Spectrum allocation problem in elastic optical networks - a branch-and-price approach

Mirosław Klinkowski[1*], Michał Pióro[2], Mateusz Żotkiewicz[2], Krzysztof Walkowiak[3], Marc Ruiz[4], and Luis Velasco[4]

[1] *National Institute of Telecommunications, Warsaw, Poland*

[2] *Warsaw University of Technology, Warsaw, Poland*

[3] *Wrocław University of Technology, Wrocław, Poland*

[4] *Universitat Politécnica de Catalunya, Barcelona, Spain*

[*] *M.Klinkowski@itl.waw.pl*

ABSTRACT In the paper we present a branch and price approach to routing and spectrum allocation – a basic optimization problem in elastic optical networks. We formulate the problem as a mixed-integer program for which we develop a branch and price algorithm enhanced with such techniques as cutting planes for improving lower bounds for the optimal objective value, and a greedy and a simulated annealing heuristics for improving the upper bounds. All these elements are combined into an effective optimization procedure. Preliminary results show that the algorithm is able to produce optimal solutions and in a vast majority of the considered cases it performs better than a standard branch-and-bound method implemented in the CPLEX solver.

Keywords: Branch and price, elastic optical networks, mixed-integer programming, routing and spectrum allocation.

1. INTRODUCTION

The use of advanced transmission and modulation techniques, spectrum-selective switching technologies, and flexible frequency grids (flexgrids), will allow next-generation optical networks to be spectrally efficient and, in terms of optical bandwidth provisioning, scalable and elastic [1], [2]. A challenging problem in the design and operation of flexgrid elastic optical networks (EONs) is the problem of routing and spectrum allocation (RSA). RSA consists in establishing optical path (lightpath) connections, tailored to the actual width of the transmitted signal, for a set of end-to-end demands that compete for spectrum resources.

The RSA optimization problem is $\mathcal{NP}$-hard [3]. In the literature, mixed-integer programming (MIP) formulations [4], [5], metaheuristics [6], [7], [8], and heuristics [3], [9], have been proposed to solve it. Both metaheuristics and heuristics can produce locally optimal solutions, however, without guarantees for global optimality. On the contrary, MIP formulations can be solved to optimality. A common approach is to use a standard branch-and-bound (BB) method, which is implemented in MIP solvers, for instance, in CPLEX [10]. The resolution of MIP using BB can be still difficult and time-consuming due to the processing of a large set of integer variables.

In this paper, we focus on developing an optimization algorithm capable of producing optimal RSA solutions and, at the same time, competitive to CPLEX. We apply several optimization approaches that are combined and built into a branch-and-price (BP) framework. The algorithm components include problem relaxation and application of cuts, both techniques used with the aim to improve lower bounds, as well as a search for upper bound solutions by means of a hybrid greedy RSA and simulated annealing algorithm. Preliminary results obtained for a 12-node network show that the algorithm is able to meet our goal. To the best of our knowledge, the presented work is among the first that aims at solving RSA to optimality in an efficient way.

The remainder of this paper is organized as follows. In Section 2, we present an MIP formulation of the considered RSA problem. In Section 3, we describe our optimization algorithm. The algorithm is evaluated in Section 4 using the results of numerical experiments. Finally, in Section 5, we conclude this work.

2. MIP FORMULATION OF RSA

We formulate RSA as an MIP problem and using a link-lightpath (LL) modeling approach that was proposed in [11]. In LL, the spectrum assignment-related constraints are removed from the MIP by using a set of pre-computed lightpaths. At the same time, the LL constraints assure that for each demand a lightpath is selected from the pre-computed set and the selected lightpaths are not in conflict with each other, i.e., their spectra do not overlap on the network links. In this work, we aim at minimizing the spectrum width required to allocate a given set of demands. Such an optimization objective has been frequently used in previous works on RSA [3].

The considered EON network is represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of optical nodes and $\mathcal{E}$ is the set of fiber links. In each link $e \in \mathcal{E}$, the same bandwidth (i.e., optical frequency spectrum) is available and it is divided into a set $\mathcal{S} = \{s_1, s_2, \ldots, s_{|\mathcal{S}|}\}$ of frequency slices of a fixed width. The set of node-to-node (traffic) demands to be realized in the network is denoted by $\mathcal{D}$. The notation has been gathered in Table I.

TABLE I: Notation.

Sets and parameters			
$\mathcal{E}$	set of links	$\mathcal{S}$	set of all frequency slices, $\mathcal{S} = \{1, 2, ..., \lvert\mathcal{S}\rvert\}$
$\mathcal{D}$	set of demands	$\mathcal{L}(d)$	set of lightpaths allowable for demand d
$\mathcal{L}$	set of allowable lightpaths, $\mathcal{L} = \bigcup_{d\in\mathcal{D}}\mathcal{L}(d)$	$\mathcal{Q}(d, e, s)$	set of lightpaths of demand d routed through link e and slice s
Variables			
x_{dl}	binary, $x_{dl} = 1$ when demand d uses lightpath l; $x_{dl} = 0$ otherwise		
x_{es}	binary, $x_{es} = 1$ when slice s is allocated in link e; $x_{es} = 0$ otherwise		
x_s	binary, $x_s = 1$ when slice s is allocated in any network link; $x_s = 0$ otherwise		

In the MIP model a notion of a lightpath is used. A lightpath is understood as a pair (p, c), where p is a route and c is a channel. The route is a path through a network from a source node to a termination node of a demand ($p \subseteq \mathcal{E}$), while the channel is a set of contiguous slices assigned to the lightpath ($c \subseteq \mathcal{S}$). Note that channel c should be wide enough to carry the bit-rate of demand d, if it is supposed to satisfy this demand. Channel c is the same for each link belonging to the routing path, what is called the spectrum continuity (SC) constraint. It is assumed that sets of allowable lightpaths $\mathcal{L}(d)$ for each demand are given, thus the problem simplifies to selecting one of those lightpaths for each demand in such a way that there are no two demands that use the same slice on the same link. Let $\mathcal{L}$ be the set of all allowable lightpaths, i.e., $\mathcal{L} = \bigcup_{d\in\mathcal{D}}\mathcal{L}(d)$.

Each lightpath $l \in \mathcal{L}(d)$ is assigned a binary variable $x_{dl}, d \in \mathcal{D}, l \in \mathcal{L}(d)$, where $x_{dl} = 1$ indicates that lightpath l is actually used to realize the traffic (bit-rate) of demand d. Besides, a binary variable $x_{es}, e \in \mathcal{E}, s \in \mathcal{S}$, indicates if there is a used lightpath allocated on slice s of link e. Eventually, the use of slice s in the network is indicated by a binary variable $x_s, s \in \mathcal{S}$. The MIP formulation is the following:

$$\text{minimize } z = \sum_{s\in\mathcal{S}} x_s \tag{1a}$$

$$[\lambda_d] \quad \sum_{l\in\mathcal{L}(d)} x_{dl} = 1 \qquad d \in \mathcal{D} \tag{1b}$$

$$[\pi_{es} \geq 0] \quad \sum_{d\in\mathcal{D}} \sum_{l\in\mathcal{Q}(d,e,s)} x_{dl} = x_{es} \qquad e \in \mathcal{E}, s \in \mathcal{S} \tag{1c}$$

$$x_{es} \leq x_s \qquad e \in \mathcal{E}, s \in \mathcal{S} \tag{1d}$$

where $\mathcal{Q}(d, e, s)$ is the set of lightpaths of demand d routed through link e and slice s. Optimization objective (1a) minimizes the number of used slices in the network, which is obtained by summing up variables x_s. Constraint (1b) assures that each demand will use one and only one lightpath from a set of allowable lightpaths. Constraint (1c) assures that there are no collisions of the assigned resources, i.e., there are no two lightpaths in the network that use the same slice on the same link. Finally, constraint (1d) defines variables x_s that indicate whether slice s is used on any link.

In the following, the linear programming (LP) relaxation of (1) is called the master problem and the variables in brackets, i.e, λ_d and π_{es}, are its dual variables.

3. Optimization Algorithm

In this section, we develop an optimization algorithm for solving problem (1). The algorithm is based on a branch-and-price (BP) framework [12], which is a combination of branch and bound (BB) and column generation (CG) methods [13]. In a BB method, a tree of linear subproblems, called restricted master problems (RMPs), related to the master problem is generated through a branching process. In particular, at each BB node a subset of variables is bounded by means of extra constraints. For a minimization problem (such as problem (1)), the solution of each RMP provides a lower bound (LB) that is used either to discard certain BB nodes from the search for an optimal solution or to set an upper bound (UB), whenever this solution is also feasible for MIP. The BB search is terminated whenever there are no nodes left for processing.

Now, in BP each RMP is solved using a CG procedure. Namely, BP is initiated with a limited set of problem variables (columns) and at each node of the BB search tree, additional variables are generated and included into RMP. Since in large problems most columns are irrelevant for the problem (their corresponding variables equal zero in any optimal solution), therefore, the processing complexity can be decreased by excluding these columns from the formulation. Note that an unalterable (possibly complete) set of columns is included into each RMP in a standard BB method. Finally, to increase the effectiveness of the BB search in BP, we implement additional procedures that aim at improving lower and upper bounds. The details of the algorithm implementation are presented in the following subsections. Due to space limitations, we restrict the formal description to the necessary minimum.

3.1 Branch and price

Let z^{lb} and z^{ub} denote, respectively, a lower and an upper bound on the optimal solution that are estimated at a given BB node. Let z^{LB} be the lowest lower bound among all the nodes that are left for processing and z^{UB} be the best upper bound found. The master node of the BB tree is initiated with $z^{lb} := z^{LB} := 0$ and $z^{ub} := z^{UB} := \infty$. At each BB node, the following actions are performed:

1) If $z^{UB} \leq z^{lb}$ then discard the node from the search.
2) Find new value of z^{ub} using a heuristic (see Sec. 3.3). If $z^{ub} < z^{UB}$ then set $z^{UB} := z^{ub}$. If $z^{UB} \leq z^{lb}$ then discard the node.
3) Solve a relaxed problem (see Sec. 3.2). If the solution is greater than z^{lb} then update z^{lb}. If $z^{UB} \leq z^{lb}$ then discard the node.
4) Initialize RMP and solve it using CG (see Sec. 3.1.1). If the solution of RMP is integral then update z^{UB} and discard the node. Otherwise, perform branching on selected variables (see Sec. 3.1.2).

After either discarding or completing the node processing, a next node to be processed is selected (arbitrarily) among the nodes for which $z^{lb} = z^{LB}$ (first condition) and z^{ub} is minimal (second condition).

1) Column generation and cuts: The RMP is initiated with a set of allowable lightpaths $\mathcal{L}$ that either correspond to the heuristic solution (in the master node) or have been generated at the parent node (in the rest of nodes). Next, this set is extended with new lightpaths. A key element of CG is to formulate and solve a *pricing* problem, which concerns finding such a new column that, when included into RMP, leads to the improvement of the objective function value. For formulation (1), the pricing problem reduces to the search for lightpath $l = (p, c)$ for which its reduced cost, calculated as $\lambda_{d(l)} - \sum_{e \in p} \sum_{s \in c} \pi_{es}$, where $d(l)$ is the demand realized by l, is positive. At each iteration of CG, for each demand, we include into set $\mathcal{L}$ a lightpath with the largest positive reduced cost, if such lightpath exists. For details, refer to [14].

As discussed in Section 3.1.2, selected lightpaths in $\mathcal{L}$ may not be permitted in some BB nodes and hence their corresponding variables x_{dl} are set to 0 in the RMP. Still, these variables may be regenerated by CG. To mitigate this problem, we assume that a (large) set of candidate lightpaths is given, only lightpaths from this set are processed by CG, and a lightpath can be included into $\mathcal{L}$ only if it is permitted.

Note that z is integer in (1) and, therefore, $z \geq \lceil z^{lb} \rceil$ holds. Since z represents the number of used slices in the network and we optimize the width of used spectrum, therefore, at least $\lceil z^{lb} \rceil$ consecutively indexed variables x_s should be equal to 1 and the RMP can be enhanced with the following valid equalities (cuts): $x_s = 1$ for $s \in \{1, 2, \ldots, \lceil z^{lb} \rceil\}$.

2) Branching: In the branching step, two child nodes (denoted as Ω_0 and Ω_1) of the currently processed (parent) node are created. The columns generated at the parent node are passed to the child nodes. Also, the values of z^{lb} and z^{ub} of the child nodes are initiated with the corresponding values of the parent node.

Next, a subset of lightpaths (referred to as restricted) is selected and imposed to be either used or not permitted, respectively, in Ω_1 and Ω_0 as well as in their child nodes. We allow two kinds of branching, namely, for a selected demand we impose/prohibit either a) a routing path or b) a lightpath that it may use. We apply the first branching rule until all demands have assigned their routes and then we use the second rule. To select a branching demand/path/lightpath, we search for the largest fractional flow that is carried through such a link which has the highest number of both shared and underutilized slices in the current optimal solution to RMP.

3.2 Lower bounds

LBs of good quality can be obtained by relaxing the SC constraint and by solving the resulting MIP problem, which can be formulated as $z^{lb} = \min\left\{y : \sum_{p \in \mathcal{P}_d} x_{dp} = 1, d \in \mathcal{D} \text{ and } \sum_{d \in \mathcal{D}, p \in \mathcal{P}_d : p \ni e} x_{dp} n_d \leq y, e \in \mathcal{E}\right\}$. Here, $\mathcal{P}_d$ denotes the set of allowable routing paths for demand d, x_{dp} is a binary variable that indicates if path p is used to realize demand d, y is an integer variable that represents the number of slices required in the most utilized link, and n_d denotes the number of slices requested by demand d. In a BB node, those routing variables x_{dp} that correspond to restricted lightpaths are also restricted in the relaxed problem.

3.3 Upper bounds

In each BB node, we run a greedy RSA algorithm that processes demands one-by-one, according to a given demand order, and allocates them with the lowest possible slice index (primary goal) and on the shortest routing path (secondary goal). The demand order is being optimized by applying a simulated annealing (SA) algorithm, in a similar way as in [7]. Again, the heuristic takes into account the restrictions imposed on routing paths and lightpaths in BB nodes. The obtained solutions provide UBs on the solution of problem (1).

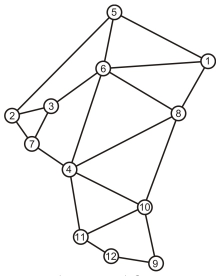

Figure 1: DT12 network.

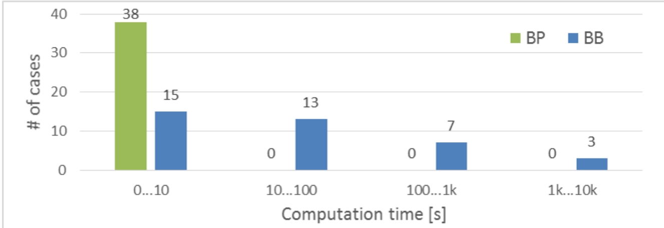

Figure 2: A histogram of computation times (T) for the problem instances that are solved in the master node of BP.

4. Numerical Results

In this section, we evaluate the BP algorithm. The evaluation is performed for a generic German network of 12 nodes and 20 links, denoted as DT12 and presented in Figure 1. We assume the flexgrid of 12.5 GHz granularity and the spectral efficiency of 2 bit/s/Hz. We consider symmetric demands with randomly generated end nodes and uniformly distributed bit-rate requests between 10 and 400 Gbit/s. The number of demands is $|\mathcal{D}| \in \{10, 20, ..., 60\}$ and for each $|\mathcal{D}|$ we evaluate 10 demand sets, i.e., 60 traffic instances in total. The number of candidate routing paths is $|\mathcal{P}_d| = 10$ and the set of candidate lightpaths consists of all possible lightpaths established on these paths and allocating any, appropriate for given demand, segment of spectrum on the flexgrid.

As a reference algorithm we use an MIP solver CPLEX v.12.5.1 (denoted as BB), which is run with its default settings (all types of cuts and heuristics enabled) and in a parallel mode (8 threads). CPLEX is also used in BP, as an LP solver in the column generation phase and as an MIP solver in the search for lower bounds (see Sec. 3.2). The rest of procedures of BP, such as processing of BB nodes and heuristics, are run in a sequential way (1 thread). The algorithms are implemented in C++. Numerical experiments are performed on a 2.7 GHz $i7$-class machine with 8 GB RAM. We set a 3-hour run-time limit for both algorithms.

Our main focus is on processing times of algorithms (T). Also, we report best solutions found (z^{UB}), lower bounds (z^{UB}), the number of generated nodes in BP, and the status of solutions (either *optimal* or *feasible*).

For 38 traffic instances (out of 60), the BP algorithm was able to find an optimal solution in the master node of its BB tree, in times between 0.2 and 10 seconds. In these cases, UB solutions obtained with the heuristic had the same values as the LBs produced by solving the relaxed problem. BB always required more time (up to over $2.5k$ seconds) to solve the problem. A detailed histogram of T is presented in Figure 2.

For the rest of traffic instances, BP performed the search for optimal solutions within its BB tree. Here, in 14 cases (out of 22) the processing times were shorter than of the BB algorithm, with the largest difference of above 3 orders of magnitude. In the remaining 8 cases, BB was faster than BP, but the difference was below 1 order of magnitude for instances solved within the 3-hour limit. All the results are presented in Table II.

TABLE II: Results for instances not solved in the master node of BP (shorter times marked in bold).

	BB (CPLEX)			BP				
$\lvert\mathcal{D}\rvert$	z^{UB}	$T[s]$	*Status*	*Nodes*	z^{LB}	z^{UB}	$T[s]$	*Status*
20	33	109	*optimal*	47	33	33	**6.5**	*optimal*
	35	73	*optimal*	7	35	35	**2.0**	*optimal*
	31	$> 3h$	*feasible*	47	31	31	**6.1**	*optimal*
30	51	**78**	*optimal*	1293	51	51	332	*optimal*
	50	103	*optimal*	63	50	50	**13**	*optimal*
	40	92	*optimal*	73	40	40	**13**	*optimal*
40	51	245	*optimal*	143	51	51	**41**	*optimal*
	53	**59**	*optimal*	687	53	53	212	*optimal*
	60	1187	*optimal*	1655	60	60	**743**	*optimal*
	69	664	*optimal*	5	69	69	**5.1**	*optimal*
	73	1746	*optimal*	1157	73	73	**625**	*optimal*
50	89	**7239**	*optimal*	9440	89	90	$> 3h$	*feasible*
	79	2261	*optimal*	407	79	79	**209**	*optimal*
	64	**1301**	*optimal*	14382	64	65	$> 3h$	*feasible*
	85	**> 3h**	*feasible*	6525	84	86	$> 3h$	*feasible*
	74	**1637**	*optimal*	23361	74	74	10288	*optimal*
60	94	5117	*optimal*	13	94	94	**15**	*optimal*
	86	**3906**	*optimal*	13417	86	87	$> 3h$	*feasible*
	91	2708	*optimal*	215	91	91	**126**	*optimal*
	91	4852	*optimal*	157	91	91	**99**	*optimal*
	92	3565	*optimal*	9319	92	92	**3558**	*optimal*
	69	**1686**	*optimal*	19887	69	69	8061	*optimal*

5. Concluding Remarks

We have developed a branch-and-price-based optimization algorithm for the routing and spectrum allocation problem in elastic optical networks. For almost 90% of evaluated traffic instances, our algorithm was able to find optimal RSA solutions in shorter times than a commercial MIP solver. The performance of BP might be further improved by implementing parallel processing of its BB nodes and its heuristics. Future work concerns appropriate extensions to our BP algorithm, such as improved branching strategies, with the aim to perform efficiently for larger network and traffic instances and to account for such features as distance-adaptive transmission.

ACKNOWLEDGMENTS

This work has been funded by the FP7 project IDEALIST (grant agreement no. 317999). This work was also supported by the Polish National Science Centre (NCN) under grant DEC-2012/07/B/ST7/01215.

REFERENCES

[1] O. Gerstel *et al.*, "Elastic optical networking: A new dawn for the optical layer?" *IEEE Comm. Mag.*, vol. 50, no. 2, pp. 12–20, 2012.
[2] M. Klinkowski and K. Walkowiak, "On the advantages of elastic optical networks for provisioning of cloud computing traffic," *IEEE Network*, vol. 27, no. 6, pp. 44–51, 2013.
[3] ——, "Routing and spectrum assignment in spectrum sliced elastic optical path network," *IEEE Commun. Lett.*, vol. 15, no. 8, pp. 884–886, 2011.
[4] M. Żotkiewicz *et al.*, "Optimization models for flexgrid elastic optical networks," in *Proc. of IEEE ICTON*, Cartagena, Spain, 2013.
[5] L. Velasco *et al.*, "Solving routing and spectrum allocation related optimization problems: from off-line to in-operation flexgrid network planning," *IEEE J. Lightw. Technol.*, vol. 32, no. 16, pp. 2780–2795, 2014.
[6] M. Klinkowski, "An evolutionary algorithm approach for dedicated path protection problem in elastic optical networks," *Cybernetics and Systems*, vol. 44, no. 6-7, pp. 589–605, 2013.
[7] M. Aibin and K. Walkowiak, "Simulated annealing algorithm for optimization of elastic optical networks with unicast and anycast traffic," in *Proc. of IEEE ICTON*, Graz, Austria, 2014.
[8] R. Goścień *et al.*, "Tabu search algorithm for routing, modulation and spectrum allocation in elastic optical network with anycast and unicast traffic," *Com. Net.*, vol. 79, pp. 148–165, 2015.
[9] K. Walkowiak *et al.*, "Routing and spectrum allocation algorithms for elastic optical networks with dedicated path protection," *Opt. Switch. and Network.*, vol. 13, pp. 63–75, July 2014.
[10] IBM, "ILOG CPLEX optimizer," 2012, http://www.ibm.com.
[11] L. Velasco *et al.*, "Modeling the routing and spectrum allocation problem for flexgrid optical networks," *Phot. Netw. Commun.*, vol. 24, no. 3, pp. 177–186, 2012.
[12] C. Barnhart *et al.*, "Branch-and-Price: column generation for solving huge integer programs," *Operat. Res.*, vol. 46, no. 3, pp. 316–329, 1998.
[13] L. Lasdon, *Optimization Theory for Large Systems*. MacMillan, New York, 1970.
[14] M. Ruiz *et al.*, "Column generation algorithm for RSA problems in flexgrid optical networks," *Phot. Netw. Commun.*, vol. 26, no. 2-3, pp. 53–64, 2013.