A Column Generation Approach for Large-Scale RSA-Based Network Planning

M. Ruiz[1]*, M. Żotkiewicz[2], L. Velasco[1], J. Comellas[1]
[1] *Universitat Politècnica de Catalunya (UPC), Barcelona, Spain*
[2] *Warsaw University of Technology (WUT), Warsaw, Poland*
**e-mail: mruiz@ac.upc.edu*

ABSTRACT
In flexgrid-based optical networks, the problem of finding optimal route and spectrum allocation (RSA) for the demands in a traffic matrix becomes harder to solve than similar problems in fixed DWDM networks, mainly due to the finer spectrum granularity and the spectrum contiguity and continuity constraints. Additionally, the increasing traffic volumes and the size of real networks lead to network planning problem instances consisting of hundreds of thousands, or even millions, of (binary) variables. In this work, we present a column generation decomposition method to obtain feasible solutions for RSA-based network planning problems. Numerical results show the utility of this method for solving intractable instances.

Keywords: Column Generation, Routing and Spectrum Allocation, Flexgrid Optical Networks

1. INTRODUCTION

Recently proposed flexgrid optical networks (FG-ON) are presented as the most promising solution to deal with the future expected high volumes of data traffic [1]. In FG-ON, the problem of finding available spectrum resources for establishing new lightpaths is called the Routing and Spectrum Allocation (RSA) problem. RSA consists in assigning a *contiguous* and *continuous* fraction of spectrum to a demand subject to the constraint of no frequency overlapping in network links. The RSA optimization problem not only is NP-hard but also it is more difficult than its predecessor Routing and Wavelength Assignment (RWA) problem in fixed grid wavelength division multiplexing (WDM) networks. Among the several alternative Mixed Integer Linear Programming (MILP) formulations found in literature, the one proposed in [2] appears to be the most effective since both continuity and contiguity constraints are removed from the MILP by using both pre-computed routes and channels. Although this formulation improves previous ones in terms of computational time and size of solvable instances, large instances are really difficult to solve due to the large set of binary variables.

To deal with those large instances, decomposition methods can be derived to improve the tractability of such instances. Column Generation (CG) is a decomposition method that allows reducing the amount of variables (*i.e.* columns) in LP-based problem formulations [3], and has been successfully used for solving design problems in communication networks. In CG, a problem with an initial and small set of columns is extended with new columns obtained by iteratively solving the so-called *pricing* problem.

In this paper we introduce a column generation algorithm for RSA-based MILP formulations. In Section 2, a RSA problem is introduced and formally stated from the formulation proposed in [2]. Then in Section 3, a CG algorithm based on providing lightpaths in accordance with spectrum requirements of FG-ON is introduced and numerically evaluated in Section 4, by means of large instances over a real reference network. Note that, although CG has been deeply studied for solving the RWA problem (*e.g.*, see [4]), such methods are not applicable to RSA since the flexible spectrum allocation in flexgrid networks differs from the fixed-grid channel assignment in WDM.

2. PROBLEM STATEMENT AND FORMULATION

The proposed RSA problem is stated as follows: find, for each demand from a set of demands, the route over the flexgrid optical network and the spectrum allocation to minimize the number of *blocked* demands (primary objective) and the amount of *unserved bit-rate* (secondary objective). The served bit-rate of each demand is a value in $[h_d, H_d]$, where h_d is the minimum and H_d is the maximum bit-rate, respectively. If h_d cannot be satisfied, then the demand is blocked. Each established lightpath is subject to the following constraints: a) *spectrum contiguity*: the channel consists of a contiguous subset of slots; b) *spectrum continuity*: the channel is the same for each link on the lightpath; c) *channel capacity*: the channel serves a bit-rate between h_d and H_d; d) *slot capacity*: a slot in a link can be allocated to one demand at most.

Table 1 shows the parameters and variables to model the above-stated RSA problem. From such notation and

This work has been partially funded by the FP7 project IDEALIST (grant agreement no. 317999) and by the Spanish Science Ministry through TEC2011-27310 ELASTIC project. M. Żotkiewicz was supported by the EC European Social Fund through the Warsaw University of Technology Development Programme.

the formulation proposed in [2], the RSA MILP can be easily obtained. However, for applying the column

TABLE 1 PROBLEM NOTATION

Sets and parameters		C_d	Channels for demand $d \in D$ $c \in C_d \leftrightarrow n_d \le m_c \le N_d$
D	Set of demands	P_d	Set of lightpaths for demand $d \in D$
h_d	Minimum volume of demand $d \in D$ (in Gbps)	P	Set of all lightpaths $P = \cup_{d \in D} P_d$
H_d	Maximum volume of demand $d \in D$ (in Gbps)	Q_{des}	Set of lightpaths for d using slot $s \in S$ on link $e \in E$
E	Set of network links	$E(p)$	Set of links traversed by lightpath $p \in P$
S	Set of frequency slots at a link	g_p	Bandwidth carried on lightpath $p \in P$, computed as follows: $g_p = \begin{cases} H_{d(p)} & \text{if } m_{c(p)} = N_{d(p)} \\ B \cdot m_{c(p)} & \text{if } n_{d(p)} \le m_{c(p)} < N_{d(p)} \end{cases}$
B	Slot bandwidth (in GHz)		
n_d	Number of slots required to carry h_d, $d \in D$ $n_d = \lceil h_d/B \rceil$	A	Weight for objective function
N_d	Number of slots required to carry H_d, $d \in D$ $N_d = \lceil H_d/B \rceil$	**Decision variables**	
C	Set of channels	X_d	binary, X_d=1 when d is blocked at all
m_c	Number of slots used by channel $c \in C$	x_{dp}	binary, x_{dp}=1 when lightpath p carries its demand d
$S(c)$	Set of (contiguous) slots composing channel c	Y_d	continuous, un-served bandwidth of demand d

generation algorithm, we need to define the *master problem* (RSA-M) which consists in the same set of constraints defined for the MILP but considering that all variables are defined in the continuous range. Thus, the objective function of the RSA-M problem is:

$$(\text{RSA-M}) \quad Minimize \quad F = A \cdot \sum_{d \in D} h_d \cdot X_d + \sum_{d \in D} Y_d \tag{1}$$

subject to the following set of constraints (where associated dual variables are represented in brackets):

$$[\lambda_d] \quad X_d + \sum_{p \in P_d} x_{dp} = 1 \quad d \in D \tag{2}$$

$$[\pi_{es} \ge 0] \quad \sum_{d \in D} \sum_{p \in Q_{des}} x_{dp} \le 1 \quad e \in E, s \in S \tag{3}$$

$$[\gamma_d] \quad Y_d + \sum_{p \in P_d} g_p x_{dp} = H_d \quad d \in D \tag{4}$$

$$X_d \ge 0, x_{dp} \ge 0, Y_d \text{ unconstrained in sign} \quad d \in D, p \in P_d \tag{5}$$

The multi-objective function (1) minimizes both, the number of blocked demands and the amount of un-served bit-rate; parameter A weights both objectives. Constraint (2) assigns a lightpath to a demand or blocks that demand if it cannot be served. Moreover, it guarantees that just one lightpath is assigned per demand. Constraint (3) makes sure that one lightpath at most uses one specific slot. Constraint (4) sets Y_d as the unserved bit-rate. Finally constraint (5) defines the sign of variables for the RSA-M according to original variable definitions.

3. PROPOSED COLUMN GENERATION ALGORITHM

Table 2 gives insight of the proposed CG algorithm. It starts generating an initial set of lightpaths (recall that each consists in the tuple {route, spectrum allocation}) and solves the RSA-M for only that limited set of variables (called *restricted master problem*, RSA-RM) (lines 1 and 2). Then, an iterative procedure is executed as long as it finds new lightpaths to be added to the RSA-RM (lines 4 to 12). Every time a subset of new lightpaths is found, it is added to the set of existing lightpaths and the problem is solved again. When no further lightpaths are found, the column generation methodology ensures that the last solution is optimal, not only for the RSA-RM but also for the RSA-M problem. Notwithstanding, that solution is continuous; to obtain an integer solution for the original problem, the RSA MILP is solved with all the generated lightpaths (line 13).

At each iteration the algorithm aims at finding, for each demand, the lightpath with the highest positive reduced cost z_{dp} (computed from (6)), which is the one that would provide the highest improvement on the solution quality. To this aim, the shortest route over the network with metric links depending on π dual variables is found. Since π variables are related with single slots in links, the metric of a given link (f_e) depends on the selected channel and is computed as defined in (7). Due to the dependency of link metrics on the

TABLE 2 PROPOSED CG ALGORITHM

```
INPUT: Network G(V,E),Demands D
OUTPUT: Solution
P ← initialize(G,D); P'←P
Generate RSA-MR with lightpath set P
while P' ≠ Ø do
 P' ← Ø
 Solution ← Solve RSA-MR
 for each demand d and  do
   for each channel c in Cd do
     Update cost links f from eq.(7)
     Compute and store Zdp from eq.(6)
     p*← arg max{Zdp,∀computed p | Zdp>0}
    if p*≠ Ø  then add p* to P'
 Add P' to RSA-MR
Solution ← Solve RSA-P
```

$$z_{dp} = \lambda_d + g_p - \sum_{e\in E(p)} \sum_{s\in S(p)} \pi_{es} \tag{6}$$

$$f_e(c) = \sum_{s\in C} \pi_{es} \quad e \in E \tag{7}$$

TABLE 3 *INITIALIZE* ALGORITHM

```
Input: Network G(V,E), Demand D
Output: Lightpath set P
P ← Ø
for each d in D do
  r ← Shortest route of d in G
  for each channel c with nd
  slots do
    p ← lightpath with r and c
    Add p to P
```

selected channel, several shortest routes (one for every possible channel size and allocation) need to be computed to find the best lightpath. This search is optimized by means of combining the Floyd-Warshall algorithm [5] for fast evaluation of reduced costs and the Dijkstra algorithm for obtaining those specific lightpaths to be used in the RSA-M. Details and references of these algorithms, as well as extended descriptions of the CG algorithm, can be found in [6].

Since our original problem aims to minimize the amount of unserved bitrate, the CG algorithm generates those lightpaths to carry high volume demands. However, our objective function strongly penalizes demands which cannot be served at all. To overcome this issue, we propose an algorithm to initialize the set of variables based on providing lightpaths for minimum bit-rates. This algorithm (in

Table 3 introduces several lightpaths per demand, *i.e.*, those characterized by the shortest route in hops and all the channels with size equal to n_d slots.

4. ILLUSTRATIVE NUMERICAL RESULTS

For numerical evaluation, we consider the 21-node Spanish Telefónica optical network (presented in Fig. 1). Several problem instances were generated, each consisting of a set of randomly generated demands. For each demand, a random source/destination pair is selected, being each possible pair chosen with uniformly distributed probability. Then, the minimum and maximum bit-rate (and required number of slots) for a demand was set by randomly choosing one of the following (h_d, H_d) pairs: (10, 40), (40, 100), (100, 200). We assumed a frequency slot width of 12.5 GHz and a spectral efficiency of 2b/s/Hz, so that the amount of bit-rate that a single slot can carry is 25 Gb/s. To identify the size of an instance we use indistinctly the number of demands or the total load of bit-rate to serve (in Tb/s). We implemented the CG algorithm in Matlab [7] and used the linear solver engine of CPLEX 12.2 [8]. All experiments were run on a 2.4GHz Quad-core machine with 8 GB RAM running Linux.

Aiming at evaluating the performance of our CG algorithm, we generated medium instances ranging from 2 to 4 Tb/s (32 to 64 demands). Table 4 shows the results obtained for the network with 40 frequency slots links. Additionally, the RSA-M was solved considering sets of pre-computed lightpaths, obtained by finding the K shortest routes for each demand and, for each route, all possible channels to serve the demand bit-rate. After a quick inspection, one can observe that our CG algorithm always meet the optimal RSA-M objective function with a number of lightpaths remarkably low when compared to pre-computed lightpaths (differences of one order of magnitude for $K = 5$). Regarding CG execution time, we distinguish between the time needed for doing iterative operations in CG algorithm (*e.g.*, computing shortest paths) and the time spent by CPLEX to solve linear relaxations. We can validate the applicability of this method when instance size grows, since both times increase softly with respect to the instance size as well as the number of CG iterations.

To illustrate the quality of the variables for obtaining integer solutions, Fig. 2 plots, for a given instance, the objective function of the integer solution obtained by our method and by pre-computing all lightpaths from the shortest K routes. In light of the results, $K = 4$ shortest routes are necessary to obtain an integer solution with the same quality that the one obtained by our method. This K value leads to a total number of lightpaths close to 27000, more than one order of magnitude higher than the number of lightpaths generated by our CG approach (lower than 2000 lightpaths).

Finally, we generated large instances based on the same network topology but with larger number of demands and spectrum slots. By also considering a larger K value (equal to $K = 10$, to practically ensure that optimal solutions will be reached), the number of variables rises to intractable values. Table 5 shows meaningful results

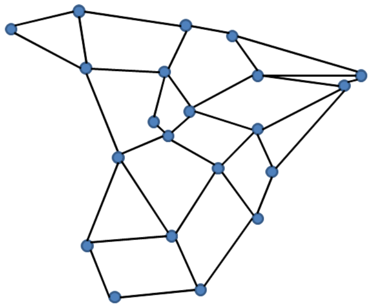

Figure 1. Evaluated network (21 nodes, 35 links).

TABLE 4 CGA PERFORMANCE (MEDIUM INSTANCES)

	Pre-Comp. Lightpaths (K=5)		CGA				
load Tb/s (|*D*|)	**|*P*|**	***o.f.***	**# iter.**	**|*P*|**	***o.f.***	**CGA Time (sec.)**	**CPLEX Time (sec.)**
2.0 (32)	21,784	0	6	1,429	0	7.86	0.58
2.5 (40)	27,230	0	7	1,824	0	9.94	1.12
3.0 (48)	32,666	0	9	2,299	0	15.20	2.67
3.5 (56)	38,122	0	10	2,748	0	19.89	4.4
4.0 (64)	43,568	220	13	3,381	220	27.83	9.24

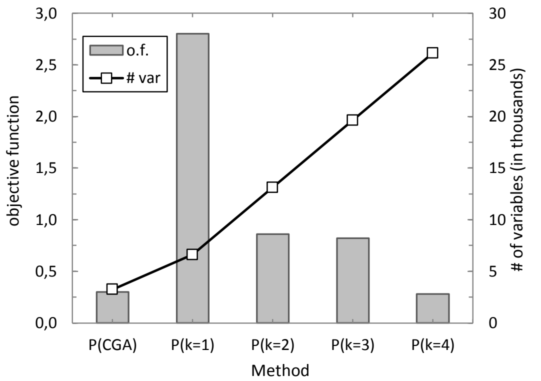

Figure 2. Quality of generated lightpaths for 3 Tb/s.

TABLE 5 CGA PERFORMANCE (LARGE INSTANCES)

		Pre-Comp. Lightpaths (K=10)			CGA		
load Tbps (|*D*|)	**|*S*|**	**|*P*|**	**lower bound**	**best int.**	**|*P*|**	**lower bound**	**best int.**
4 (64)	40	132,400	520	1895	3,287	520	775
7 (112)	68	269,640	0	7420	9,338	0	785
10 (180)	96	8,675,190	*out-of-memory*		17,203	0	857

for these large instances. In both cases (pre-computed lightpaths and CG), the time limit set for CPLEX was 10 hours. As can be observed, our method provides, with up to 2 orders of magnitude less number of variables, better integer solutions in the same running time than simply applying CPLEX with pre-computed lightpaths. Moreover, in such cases when the number of variables is too large to allow generating the problem by CPLEX (*out-of-memory* messages appear), our CG approach provides an affordable way to obtain feasible and good-quality solutions.

5. CONCLUSIONS

In this work a specific method for generating lightpaths for RSA-based problems has been presented. The method, based on column generation decomposition methods, finds good quality lightpaths to introduce in the decision variable set of ILP formulations. This method aims to improve current methodology of adding large sets of pre-computed lightpaths that leads to intractable instances when the number of variables increases. Illustrative numerical results over medium and large instances proved that the application of our CG approach allows reducing remarkably the number of variables (in up to 2 orders of magnitude) while keeping (or even improving) the quality of the obtained solutions. Moreover, our method becomes a practical way for obtaining feasible solutions from ILP formulations for those instances where the number of variables is expected to be large. This CG algorithm could be adapted into a branch-and-price algorithm with the aim to obtain optimal integer solutions in a more efficient way than the classical branch-and-bound approach.

REFERENCES

[1] ITU-T Recommendation G.694.1 (ed. 2.0), "Spectral grids for WDM applications: DWDM frequency grid," Feb. 2012.

[2] L. Velasco, M. Klinkowski, M. Ruiz, and J. Comellas, "Modeling the routing and spectrum allocation problem for flexgrid optical networks," *Phot. Netw. Commun.*, vol. 24, no. 3, pp. 177–186, 2012.

[3] M. Pióro and D. Medhi, "*Routing, Flow, and Capacity Design in Communication and Computer Networks*," Morgan Kaufmann, 2004.

[4] B. Jaumard, C. Meyerb, and B. Thiongane, "On column generation formulations for the RWA problem*," Discrete. Appl. Math.*, vol. 157, pp. 1291-1308, 2009.

[5] R.W. Floyd, "Algorithm 97: Shortest path," *Commun. ACM* 5(6), 345– (1962). DOI 10.1145/367766.368168.

[6] M. Ruiz, *et al.*, "Detailed description of column generation algorithms for RSA problems in flexgrid optical networks," Tech. rep., UPC-DAC-RR-2013-15, Barcelona, Spain (2013).

[7] Matlab, http://www.mathworks.es/products/matlab/

[8] IBM, "ILOG CPLEX optimizer," 2012, http://www.ibm.com.