Solving Routing and Spectrum Allocation Related Optimization Problems

L. Velasco[*], A. Castro, M. Ruiz

Universitat Politècnica de Catalunya (UPC), Barcelona (Spain), Email: lvelasco@ac.upc.edu

Abstract *We provide a comprehensible introduction to RSA-related problems in flexgrid networks. Starting from its formulation, we analyze network live cycle and indicate different solving methods for the kind of problems that arise at each network phase: from the initial network planning to network re-optimization, going through network operation.*

Introduction

Flexgrid optical networks [1] are attracting huge interest due to their higher spectrum efficiency and flexibility compared to wavelength switched optical networks (WSON). To properly analyze, design, plan, and operate flexgrid networks, efficient methods are required for the routing and spectrum allocation (RSA) problem.

The RSA problem is similar to the routing and wavelength assignment (RWA) problem in WSON. The allocated spectral resources must be, in absence of spectrum converters, the same along the links in the route (the *continuity constraint*). However, the RSA problem adds new constraints to guarantee that spectrum is also contiguous in the spectrum (the *contiguity constraint*). The contiguity constraint adds huge complexity to the RSA problem, which was proved to be NP-complete in [2]; as a consequence, it is crucial that efficient methods are available to allow solving realistic problem instances in practical times.

In this paper, we review different RSA-related optimization problems that arise along the live cycle of flexgrid networks. Different methods to solve those optimization problems are reviewed along with the different requirements where those problems need to be solved.

RSA-related problems

In the context of flexgrid optical networks, the majority of optimization problems are extensions of the basic RSA problem. An example of optimization problem involving the RSA problem is the typical network planning problem, which can be stated as:

Given: *i*) a set *N* of locations and a set of optical fibers *E* connecting those locations; *ii*) the characteristics of the optical spectrum (i.e., spectrum width, frequency slot width) and the set of modulation formats; *iii*) a traffic matrix *D* with the amount of bitrate exchanged between each pair of locations in *N*; *iv*) the cost of every component, such as optical cross-connects (OXC) and transponder (TP) types specifying its capacity and reach.

Output: *i*) Network dimensioning including the type of OXC and TPs in each location; *ii*) the route and spectrum allocation for each demand in *D*.

Objective: Minimize the total cost to transport the given traffic matrix.

Those problems can be formulated using mixed integer linear program (MILP). Several ways to model the same problem usually exists, being some of them more preferable than others. In that regard, we showed in [3] that the use of a pre-computed set of channels (i.e. contiguous spectrum fractions) allows considerably reducing the problem complexity. In that study, we addressed a RSA problem in which enough spectrum needed to be allocated for each demand of a given traffic matrix and presented novel MILP formulations of RSA based on the assignment of channels. The evaluation results revealed that the proposed approach allows solving the RSA problem much more efficiently than previously proposed ILP-based methods and it can be applied even for realistic problem instances, contrary to previous ILP formulations.

Solving RSA-related problems

MILP formulations can be solved to optimality using commercial solvers. Notwithstanding, due to the NP-completeness of the RSA problem, un-tractability of MILP formulations appears when instances to be solved involve a large number of variables (e.g. large size of sets *N*, *E* and *D* in the above problem).

Since the size of the problems can be really large, solving MILPs might entail problems with literally thousands of millions of (integer or binary) variables. To deal with this complexity, *large scale optimization* (LSO) methods can be used. The objective of LSO methods is to improve the exact methodology based on classical Branch & Bound (B&B) algorithm for solving MILP formulations.

Among different methods, decomposition methods such as *column generation* (CG) and *Benders decomposition* have been successfully used for solving communications network design problems. Detailed algorithms for decomposition and other methods can be found in [4].

CG consists in using a small set of variables (columns) to solve the linear relaxation of the MILP formulation. Then, according to the dual variables obtained after solving linear relaxation, an algorithm runs looking for new variables suitable to improve the current solution. At each iteration, the problem is solved adding those new variables until the iterative algorithm ends when no more variables are found. In the context of networks variables are mainly paths so this technique is also known as path generation; search for new variables to add can be roughly defined as a secondary problem (*pricing problem*) consisting in finding paths in a network whose costs depend on dual variables values. Note that CG does not ensure integer optimal solutions, however it can be combined with the B&B algorithm so CG is applied inside each tree node to create the Branch & Price algorithm.

Benders decomposition method is an iterative procedure based on projecting out a subset of variables from the original problem with the whole set of variables and creating new constraints from the projected ones. In contrast with previous decomposition method, this methodology adds inequalities to the linear formulation to solve, thus strengthening lower bounds and speeding up the convergence to integer optimal solutions. Similar to this strategy, the combination of B&B with other methods to generate inequalities or cuts, such as cutting plane, derives into the Branch & Cut algorithm.

Nonetheless, when the time to find a solution is critical, which happens when the network is in operation, a better trade-off between solutions' quality and time-to-compute can be obtained by relaxing optimality condition to find near optimal solutions much more quickly.

In that regard, heuristic algorithms are the way to produce sub-optimal feasible solutions. In particular, *metaheuristics* (high-level strategies) guide a problem specific heuristic, to increase their performance avoiding the disadvantages of iterative improvement allowing escaping from local optima. Although a large variety of meta-heuristic methods have appeared in the literature, in this paper we restrict ourselves to describe only two: the greedy randomized adaptive search procedure (GRASP)[5] and the biased random-key genetic algorithm (BRKGA)[6].

The GRASP procedure is an iterative two phase metaheuristic method based on a multi-start randomized search technique. In the first phase, a greedy randomized feasible solution of the problem is generated through a *construction* algorithm. Then, in the second phase, a *local search* technique to explore an appropriately defined neighborhood is applied in an attempt to improve the current solution. These two phases are repeated until a stopping criterion (e.g., a number of iterations) is met, and once the procedure finishes the best solution found over all GRASP iterations is returned. In addition to local search, path-relinking (PR) can be used as an intensification strategy that explores trajectories connecting GRASP solutions. It starts at a so-called initiating solution and moves towards a so-called guiding solution. To ensure that PR is only applied among high-quality solutions, a set ES must be both maintained and cleverly managed during all GRASP iterations. With the attribute high-quality we are not only referring to their cost function value but also to the diversity they add to ES. GRASP+PR have been successfully used in many applications including flexgrid network defragmentation [7].

The BRKGA metaheuristic, a class of GA, has been recently proposed to effectively solve RSA-related optimization problems [8]. Compared to other meta-heuristics, BRKGA has provided better solutions in shorter running times. As in GAs, each individual solution is represented by an array of n genes (chromosome), and each gene can take any value in the real interval [0,1]. Each chromosome encodes a solution of the problem and a fitness value, i.e., the value of the objective function. A set of individuals, called a population, evolves over a number of generations. At each generation, individuals of the current generation are selected to mate and produce offspring, making up the next generation. In BRKGA, individuals of the population are classified into two sets: the *elite* set with those individuals with the best fitness values and *non-elite* set. Elite individuals are copied unchanged from one generation to the next, thus keeping track of good solutions. The majority of new individuals are generated by combining two elements, one elite and another non-elite, selected at random (crossover). An inheritance probability is defined as the probability that an offspring inherits the gene of its elite parent. Finally, to escape from local optima a small number of mutant individuals (randomly-generated) to complete a population are introduced at each generation. A deterministic algorithm, named *decoder*, transforms any input chromosome into a feasible solution of the optimization problem and computes its fitness value. In the BRKGA framework, the only problem-dependent parts are the chromosome internal structure and the decoder, and thus, one only needs to define them to completely specify a BRKGA heuristic.

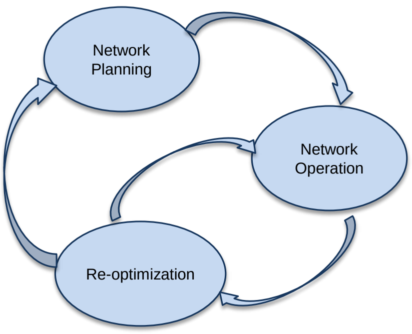

Fig. 1. Three phases of flexgrid network live cycle.

Optimization within the network live cycle

Different optimization problem arise during the live cycle of telecom networks. The classic telecom network life-cycle consists of two main phases: *i*) before operation, the network is designed and dimensioned during the *planning phase*; *ii*) *network operation* where the network is in service. However, dynamic connection set-up and teardown entails suboptimal resources utilization. Thus, a new phase, named *re-optimization*, is needed. Fig. 1 presents the entire flexgrid network live cycle.

As introduced above, LSO methods are oriented to solve MILP formulations when the number of variables is very large (e.g. millions of variables). In network planning problems, it is easy to find real huge size instances. Indeed, the complexity of solving RSA for a static demand matrix increases rapidly when other features such as, for example, network topology design, protection against failures, or multi-hour traffic variations are considered. Although the majority of complex network planning problems are finally solved by means of a heuristic method, the benefits provided by obtaining the optimal solution instead of just sub-optimal ones backs the application of LSO.

Once the network is in operation a control plane base on the PCE architecture presented in Fig. 2 can be used for connection provisioning and network re-optimization. Requests arrive to the front-end PCE which are computed running one of the algorithms locally deployed. To solve the RSA problem for just one connection, exact algorithms that are able to provide an optimal solution in real-time can be used; see e.g. [7].

In the case of bulk RSA computation, the RSA problem need to be solved for a limited number of connections, e.g. after a link has failed restoration routes need to be computed for the set of optical connections affected. In this scenario, computation times need to be kept as short as possible, e.g. hundreds of milliseconds are preferred to several seconds. In such cases, GRASP heuristics can be devised to provide good solutions in really short times [9].

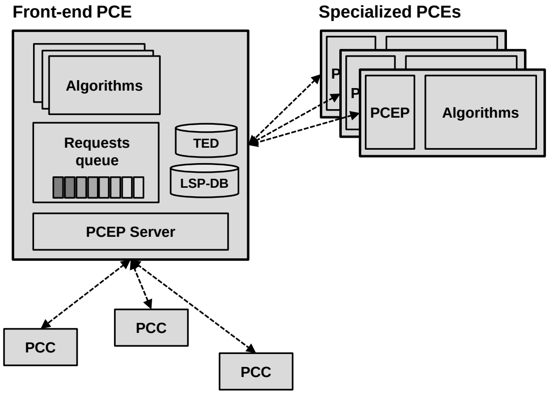

Fig. 2. Front and back-end PCEs for network operation and re-optimization.

Finally, re-optimization problems need to be solved triggered by some condition (e.g., provisioning blocking, network performance reaches a threshold or new available resources appear after a failure repair). In this re-optimization scenario, the problem size can be small but also as large as those for network planning.

When the problem is related to provisioning, ultra-fast heuristics such as those in [7,9] can be applied. However, when the temporal requirements to solve those re-optimization problems can be relaxed to tens of minutes, LSO methods can be used to find good-quality solutions. Re-optimization algorithms can be run in specialized PCEs in the back-end, and are called by the front-end PCE upon some network condition arises.

Conclusions

Several RSA-related optimization problems that arise along the live cycle of flexgrid networks have been introduced. To solve them several methods have been reviewed which can be used as a function of the requirements where those problems need to be solved.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2012-2015 under grant agreement no. 317999 IDEALIST project and by the MINECO project ELASTIC (TEC2011-27310).

References

1 M. Jinno, et al., IEEE Comm. Mag. 47 (2009)
2 K. Christodoulopoulos et al., J. Lightwave Technol. 29 (2011)
3 L. Velasco et al., Photon Netw Comm. 24 (2012)
4 E. Serpedin et al., Taylor & Francis Group (2012)
5 M. Resende, et al., Handbook of Metaheuristics (2010)
6 J. Gonçalves et al., J. Heuristics (2010)
7 A. Castro et al., Comp. Netwk. 56 (2012)
8 L. Velasco et al., Optics Express 21, (2013)
9 A. Castro et al., in Proc. DRCN (2013)