics conference (SBFoton IOPC), pp 1–5
Nemhauser GL, Sigismondi G (1992) A strong cutting plane/branch-and-bound algorithm for node packing. J Oper Res Soc 443–457
Nemhauser GL, Wolsey LA (1988) Integer and combinatorial optimization. Wiley, Hoboken
Nguyen DM, Ngoc LA, Huong PTV, Son NH, Hai DT (2019) An efficient column generation approach for solving the routing and spectrum assignment problem in elastic optical networks. In: the 6th NAFOSTED conference on information and computer science (NICS), Hanoi, Vietnam, pp 130–135
Orlowski S, Pióro M, Tomaszewski A, Wessäly R (2007) SNDlib 1.0-survivable network design library. In: Proceedings of the 3rd international network optimization conference (INOC 2007), Spa, Belgium
Padberg MW (1973) On the facial structure of set packing polyhedra. Math Program 5:199–215

Pedro H, Fernandes da Silva, Kerivin H, Nant JP, Wagler A (2022) A framework for routing and spectrum assignment in optical networks, driven by combinatorial properties. In: The international network optimization conference (INOC), pp 1–6

Ramaswami R (2009) Optical networks: a practical perspective, 3rd edn. Morgan Kaufmann Publishers Inc

Ramaswami R, Sivarajan K, Sasaki G (1993) Multiwavelength lightwave networks for computer communication. In: IEEE communications magazine, pp 78–88

Rebennack S, Reinelt G, Pardalos PM (2012) A tutorial on branch and cut algorithms for the maximum stable set problem. J Int Trans Oper Res 161–199

Ruiz M, Pioro M, Zotkiewicz M, Klinkowski M, Velasco L (2013) Column generation algorithm for RSA problems in flexgrid optical networks. Photon Netw Commun 53–64

Ruiz M, Pioro M, Zotkiewicz M, Klinkowski M, Velasco L (2020) A hybrid meta-heuristic approach for optimization of routing and spectrum assignment in Elastic Optical Network (EON). J Enterprise Inform Syst 11–24

Salani M, Rottondi C, Tornatore M (2019) Routing and spectrum assignment integrating machine-learning-based QoT estimation in elastic optical networks. In: IEEE INFOCOM - IEEE Conference on Computer Communications, p 173846

Shirazipourazad S, Zhou C, Derakhshandeh Z, Sen A (2013) On routing and spectrum allocation in spectrum-sliced optical networks. In: Proceedings IEEE INFOCOM, pp 385–389

Shiva-Kumar M, Sreenivasa-Kumar P (2002) Static lightpath establishment in wdm networksnew ilp formulations and heuristic algorithms. Comput Commun 109–114

Simonis H (2011) Solving the static design routing and wavelength assignment problem. In: Recent advances in constraints. Springer, pp 59–75

Siregar H, Takagi H, Zhang Y (2003) Efficient routing and wavelength assignment in wavelength-routed optical networks. In: Proceedings of 7th Asia-Pacific Network Operations and Mgmt Symposium, pp 116–127

Skorin-Kapov N (2007) Routing and wavelength assignment in optical networks using bin packing based algorithms. Eur J Oper Res 1167–1179

Talebi S, Alam F , Katib I, Khamis M, Salama R, Rouskas GN (2014) Spectrum management techniques for elastic optical networks: a survey. In: Optical switching and networking, pp 34–48

Velasco L, Klinkowski M, Ruiz M, Comellas J (2012) Modeling the routing and spectrum allocation problem for flexgrid optical networks. Photon Network Commun 177–186

Zang H, Jue JP, Mukherjee B et al (2000) A review of routing and wavelength assignment approaches for wavelength-routed optical wdm networks. In: Optical networks magazine, pp 47–60

Zhang Y, Xin J, Li X, Huang S (2020) Overview on routing and resource allocation based machine learning in optical networks. J Opt Fiber Technol 1–21

Zotkiewicz M, Pioro M, Ruiz M, Klinkowski M, Velasco L (2013) Optimization models for flexgrid elastic optical networks. In: 15th international conference on transparent optical networks (ICTON), pp 1–4

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.