

Valid Inequalities for the Routing and Spectrum Allocation Problem in Elastic Optical Networks

Mirosław Klinkowski^{1*}, Michał Pióro², Mateusz Żotkiewicz², Marc Ruiz³, and Luis Velasco³

¹ *National Institute of Telecommunications, Warsaw, Poland*

² *Warsaw University of Technology, Warsaw, Poland*

³ *Universitat Politècnica de Catalunya, Barcelona, Spain*

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

ABSTRACT We deal with a natural integer-linear programming formulation of the routing and spectrum allocation (RSA) problem in elastic optical networks. We focus on strengthening the formulation with valid inequalities (cuts). For this purpose we apply clique inequalities that proved to be useful in other applications of this kind. We develop and combine a clique cut generation procedure with a column generation algorithm with the aim to improve the quality of generated columns. The presented results, obtained for a set of problem instances, illustrate the effectiveness of the optimization algorithm.

Keywords: Cliques, column generation, cutting planes, elastic optical networks, ILP, routing and spectrum allocation.

1. INTRODUCTION

The recent developments in optical networking, which include the use of advanced transmission and modulation techniques, spectrum-selective switching technologies, and flexible frequency grids (flexgrids), will let next-generation optical networks to be more spectrally efficient and, in terms of optical bandwidth provisioning, more scalable and elastic [1]. 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 for a set of end-to-end demands that compete for spectrum resources. In an EON, the width of spectrum segment assigned to a lightpath is not rigidly defined but can be tailored to the actual width of the transmitted signal. The RSA optimization problem is \mathcal{NP} -hard [2].

Several alternative mixed-integer programming (MIP) formulations of the RSA problem can be found in the literature; we review them in [3] and [4]. Among these formulations, the formulation proposed in [3], which we refer to as a link-lightpath (LL) formulation, seems to be the most natural one. In the LL modeling approach, the spectrum assignment-related constraints are removed from the MIP by using a set of precomputed lightpaths. At the same time, the LL constraints assure that for each demand a lightpath is selected from the precomputed set and the selected lightpaths are not in conflict with each other, i.e., their spectrum do not overlap on the network links. Although the LL formulation is superior to others in terms of the computation time and the size of network instances handled [3], realistic instances are still difficult to solve due to the large set of the involved integer path-flow variables. To make large instances of RSA tractable by MIP formulations, decomposition methods must be applied [5]. Such methods usually involve the dynamic addition of variables (columns) and/or constraints (cutting planes, cuts) to the MIP model. Recently, a novel column generation-based algorithm (referred to as CG) for dynamic generation of lightpaths for the LL-based MIP formulations of RSA has been proposed in [6].

In this paper, we focus on developing valid inequalities (i.e., cuts) for the LL formulations of RSA. Our focus is on the clique inequalities that have proved to be useful in such kind of applications [7]. In the proposed optimization algorithm, referred to as the column and cut generation (CCG) algorithm, we apply the clique cuts in a column generation framework with the aim to strengthen the problem formulation and, by these means, to generate better quality columns. The effectiveness of the presented method is evaluated for the RSA problem that implements the same objective function as in [6] and [8]. To the best of our knowledge, the presented work is among the first concerning application of cutting planes to RSA problems in flexgrid elastic optical networks.

The remainder of this paper is organized as follows. In Section 2, we present a MIP formulation of the considered RSA problem and, in Section 3, we develop valid clique inequalities appropriate for this formulation. In Section 4 we describe our optimization algorithm. The algorithm is evaluated in Section 5 using the results of numerical experiments. Eventually, in Section 6 we conclude this work.

2. OPTIMIZATION MODEL

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 $\mathcal{S} = \{s_1, s_2, \dots, s_S\}$ 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 used in this and the following section is gathered in Table I.

TABLE I: Notation.

Sets and parameters			
\mathcal{E}	set of links	\mathcal{S}	set of all frequency slices, $\mathcal{S} = \{1, 2, \dots, \mathcal{S} \}$
\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 for demand d using slice s on link e
$d(l)$	demand of lightpath l	$g(l)$	bit-rate carried by lightpath l
$H(d)$	requested bit-rate of demand d	$h(d)$	minimum (guaranteed) bit-rate of demand d
Variables			
x_{dl}	binary, $x_{dl} = 1$ when demand d uses lightpath l ; $x_{dl} = 0$ otherwise		
X_d	binary, $X_d = 1$ when demand d is rejected and $X_d = 0$ when it is accepted		

The MIP model of RSA that we study in this paper is relatively simple and natural [3]. In the 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 ($c \subseteq \mathcal{S}$). 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. 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 . We will refer to such a formulation as a link-lightpath (LL) formulation.

In this paper, we implement and evaluate the same LL formulation as in [6], which is the following:

$$\text{minimize } z = A \sum_{d \in \mathcal{D}} h(d) X_d + \sum_{d \in \mathcal{D}} \left(H(d) - \sum_{l \in \mathcal{L}(d)} g(l) x_{dl} \right) \quad (1a)$$

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

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

where X_d is a binary variable indicating that demand $d \in \mathcal{D}$ is either rejected ($X_d = 1$) or served ($X_d = 0$). For each demand $d \in \mathcal{D}$, $H(d)$ and $h(d)$ denote the requested and the minimum (guaranteed) bit-rate, respectively. For $d \in \mathcal{D}$, $l \in \mathcal{L}(d)$, $g(l)$ denotes the bit-rate of demand d carried by lightpath l . Evidently, $g(l)$ should be in interval $[h(d), H(d)]$ and if such a lightpath cannot be established then demand d is rejected. $\mathcal{Q}(d, e, s) \subseteq \mathcal{L}(d)$ denotes the set of lightpaths of demand d routed through link e and slice s . Eventually, variables in brackets, i.e., λ_d and π_{es} , are dual variables of a linear programming (LP) relaxation of problem (1).

Similarly as in [6], optimization function (1a) minimizes the total amount of unserved bit-rate. By using the weight factor A , the unserved bit-rate from all refused demands (i.e., $X_d = 1$) prevails over the unserved bit-rate from accepted ones ($x_{dl} = 1$), thus giving the priority to the minimization of demand rejection (the first component of (1a)). Constraint (1b) assures that each demand, if accepted (i.e., $X_d = 0$), will use one and only one lightpath from a set of allowable lightpaths. Finally, 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.

3. CLIQUE INEQUALITIES

The LL model assures that in a MIP solution there is at most one selected lightpath for demand d (by constraint (1b)), as well as there is at most one active lightpath that uses slice s on link e (by constraint (1c)). It is reflected by variable x_{dl} that takes a positive value for each such lightpath (in fact, $x_{dl} = 1$ in a MIP solution) and is equal to 0 for the rest of lightpaths. However, for a solution of a linear problem relaxation, in which the integrality of variable x_{dl} is relaxed and substituted by constraint $x_{dl} \geq 0$, it is not a case. To illustrate it, in Fig. 1 we present an example of two allowable lightpaths for demand 1 (denoted as 11 and 12) and one allowable lightpath for demand 2 (denoted as 21) sharing some slices in a network link. According to constraint (1b), lightpaths 11 and 12 are in conflict, i.e., only one of them can be active in a MIP solution. Similarly, constraint (1c) indicates the conflict between lightpaths 11 and 21, and between 12 and 21. Still, in the relaxed problem the values of variables x_{dl} can be all positive (here, equal to 0.5).

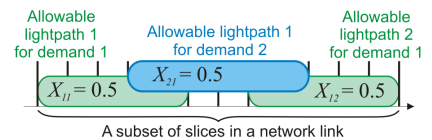


Figure 1: An exemplary clique.

Now, note that only one of these three lightpaths can be active in a MIP solution since they are in conflict with each other. Such a relation can be represented by inequality $x_{11} + x_{12} + x_{21} \leq 1$. Note that this inequality:

- 1) is valid with respect to the feasible solution space of our RSA problem and, hence, for formulation (1),
- 2) may not be present in formulation (1),
- 3) when added to (1), it can cut off a solution of the relaxed problem. Indeed, in our example this inequality is violated by the relaxed solution since we have $x_{11} + x_{12} + x_{21} = 1.5 > 1$.

In the following, we formalize the above observations. We say that a subset of allowable lightpaths $\kappa = \{l_1, l_2, \dots, l_{|\kappa|}\} \subseteq \mathcal{L}$ is in conflict if at most one of these lightpaths can appear in a MIP solution to the RSA problem. We apply a notion of clique [7] when we refer to such a set of lightpaths. For clique κ , we have the following valid clique inequality:

$$\sum_{l \in \kappa, d=d(l)} x_{dl} \leq 1, \tag{2}$$

where $d(l)$ denotes the demand of lightpath l . Obviously, any subset of κ is a clique. Also, note that each of constraints (1b) and (1c) represents a clique inequality.

Now, let $\bar{x} = (\bar{x}_{11}, \dots, \bar{x}_{1|\mathcal{L}(d)|}, \dots, \bar{x}_{|\mathcal{D}|1}, \dots, \bar{x}_{|\mathcal{D}||\mathcal{L}(|\mathcal{D}||)}|)$ denote the current LP solution of the relaxed problem. To be useful as a cut when solving MIP problem (1), a clique inequality should be able to cut off \bar{x} when it is non-integral. The separation problem for the class of clique inequalities is a maximum weighted clique (MWC) problem [7] on a conflict graph. The MWC problem is \mathcal{NP} -complete [7].

Let $\mathcal{G}^c = (\mathcal{L}, \mathcal{Y})$ be the conflict graph for problem (1), where \mathcal{L} is the set of allowable lightpaths, and \mathcal{Y} is the set of edges connecting pairs of lightpaths being in conflict. Graph \mathcal{G}^c can be built using constraints (1b) and (1c). Namely, each pair of lightpaths that appears in the summation at the left hand side of each of these constraints is in conflict and, therefore, an edge connecting the vertices corresponding to these lightpaths can be included into set \mathcal{Y} . Eventually, we assume that each vertex $l \in \mathcal{L}$ in graph \mathcal{G}^c has associated a weight corresponding to the value of variable $\bar{x}_{d(l)l}$ in the current LP solution.

Having graph \mathcal{G}^c built, the MWC problem concerns finding in \mathcal{G}^c a complete subgraph with the maximum sum of the weights of its vertices. The vertices in this subgraph represent a clique and if (2) is violated by this clique then it can be added to formulation (1) to cut off the current LP solution \bar{x} .

4. OPTIMIZATION ALGORITHM

In this section, we develop an optimization algorithm for solving problem (1). The algorithm is implemented in a price-cut-and-branch (PCB) framework and it consists of two phases: first, a set of allowable lightpaths is generated using a column and cut generation (CCG) algorithm, then, the MIP problem (1) is solved for the obtained set of lightpaths using a branch-and-bound (BB) solver. A block diagram of the PCB algorithm is presented in Fig. 2.

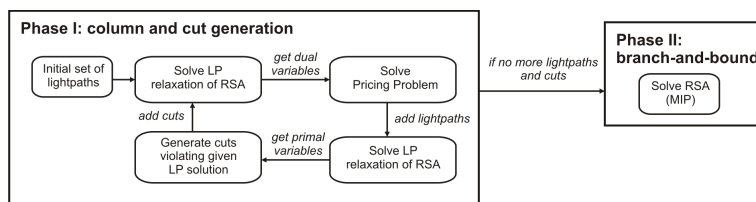


Figure 2: Optimization algorithm.

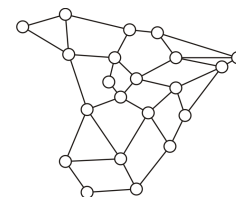


Figure 3: TEL21 network.

4.1 Phase I: column and cut generation

The CCG algorithm is a modified version of the CG algorithm presented in [6] and the modification concerns the application of clique cuts for strengthening formulation (1). In details, in CCG an LP relaxation of problem (1) is initiated and solved with a small, feasible set of allowable lightpaths, denoted as \mathcal{L} . In our implementation, this initial set corresponds to the set of all allowable lightpaths, for each demand d , carrying bit-rate $h(d)$ on a shortest route. Apart from that, a set of clique cuts (i.e., violated clique inequalities), denoted as \mathcal{K} , is generated for these lightpaths and included into formulation (1). Sets \mathcal{L} and \mathcal{K} are then extended iteratively with new lightpaths and cuts. CCG terminates if neither new lightpaths nor cuts can be found and added to \mathcal{L} and \mathcal{K} , respectively.

1) *Column generation*: A key element of column generation is to formulate and solve a pricing problem, which consists of finding such a new lightpath that, when included into the problem formulation, it leads to the improvement in the objective function value of the LP problem relaxation. For formulation (1), the pricing problem reduces to the search for lightpath $l = (p, c)$ for which its reduced cost, calculated as $\lambda_{d(l)} + g(l) - \sum_{e \in p} \sum_{s \in c} \pi_{es}$, is positive. Note that for a given LP solution the values of dual variables λ_d and π_{es} are also given – we use them to calculate the reduced costs. In each iteration of our CCG, for each demand, we include into set \mathcal{L} a lightpath with the largest positive reduced cost, if such lightpath exists. For more details on deriving reduced costs in MIP formulations of RSA, refer to [6].

2) *Clique generation*: To generate clique cuts, we make use of weighted conflict graph \mathcal{G}^c (as discussed in Section 3). Graph \mathcal{G}^c is built for an initial set of allowable lightpaths \mathcal{L} and its vertices and edges are extended for generated lightpaths at each algorithm iteration. The weights of the vertices of \mathcal{G}^c correspond to the components of a given LP solution \bar{x} , thus they change at each iteration. A key element of clique generation is the search for a maximal complete subgraph in \mathcal{G}^c for which the sum of the weights of its vertices is greater than 1. To find such subgraphs, we apply a modified version of the branch-and-bound algorithm solving a maximum clique (MC) problem [9]. The objective of the original MC algorithm is to find a clique that consists of a maximum number of vertices. To be useful for our CCG algorithm, the modification of MC consists in: a) taking into account the weights of cliques when looking for maximal cliques in \mathcal{G}^c , and b) providing a number of maximal cliques with just one algorithm run. To speed up the procedure, the MWC search is performed first on a subgraph of \mathcal{G}^c including only the vertices with positive weights, and then after finding a set of cliques on this subgraph, these cliques are lifted by considering the rest of vertices from \mathcal{G}^c (see [7] for more details).

4.2 Phase II: branch-and-bound

Once set \mathcal{L} is found using the CCG procedure, it is used as a set of allowable lightpaths in the MIP problem and this problem is now solved using branch-and-bound. Since set \mathcal{L} is much smaller than the set containing all possible lightpaths, our MIP problem can be solved in shorter times. Note that although the optimality of the LP problem relaxation is guaranteed after column and cut generation, still the obtained set of allowable lightpaths does not necessarily lead to an optimal solution of the MIP problem.

5. NUMERICAL RESULTS

In this section, we evaluate the performance of the CCG algorithm. The evaluation is performed for a generic Spanish network with 21 nodes and 35 links, denoted as TEL21 and presented in Figure 3. We consider the flexgrid of 12.5 GHz granularity and different values of the available spectrum in network links (denoted as S). The spectral efficiency of a modulation format is equal to 2 bit/s/Hz. The demand sets are randomly generated and according to the same traffic profile as in [6]. Namely, we have three types of demands with bit-rate requests $(h(d), H(d)) \in \{(10, 40), (40, 100), (100, 200)\}$ Gbit/s, where the percentage of each demand type in the demand set is distributed as 25% : 25% : 50%, respectively. The number of demands is $|\mathcal{D}| \in \{64, 112, 160\}$.

As reference algorithms we use a column generation (CG) algorithm, which is a reduced version of CCG that does not include cut generation, and a branch-and-bound (BB) solver solving formulation (1) for a complete set of candidate lightpaths. The number of candidate routing paths $k \in \{10, 30\}$ 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. The algorithms are implemented in C++. We use CPLEX v.12.5.1 [10] as a LP solver in the column and cut generation phase and as a BB solver in the final algorithm phase. Numerical experiments are performed on a 2.5 GHz i5-class machine with 8 GB RAM. We set a time limit equal to either 1 or 10 hours for solving MIP problems in CPLEX.

When comparing the algorithms, our main focus is on the best integer solution found, denoted as z_Y^X , where $X \in \{BB, CG, CCG\}$ denotes the algorithm and $Y \in \{1h, 10h\}$ denotes the CPLEX run-time limit. Apart from that, we report: the number of initial allowable lightpath (L^{init}), the number of generated lightpaths (L^{gen}), the number of generated cuts (K^{gen}), the value of LP relaxation after performing the CG and CCG procedure (z^{LP}), the CPLEX gap between the best integer solution and a lower bound solution (gap), and algorithm runtime (T).

In Table II, we present numerical results obtained for a set of $|\mathcal{D}| = 64$ demands, $k = 30$, and for different values of available spectrum (S) after 1 hour of CPLEX computation. We can see that the BB algorithm is able to provide optimal (or near-optimal) solutions only for the scenarios with low values of S . Concurrently, both CG and CCG, which require a smaller set of allowable lightpaths, are always able to find solutions close to the LP lower bound solution (z^{LP}). We can also see that CCG outperforms CG in almost all cases and it is able to find optimal results for smaller scenarios at the cost of some increment in computation time (required for column and cut generation). In smaller scenarios (with $S \in \{100, 150\}$), we can see that CCG provides better lower bounds (z^{LP}) than CG.

TABLE II: Comparison of algorithms (in bold we mark best results); $|\mathcal{D}|=64$, $k=30$.

S [GHz]	BB				CG						CCG						
	L^{init}	z_{1h}^{BB}	gap	T	L^{init}	L^{gen}	z^{LP}	z_{1h}^{CG}	gap	T	L^{init}	L^{gen}	K^{gen}	z^{LP}	z_{1h}^{CCG}	gap	T
100	30240	20940	0%	12	400	294	20815	21175	0%	2	400	1842	1878	20921.2	20940	0%	39
150	59040	13300	0%	351	656	633	13097.5	13350	0%	7.6	656	2366	2733	13122.5	13300	0%	299
200	87840	7930	3.1%	3600	912	690	7680	8155	0%	1788	912	1299	1571	7680	8005	0.6%	3667
300	145440	2555	20%	3600	1424	575	2040	2590	10%	3610	1424	657	830	2040	2615	9.2%	3626
400	203040	4000	75%	3600	1936	638	1000	1350	0%	1933	1936	703	1308	1000	1335	1.3%	3641
500	260640	4015	95%	3600	2448	820	200	690	8.9%	3623	2448	1016	1417	200	410	0%	1270

TABLE III: Averaged results over 5 experiments for larger scenarios.

S [GHz]	$ \mathcal{D} $	k	BB		CG				CCG				CCG vs. CG		
			L^{init}	z^{LP}	L^{init}	L^{gen}	z_{1h}^{CG}	z_{10h}^{CG}	L^{init}	L^{gen}	F^{gen}	z_{1h}^{CCG}	z_{10h}^{CCG}	Δ_{1h}	Δ_{10h}
500	64	30	260640	220	2448	796.8	713	713	2448	1043.2	1476.6	575	573	24%	24%
850	112	10	269640	704	7420	2435.2	1703	1567	7420	2938.6	2474.8	1576	1407	8.1%	11.4%
1200	160	10	553200	692	15080	3578.8	1893	1626	15080	4208.8	3273.2	1829	1525	3.5%	6.6%

In Table III, we present the results obtained for larger problem instances. In each scenario, the evaluation is performed for 5 different demand sets and the presented results are averaged. We report the best solution found (z_{γ}^X) after both 1 hour and 10 hours of CPLEX computation. We do not present the results for BB since either their quality is low or the algorithm is not able to find any in a given time limit, which is mainly due to the large number of allowable lightpaths (L^{init}) given in these scenarios. Both CG and CCG are still able to provide good feasible solutions and, on average, the quality of CCG solutions is better than after running the CG procedure only. The average relative difference (denoted as Δ_{γ}) between the results of z^{CG} and z^{CCG} is up to some tens of percents.

6. CONCLUSIONS

We have focused on developing valid clique inequalities for a natural integer-linear programming formulation of the routing and spectrum allocation problem in elastic optical networks. To generate the clique cuts we have proposed a cut generation procedure. The generated cuts have been used in a column generation algorithm with the aim to strengthen the MIP problem formulation and, by these means, to improve the quality of generated columns. The presented results, obtained for a set of problem instances, show that the proposed optimization algorithm is able to provide better solutions than a basic column generation algorithm. In future work we will focus on the development of exact methods for solving RSA in EONs, based on the price and branch approach.

ACKNOWLEDGMENTS

This work has been funded by the Polish National Science Centre under grant DEC-2011/01/D/ST7/05884. This work was also supported by the FP7 project IDEALIST (grant agreement no. 317999).

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, "Routing and spectrum assignment in spectrum sliced elastic optical path network," *IEEE Commun. Lett.*, vol. 15, no. 8, pp. 884–886, 2011.
- [3] 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.
- [4] M. Żotkiewicz *et al.*, "Optimization models for flexgrid elastic optical networks," in *Proc. of IEEE ICTON*, Cartagena, Spain, 2013.
- [5] L. Lasdon, *Optimization Theory for Large Systems*. MacMillan, New York, 1970.
- [6] 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.
- [7] A. Atamtürk *et al.*, "Conflict graphs in solving integer programming problems," *Europ. J. of Operat. Res.*, vol. 121, no. 1, pp. 40–55, 2000.
- [8] M. Klinkowski *et al.*, "Elastic spectrum allocation for time-varying traffic in flexgrid optical networks," *J. of Sel. Areas in Comm.*, vol. 31, no. 1, pp. 26–38, 2013.
- [9] J. Konc and D. Janezic, "An improved branch and bound algorithm for the maximum clique problem," *MATCH Commun. Math. Comput. Chem.*, vol. 58, pp. 569–590, 2007.
- [10] IBM, "ILOG CPLEX optimizer," 2012, <http://www.ibm.com>.