Algorithmic Aspects in Planning Fixed and Flexible Optical Networks With Emphasis on Linear Optimization and Heuristic Techniques

Emmanouel A. Varvarigos and Konstantinos Christodoulopoulos

(Invited Tutorial)

***Abstract*—From an algorithmic perspective, planning and operating optical networks falls in the broad category of network optimization problems. We give a short introduction on algorithmic techniques that can be used to solve network optimization problems, emphasizing on linear optimization and heuristics. We present examples of applying these techniques to optimize resource allocation during the planning of optical networks. In particular, we focus on fixed-grid WDM networks, which is the current practice, and on flexible optical networks, considered as the most promising architecture for meeting next generation core and metro network requirements. In doing so, we describe a generic problem definition that can capture both types of networks in a unified manner.**

***Index Terms*—Heuristics, integer linear programming (ILP), LP-relaxation, linear programming (LP), meta-heuristics, network optimization, routing and wavelength assignment, routing and spectrum allocation, static planning.**

I. Introduction

WAVELENGTH division multiplexing (WDM) is the most common architecture used for establishing communication in optical networks. In WDM networks data is transmitted over all-optical channels, called lightpaths, that may transparently pass intermediate switches and span over multiple consecutive fibers [1]. In WDM networks, the spectrum is divided into wavelengths of specific bandwidth (50 or 100 GHz) and establishing a lightpath requires the selection of a path and a wavelength on the links that comprise the path. WDM networks are widely deployed in core and metro area networks, but due to their wavelength-level coarse granularity and inflexibility in assigning resources, researchers are looking into alternative technologies to meet the requirements of next generation networks [2].

Manuscript received June 24, 2013; revised September 13, 2013 and November 7, 2013; accepted November 7, 2013. Date of publication November 13, 2013; date of current version January 10, 2014. This paper is based on the invited tutorial: E. Varvarigos: "An introduction to routing and wavelength assignment algorithms for fixed and flexgrid," Optical Fiber Communications Conference (OFC) 2013. This work has been partially supported by the European Commission through the IDEALIST project under Grant 317999.

The authors are with the Department of Computer Engineering and Informatics, University of Patras, and also with the Computer Technology Institute and Press "Diophantus," Patra 26500, Greece (e-mail: manos@ceid.upatras.gr; kchristodou@ceid.upatras.gr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JLT.2013.2290842

A technology that has received a lot of attention lately is *flexible* or *elastic* optical networking [3], [4]. Flexible optical networks assume the use of tunable transponders and a flexible spectrum grid or *flex-grid*. Flex-grid's granularity is much finer than that of standard WDM systems: the spectrum is divided into spectrum slots (12.5 GHz, as standardized by ITU G.694.1) that can be combined to create channels that are as wide as needed. Tunable optical transponders, also called bandwidth variable transponders (BVT) or software defined transponders [5], have lately been proposed, which are able to adapt several transmission parameters such as the modulation format, the spectrum they utilize, the transmission rate, etc.

Both in standard fixed-grid WDM and in flexible optical networks, establishing connections involves the solution of some sort of resource allocation problem, resources being the transponders, the regenerators, the links and the spectrum units (wavelengths or spectrum slots). In traditional WDM networks, resource allocation corresponds to the routing and wavelength assignment (RWA) problem, used to select the paths and wavelengths to create the lightpaths. Accounting for physical layer impairments (PLIs) and regeneration placement, traffic grooming, etc, are problems added to the basic RWA problem. In flexible optical networks the related problem is the routing and spectrum allocation (RSA). Apart from the difference in allocating spectrum resources, the choice of the transmission parameters of the tunable transponders present in flexible networks directly or indirectly affect the resource allocation decision. This is because the transmission configurations of the transponders interrelate the transmission reach with the spectrum used, the rate, and other parameters, and thus have to be included in the RSA problem. The multiple degrees of freedom present in flexible optical networks and their interdependencies make connection establishment in such networks (RSA) more complicated than in fixed-grid WDM networks (RWA).

Resource allocation is typically considered under two alternative traffic models. In the planning phase when the network is empty, connections are established for the traffic it is predicted to handle. In this case, the set of connection requests is assumed to be known in advance, for example given in the form of a traffic matrix, and the problem is referred to as *offline* or *static* resource allocation problem. In the operational phase, new connection requests arrive dynamically, and they have to be established upon their arrival or as a set, taking into account the current utilization state of the network, that is, the previously established connections. This is referred to as *online* or *dynamic* problem.

Offline and online resource allocation problems fall in the general category of network optimization problems. We will focus our study on the planning phase of fixed WDM and flexible optical network, namely the offline RWA and RSA, which are known to be NP-complete problems [6], [7]. Offline resource allocation is more *difficult* than online, since it aims at jointly optimizing the establishment of all the connections and minimizing the resources used, giving a more combinatorial character to the problem, in the same way that the multicommodity flow problem is more difficult than the shortest path problem in general networks.

Network optimization problems range from simple problems, such as shortest-path, max-flow, minimum spanning tree [9], etc, to more complicated ones, such as multicommodity integer flow, graph coloring, traveling salesman, etc. There are several ways and metrics to classify problems as being difficult or not and algorithms as being efficient or not. The most commonly used way to classify an algorithm as efficient is based on analyzing the worst–case number of arithmetic operations required during the execution of the algorithm [8]. According to this definition an algorithm is efficient if it can find the solution in polynomial time for any input, and a problem is provably difficult if it belongs to the class of NP-complete problems, for which no polynomial time algorithm is known. An algorithm runs in polynomial time if the number of elementary operations taken on any instance of the input and the space that it uses is bounded by a polynomial on the size of the input instance. This analysis considers all possible input instances and keeps the complexity of the worst one(s), which however might not be encountered in reality. On the contrary, average-case analysis targets to measure the performance of the algorithm for average realistic input, but suffers from the difficulty of formally defining the average case. Lately, smoothed analysis that combines the benefits of both worst- and average-case analysis has been introduced, and is considered a very realistic way to measure the practical performance of the algorithms [10].

The offline RWA and RSA resource allocation problems we focus on in this paper fall in the category of difficult problems (NP-complete), according to the worst-case analysis definition. However, algorithms that have good smoothed performance can be used to solve such problems with running time that is acceptable for realistic problem instances.

In this paper, we will present general algorithmic techniques that can be used to solve many network optimization problems. Given the large number of algorithms available in the literature for the RWA and, more recently, the RSA problems, in this paper we will first present general techniques and then give specific examples of the most used approaches to showcase how these can be applied to the problems at hand. In doing so, we will also give a problem definition and an algorithm that can solve the RWA and RSA planning problems in a unified manner. Due to the limited space, we will also avoid presenting results on the algorithms' performance, and rely on corresponding pointers to references when needed.

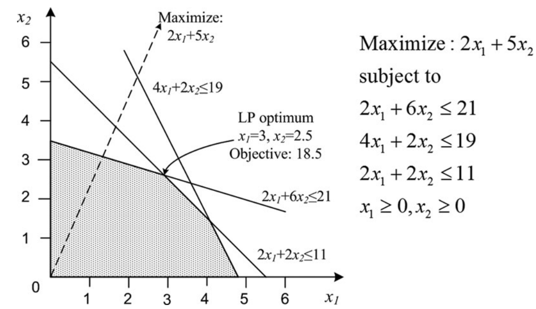

Fig. 1. An LP problem example.

II. Network Optimization

In this section we describe general optimization techniques that can be applied to solve resource allocation problems in both fixed and flexible optical networks.

A. General Optimization Problem

The general optimization problem is defined as follows:

$$\text{Minimize: } f_0(\boldsymbol{x})$$
$$\text{subject to } f_i(\boldsymbol{x}) \le b_i, i = 1, \ldots, m$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in R^n$ are the optimization variables, f_0: $R^n \rightarrow R$ is the optimization function, and f_i: $R^n \rightarrow R, i = 1, 2, \ldots, m$ are the constraint functions. Note that maximization problems can be transformed into the above form by minimizing the negative of the optimization function f_0.

Many optimization problems are hard to solve, but there are certain classes of problems for which we know efficient (polynomial) algorithms to find the optimal solution, such as linear optimization problems. For other classes that are provable hard, such as integer linear problems, we have sophisticated algorithms that are quite efficient and yield exact solutions for small and medium size problems and near-optimal solutions for large problems. In the following we focus on specific classes of optimization problems that can be used to formulate a large number of network optimization problems, including the offline RWA and RSA problems.

B. Linear Programming

The general linear programming (LP) optimization problem is defined as follows:

$$\text{minimize } \boldsymbol{c}^T \cdot \boldsymbol{x}$$
$$\text{subject to } \boldsymbol{A} \cdot \boldsymbol{x} \le \boldsymbol{b}, \boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in R^n$$

where $\boldsymbol{A}$ is a $m \times n$ matrix, and $\boldsymbol{c}$ and $\boldsymbol{b}$ are vectors of size n and *m,* respectively. Compared to the general optimization problem presented above, in LP problems the optimization function f_0 and the constraints $f_i, i = 1, \ldots, m$, are linear on $\boldsymbol{x}$.

Fig. 1 displays an example of a linear optimization problem with two variables and the corresponding geometrical representation. A linear equality constraint corresponds to a hyperplane (a line in the two dimensional space), and the set of constraints define the feasible region that is colored grey in the geometric representation. The optimal solution is the point (3, 2.5) that corresponds to a *vertex* (corner) of the feasible region. In the general n-dimensional optimization problem (n corresponds to the number of optimization variables, that is, the dimension of vector $\boldsymbol{x}$), the feasible region defined by the m constraints forms a convex n-dimensional polyhedron. Since the objective function is linear, all local optima are also global optima. In an LP problem, an optimal solution always exists at a vertex of the feasible region polyhedron [8] and this gives rise to a number of algorithms to solve such problem.

A naïve algorithm to solve an LP problem would enumerate all vertices (which are exponential on the number of constraints), calculate the objective value at those vertices and select the one with the optimal value. Note that because we can enumerate all feasible solutions we can classify LP problems as combinatorial optimization problems, but there are much more efficient ways to solve an LP. Simplex algorithm developed in 1947 by Dantzig, moves from vertex to vertex of the feasible region (n-dimensional polyhedron), going always to a vertex with a better or equal objective cost (higher or lower for maximization or minimization problems, respectively). Simplex moves from vertex to vertex according to what is called a *pivoting rule*, and its running time depends on the rule used. Sophisticated pivoting rules have been developed to avoid Simplex being stuck at so called degenerate vertices. However, for pivoting rules proposed until now it has been proved that they run in exponential time (one of the best results proves a rule to be sub-exponential [11]), and it is an open research issue if a pivoting rule can be devised that solves in polynomial time all problem instances. Although Simplex is considered an exponential time algorithm (worst case), until proven otherwise, it is very efficient for the majority of inputs, since it has good average and smoothed performance [10]. Algorithms such as the Ellipsoid and Interior Point that were later devised can provably solve any LP problem in polynomial time [8]. Still Simplex is used in the majority of cases due to its better average running time.

Many known network optimization problems can be formulated as LP problems and solved using Simplex or another LP algorithm. Fig. 2(a) and (b) display the shortest path and the multicommodity flow problems formulated as LPs. Importantly for the RWA and RSA problems, the multicommodity flow problem, which is closely related to them, becomes difficult to solve if we require the flows to be integer. We will discuss such problems in the next paragraph.

C. Integer Linear Programming

The general integer linear programming (ILP) optimization problem is defined as follows:

$$\text{Minimize: } \boldsymbol{c}^T \cdot \boldsymbol{x}$$
$$\text{subject to } \boldsymbol{A} \cdot \boldsymbol{x} \le \boldsymbol{b}, \quad \boldsymbol{x} = (x_1, \ldots, x_n) \in Z^n$$

where $\boldsymbol{A}$ is again a $m \times n$ matrix, and $\boldsymbol{c}$ and $\boldsymbol{b}$ are vectors of size n and m, respectively. The only difference from the general linear optimization problem of Section III-B is that variables $\boldsymbol{x}$ are now constrained to take integer, instead of real, values. Fig. 3 presents the corresponding ILP problem of Fig. 1.

An optimal solution is not guaranteed to exist among the polyhedron vertices anymore, as is the case for LP problems,

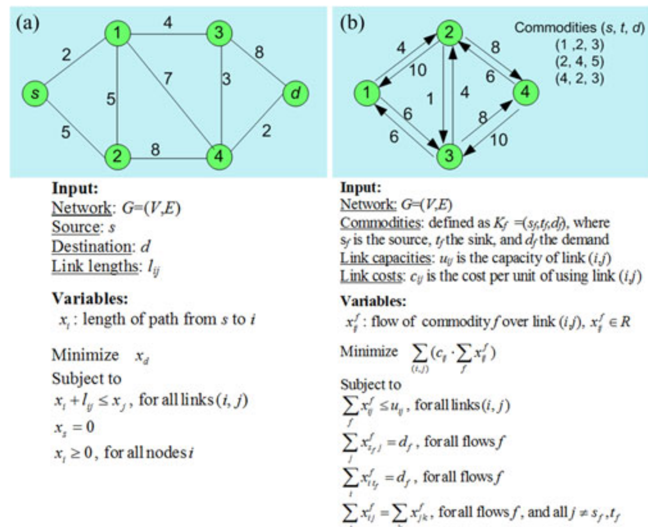

Fig. 2. (a) Single source-destination shortest path problem, and (b) multicommodity flow problem, writen as LP formulations.

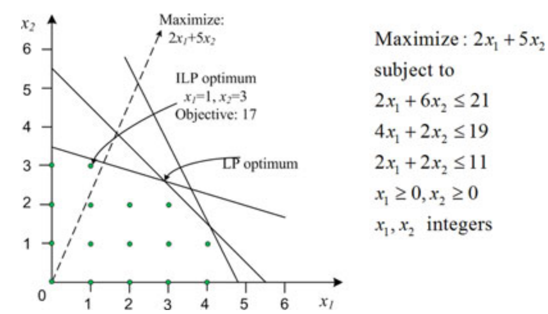

Fig. 3. ILP problem that corresponds to the example of Fig. 1.

and LP algorithms such as Simplex cannot be used to find an ILP optimal solution. The general ILP problem falls in the category of NP-complete problems and, up until today, there is no known efficient algorithm to solve any of these problems. If only some of the variables are required to be integer and others are real, the problem is called a mixed integer linear programming (MILP) problem, which is also NP-complete.

Since the number of values that the variables can take is finite (ILP problems are combinatorial optimization problems), we can again use a naïve algorithm to enumerate all feasible solutions, calculate the objective and select the optimal one. Branch-and-bound and cutting plane techniques can be used to improve average time performance and solve optimally small and medium size ILP problems. However, these techniques do not scale well, as their worst-case running time is exponential to the input size. The branch-and-bound technique is based on a sophisticated enumeration and pruning of solutions. It explores the solution space as a tree-like structure (branch) and stops the exploration of a certain solution subtree (bound), if this is infeasible or would not produce a better solution than the integer solution already found. The cutting-plane technique reduces the feasible solution space by adding constraints in a way that does not discard optimal integer solutions, and stops when an optimal solution is located at a vertex of the feasible region. More information about these techniques can be found in [8], [13].

Commercial ILP solvers, such as IBM CPLEX [12], utilize both techniques in so called branch-and-cut algorithms, combining their advantages. Such solvers can provide efficient solutions to small and medium sized ILP problems.

The RWA and RSA planning problems considered in this paper can be expressed as ILP problems. This is because the utilization variables for the resource entities (wavelengths or paths) take integer values; we cannot establish a lightpath with half a wavelength. Moreover, in the planning problem we want to optimize the network for all connections, and the allocation of integer resources to one connection affects the allocation of integer resources to the others, yielding a combinatorial optimization problem. Since ILP problems are hard, we cannot hope to find efficient algorithms that will give optimal solutions in all cases. In what follows we report on some theoretical tools that can help us in practice in this task.

D. Connection of ILP and LP

1) LP-Relaxation: Consider an ILP problem for which, as stated before, no known polynomial time algorithm exists. Instead of solving this ILP problem we can solve the same problem without constraining the variables to be integer (for example, instead of solving the problem presented in Fig. 3 we solve the problem of Fig. 1). The problem without the integer constraints is called the LP-relaxation of the ILP problem. As discussed previously, the relaxed LP problem can be solved efficiently using, e.g., Simplex or Interior Point algorithms.

Solving the related LP-relaxation problem can be quite beneficial. First of all, if the solution happens to be integer (all x_i take integer values), then we have found an optimal solution for the initial ILP problem. Although this might seem improbable (except for some special cases e.g., problems with totally unimodular matrix in which case all vertices are provably integer), there are certain techniques and rules to write ILP formulations that can increase this probability. We will discuss this more in the next section. Moreover, solving the LP-relaxation gives a lower or upper bound on the objective cost for the initial ILP problem, depending on whether we have a minimization or maximization problem, respectively. Indeed, if there was a better integer solution to the initial ILP problem, it would have been found, since it would also be an optimal solution for the LP-relaxation problem. The branch-and-bound technique can use the LP-relaxation to calculate the objective lower (or upper) bound of a subtree/branch of the solution tree. It considers these bounds in order to decide the search ordering and also stops the exploration of a certain branch if its calculated lower (upper) bound is higher than the best integer solution found up to that point. Finally, given a non-integer solution of the LP-relaxation problem, we can use rounding methods, such as randomized rounding [13], [14], to obtain approximate solutions for the initial ILP problem.

2) Convex Hull: Given an ILP optimization problem, the set of feasible integer solutions can be described by different sets of constraints. Having in mind the geometrical representation of the feasible region as an n-dimensional polyhedron we can visualize the same set of integer solutions to be included in different-shaped polyhedrons. Figs. 4(a) and (b) show an example of an ILP problem that has the same integer solutions but is described by different sets of linear constraints. The *convex hull* is the minimum convex set that includes all the integer solutions of the problem (see Fig. 4(c)), which is an n-dimensional polyhedron with integer vertices. If we could define the variables and/or write the constraints so that the polyhedron obtained is the same with the convex hull, then we could use LP algorithms, e.g. Simplex, to optimally solve the related ILP problem efficiently. However, the finding of the constraints to transform a general n-dimension polyhedron to the corresponding convex hull is difficult, and is the main idea behind the cutting plane techniques.

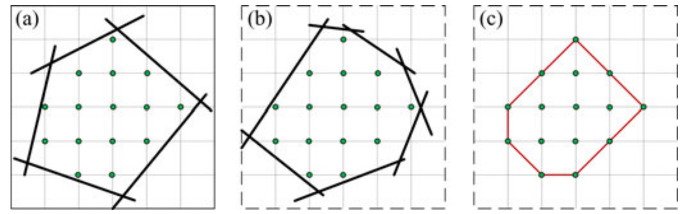

Fig. 4. Integer solution set and convex Hull.

3) Good ILP Formulations: Based on the above, we can intuitively say that an ILP formulation is "good" if the polyhedron defined by its linear constraints is close (tight) to its integer convex hull. This indicates that not all ILP formulations of a problem are equally good and there is considerable space for skill and luck [13]. As a general rule, adding more linearly independent constraints gives tighter formulations. Also formulating the problem with decision (Boolean) variables and avoiding integer variables and Big-M constraints can yield stronger formulations. In a tight polyhedron the solution of the LP-relaxation is close or equal to the integer optimum. Thus branch-and-cut techniques run fast and can help us solve quickly and efficiently the corresponding ILP problem. An ILP formulation with feasible region that is almost tight to the convex hull would have a large number of vertices with integer coordinates and would increase the probability of obtaining an integer optimal solution by solving the LP-relaxation. Note that branch-and-cut is an exact algorithm, meaning that given enough time it will always terminate with the optimal solution. Solving the LP-relaxation of a tight formulation may not yield the optimal solution, and randomized techniques such as perturbation or rounding can be used to get us to integer solutions. In Section III we present such a formulation to solve the offline RWA problem.

E. Heuristics, Meta-Heuristics

We already mentioned a number of ways to attack difficult problems, which rely on defining good formulations and using branch-and-cut or LP-relaxation with/without randomized rounding. The former case is an *exact* method, which comes with a proof that the optimal solution will be found in a finite (although often prohibitively large) amount of time. The latter is considered a *heuristic* approach.

A heuristic algorithm is not guaranteed to find optimal solutions but often gives near-optimal solutions at acceptable times, trading off optimality for speed. A heuristic is typically tailored to a problem and constructs the solution in a sophisticated manner that is problem-specific. Greedy heuristics is a class that is widely used in network optimization problems. A greedy heuristic, at each step of creating the solution, makes a decision that seems the best (towards a local optimum), that is, a decision that increases the optimization function the least or the most (depending on whether we have a minimization or a maximization problem), with the hope of finding a global optimum.

Meta-heuristics are general methods that, as opposed to problem-specific heuristics, can be used to solve different optimization problems [15]. A meta-heuristic is an iterative process that modifies or uses simpler heuristics to produce solutions beyond those that are normally generated in a quest for a local optimum. So it examines solutions produced by a heuristic and moves to better ones in a sophisticated manner. A meta-heuristic defines a representation or encoding of a solution, the cost function that maps the representation to the objective cost (using, e.g., a heuristic algorithm) and the iterative procedure that is used to obtain new solutions. Popular meta-heuristics include genetic/evolutionary algorithms, ant-colony, tabu search, simulated annealing and others. Other interesting techniques that have been used recently are the so called math-heuristics [16] that combine ILP exact algorithms with meta-heuristics. Given a solution of the problem, the math-heuristic performs a local search around it using an ILP formulation and solver, but because the solution space searched is limited around that solution this is done quite efficiently.

Static resource allocation problems, such as the offline RWA and RSA that involve the establishment of multiple connections simultaneously can be solved using heuristic and meta-heuristic algorithms. A quite intuitive approach is to use a sequential heuristic algorithm to establish connections one-by-one in a specific order, taking into account all previously established connections when processing a new one. Depending on the problem, the optimal resource allocation can be found if the connections are served in the right order, but the number of orderings grows exponentially and cannot be all checked. Meta-heuristics can be used to search among different orderings. We will see such an example in the last section of this paper, where we will outline how a sequential heuristic can be combined with simulated annealing meta-heuristic to search among different orderings and find better RSA solutions.

While heuristics in general give no guarantees as to their effectiveness, approximation algorithms [14] provide provable solution quality within provable run-time bounds; for example, the optimal solution is guaranteed to be, say, within a constant factor of the solution calculated by the approximation algorithm in polynomial time. Although approximation algorithms have been widely used to solve hard problems, their application in network planning is limited.

F. Optimization Objective, Single- and Multi-objective

An important issue when formulating resource allocation problems in optical networks is the choice of the optimization objective. The definition given at the start of this section corresponds to a *single objective* optimization problem. Most optical network planning problems consider a single metric to optimize, e.g. they seek to minimize the number or cost of transponders, or the number of wavelengths or spectrum used, or the energy consumption, etc. However, in many cases we are interested in more than one metric, such as both the number of transponders and the wavelengths used, or other combinations of the above metrics. In a *multi-objective* optimization problem, it is usually the case that no single solution simultaneously accomplishes the optimization of all metrics. Then we have what is called the non-dominated or Pareto front of solutions, which is the set of solutions that cannot be improved in one objective without deteriorating in at least one of the rest (see Fig. 5).

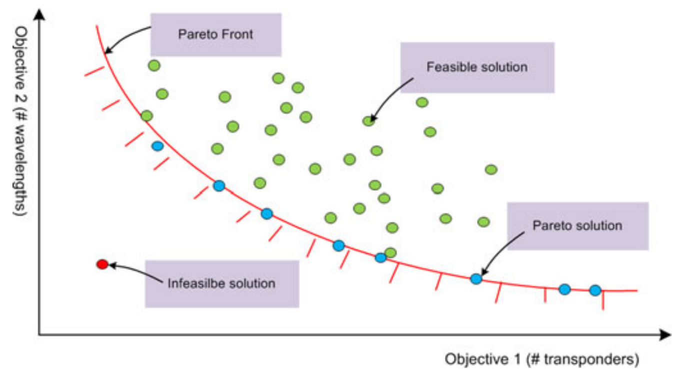

Fig. 5. The Pareto front of a multi-objective optimization problem.

Multi-objective optimization algorithms exist, though they are not efficient in most cases. Genetic or evolutionary algorithms are widely used to solve multi-objective problems. However, the most common approach is to *scalarize* the problem, by defining a single objective as a weighted combination of the multiple objectives, and use a single-objective method, such as the ones described in the previous sections, to solve it. For example, assuming a weighting coefficient w, we can formulate the multi-objective problem of minimizing the number of transponders N_t and the wavelengths N_w as follows:

$$\text{Minimize: } (w \cdot N_t) + [(1 - w) \cdot N_w)].$$

By ranging the coefficient w between 0 and 1, and using a single-objective optimization algorithm to solve the problem we can obtain the corresponding Pareto front.

III. Routing and Wavelenth Assignment Problem

In this section we apply some of the general techniques presented in the previous section to solve a basic version of the offline RWA problem in a fixed-grid WDM network. Note that similar techniques can be applied to solve the offline RSA problem in flexible networks. For example, the LP-relaxation formulation with piecewise linear cost function, as presented here, can be used in any multicommodity flow problem, and thus can be applied to the RSA problem as well. The same applies to random perturbation, and the iterative fixing and rounding techniques that we present here.

The switched entity in a fixed-grid WDM network is a *lightpath* that transparently passes intermediate nodes so that its signal remains in the optical domain from source to destination.

The basic RWA problem refers to the allocation of a path and a wavelength to each lightpath, without accounting for the grooming of traffic for these connections. In the end of this section we comment on extensions to this basic RWA problem that also take into account more realistic parameters, such as physical layer impairments, the placement of transponders and regenerators, grooming, and survivability.

The WDM network topology is represented by a connected graph $G = (V, E)$. V denotes the set of nodes, which are the optical switches, assumed not to be equipped with wavelength converters. E denotes the set of single-fiber links, each supporting W distinct wavelengths, $C = \{1, 2, \ldots, W\}$. The static version of RWA assumes a-priori known traffic, given in the form of an integer *traffic matrix* Λ. Let Λ_{sd} denote the number of requested lightpaths from source s to destination d.

In this basic example we will not consider reach constraints, assuming for now that lightpaths can be established over any path. The algorithm takes as input a specific RWA instance; that is, the network topology, the set of wavelengths, and the traffic matrix. It returns the RWA instance solution, in the form of lightpaths (paths and wavelengths). The goal is to find the solution that serves the traffic using the minimum number of wavelengths (if a solution cannot be found for the given number of wavelengths then we minimize the number of blocked ones – or find the minimum number of wavelengths to support the traffic; we omit such variations here). Lightpaths sharing a link cannot be assigned the same wavelength (distinct wavelength assignment constraint). Also a lightpath, in the absence of wavelength converters, must use the same wavelength on all its links (wavelength continuity constraint).

In what follows we will present an ILP formulation and then an algorithm that is based on an LP-relaxation formulation with appropriate techniques to obtain integer solutions to solve the above described basic RWA problem.

A. ILP RWA Formulation

The proposed ILP formulation uses the following variables:

1) $x_{l,w}^{s,d}$: a decision (Boolean) variable, equal to 1 if there exists a lightpath between node s and node d that uses wavelength w on link l, and equal to 0, otherwise.
2) *M:* the number of the maximum wavelength utilized in the network, it takes integer values between 0 and W.

Minimize: M

subject to the following constraints:

a) distinct wavelength assignment constraints,

$$\sum_{s,d} x_{l,w}^{s,d} \leq 1, \text{ for all } l \in E \text{ and all } w \in C$$

b) incoming flow constraints,

$$\sum_{l \in s^+} \sum_{w} x_{l,w}^{s,d} = \Lambda_{sd}, \text{ for all } (s,d) \in V^2$$

c) outgoing flow constraints,

$$\sum_{l \in d^-} \sum_{w} x_{l,w}^{s,d} = \Lambda_{sd}, \text{ for all} (s,d) \in V^2$$

d) wavelength continuity constraint (flow conservation)

$$\sum_{l \in n^+} x_{l,w}^{s,d} = \sum_{l \in n^-} x_{l,w}^{s,d}, \text{ for all } (s,d) \in V^2,$$
$$\text{all } n \in V \neq s, d, \text{ and all } w \in C$$

e) maximum wavelength utilization per link

$$M \geq w \cdot \sum_{s,d} x_{l,w}^{s,d}, \text{ for all } l \in E, \text{ and all } w \in C$$

f) integrality constraints

$$x_{l,w}^{s,d} \in \{0,1\}, \; M \in \{1, 2 \ldots, W\}$$

In the above formulation we denote by n^+ and by n^- the set of outgoing and incoming links of a node $n \in V$, respectively.

The preceding ILP formulation can be solved using commercial tools. Although the RWA problem is provably NP-complete and ILP solvers are not guaranteed to finish in polynomial time, such an approach scales well up to medium sized problems of a few tens of nodes and wavelengths. Complexity becomes prohibitive for more realistic problems in WDM networks, when we account for physical layer impairments, regenerator placement, grooming, etc. To describe such problems we need to add several ILP variables and constraints, which makes it computationally intractable except for small toy-size networks. In the next section we give an example of an RWA algorithm that is based on LP-relaxation and scales considerably better than the one presented here, while being able to find optimal solutions with high probability.

B. RWA LP-Relaxation-Based Algorithm

The proposed LP-relaxation RWA algorithm consists of four phases [21]. The first (pre-processing) phase computes a set of candidate paths to route the requested connections. RWA algorithms that do not use a set of predefined paths but allow routing over any feasible path, as the one described above, are also used in the literature. These algorithms are bound to give at least as good solutions as the algorithms that use pre-calculated paths, but employ a much higher number of variables and constraints and scale worse. In any case, the optimal solution can also be found with an RWA algorithm that uses pre-calculated paths, given a large enough set of paths. In particular, in our algorithm a set P_{sd} of k candidate paths is computed for each commodity pair $s{-}d$, and the set $P = \cup_{sd} P_{sd}$ is passed to the next phase of the algorithm. The pre-processing phase clearly takes polynomial time.

The second phase of the algorithm formulates the given RWA instance as a linear program (LP), without using integrality constraints. We start by trying to find a solution for a specific set of wavelengths $C' = \{1, 2, \ldots, W'\}$, where $W' \leq W$ (to identify W' we solve the LP-relaxation of the ILP algorithm presented in the previous section and round up to the closest integer—the optimal integer solution will use at least that many wavelengths). The defined LP formulation is solved using the Simplex algorithm that is generally efficient (is polynomial when considering average and smoothed complexity, which is what we experience in practice), and has additional advantages, as we will see, for the formulation at hand. If the solution returned by Simplex is not integer, the third phase uses iterative fixing and rounding techniques to obtain an integer solution. Note that a non integer solution is unacceptable, since a lightpath is not allowed to bifurcate between alternative paths or wavelength channels. Finally, if a solution is not found, implying that the problem is infeasible for the examined number of wavelengths W', phase 4 increases W' and goes back to the second phase, trying to solve the problem for a higher number of wavelengths. Phases 2 and 3 can be executed up to W-W' times, and each of these phases takes polynomial time. We now focus on the second phase of the algorithm.

1) RWA Formulation Using a Piecewise Linear Cost Function: The proposed LP formulation aims at minimizing the maximum number of distinct wavelengths used on network links. Let $F_l = f(w_l)$ denote the flow cost function, assumed to be an increasing function on the number of lightpaths w_l traversing link l (the actual function used is presented in the next section). The LP objective is to minimize the sum of all F_l values. The following variables are used:

Variables:

1) $x_{p,w}$: a decision (Boolean) variable, equal to 1 if path p occupies wavelength w, that is, if lightpath (p, w) is activated, and equal to 0, otherwise
2) F_l: the flow cost function value of link l

Minimize: $\sum_l F_l$

subject to the following constraints:

a) Distinct wavelength assignment constraints,

$$\sum_{\{p|l\in p\}} x_{p,w} \leq 1, \text{ for all } l \in E \text{ and all } w \in C'$$

b) Incoming traffic constraints,

$$\sum_{p\in P_{sd}} \sum_{w} x_{p,w} = \Lambda_{sd}, \text{ for all } (s,d) \in V^2$$

c) Flow cost function constraints,

$$F_l \geq f(w_l) = f\left(\sum_{\{p|l\in p\}} \sum_{w} x_{p,w}\right), \text{ for all } l \in E$$

d) The integrality constraints are relaxed to

$$0 \leq x_{p,w} \leq 1 \text{ for all } p \in P \text{ and all } w \in C'.$$

Note that the wavelength continuity is implicitly taken into account by the definition of the path-related variables.

2) Flow Cost Function: The variable F_l expresses the cost of congestion on link l, for a specific selection of lightpaths used. We choose F_l to be any increasing function $f(w_l)$ of the number of lightpaths $w_l = \sum_{\{p|l\in p\}} \sum_w x_{p,w}$ crossing link l. $F_l = f(w_l)$ is chosen to be strictly convex (instead of, e.g., linear), implying a greater degree of 'undesirability', when a link becomes highly congested. This is because it is preferable for overall network performance to serve an additional unit flow using several low-congested links than a close to saturation link.

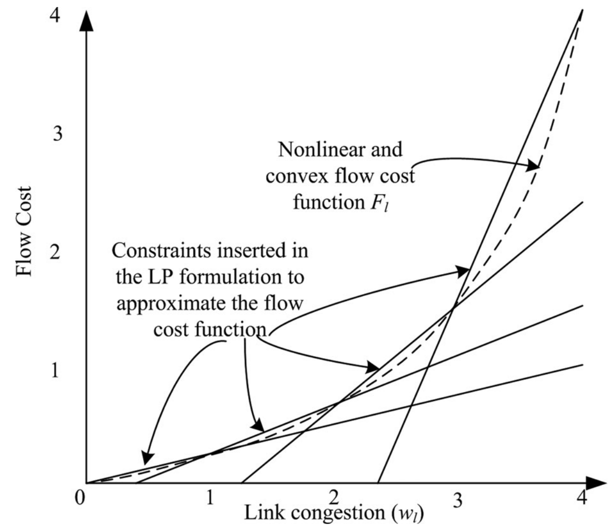

Fig. 6. The set of inequality constraints used in our LP formulation, limiting the search in the colored area. Since we minimize the flow cost we search the area borders, where one (or two) of the constraints is satisfied as equality. So, these constraints yield the piecewise linear approximation of the flow cost function F_l (dashed line).

In particular, we utilize the following flow cost function:

$$F_l = f(w_l) = \frac{w_l}{W' + 1 - w_l}, \ 0 \leq w_l \leq W'$$

This (nonlinear) function is inserted to the LP formulation in the approximate form of a piecewise linear function; i.e., a continuous function consisting of W' consecutive linear segments (Fig. 6). Since the LP objective is to minimize $\sum_l F_l$, one (or two if it is a vertex) of these W' linear cost functions is satisfied with equality at each link at the optimum, while all the remaining linear functions are de-activated, in the sense that they are satisfied as strict inequalities.

This piecewise linear function is equal to the nonlinear function $F_l = f(w_l)$ at integer argument values ($w_l = 1, 2, \ldots, W'$). Inserting such a piecewise linear function to the LP objective, results in the identification of integer optimal solutions by Simplex, in most cases [18]. This is because the vertices of the polyhedron defined by the constraints tend to correspond to vertices of the piecewise linear function and tend to consist also of integer components. Since the Simplex algorithm moves from vertex to vertex of that polyhedron, there is a higher probability of obtaining integer solutions than using other methods (e.g., Interior Point methods).

3) Random Perturbation and Iterative Fixings/Roundings: To improve the probability of our LP-formulation yielding integer solutions we devised a random perturbation technique. In the general multicommodity flow problem, given an optimal fractional solution, a flow that is served by more than one path has equal sum of first derivatives of the costs of the links comprising these paths [17]. In the RWA formulation at hand, a flow variable $x_{p,w}$ corresponds to a candidate lightpath (p, w), and the objective function sums the flow costs of the links comprising the lightpath. Thus, a request served by more than one lightpath has equal sums of first derivates over the links of these lightpaths. To increase the number of integer variables obtained we need to make the situations where two lightpaths have equal first derivative lengths less probable. To do so, we multiply the cost slopes of each link with a random number very close to 1, thus defining different slopes $a_{l,p,w}$ for each lightpath (p, w) and each link l.

If even with the randomly perturbed piecewise linear cost function presented above we do not obtain an integer solution we continue by "fixing" and "rounding" the variables. We start by fixing the variables; that is, we treat the variables that are integer as final, and solve the reduced problem for the remaining variables. Fixing variables does not change the objective cost returned by the LP, so we move with each fixing to a solution with equal or more integers with the same cost. Since the cost does not change, if we reach after successive fixings an all-integer solution, we are sure that it is an optimal one. Note, however, that fixing variables is not guaranteed to return an integer optimal solution, if one exists. If we reach a point beyond which the fixing process does not increase the integrality of the solution, we proceed to the rounding process. We round a single variable, the one closest to 1, and continue solving the reduced LP problem. While fixing variables helps us move to solutions that have more integer variables and the same objective cost, rounding makes us move to higher objective values and search for an integer solution there. However, if after rounding the objective cost changes, we are not sure anymore that we will end up with an optimal solution. Note that the maximum number of fixing and rounding iterations is the number of connection requests which is polynomial on the size of the problem input.

C. *Performance of the ILP and LP-relaxation Algorithm*

We implemented the LP-relaxation based RWA algorithm of Section III-B and compared it against the optimal ILP algorithm of Section III-A under realistic network settings. Our experiments showed that the LP-relaxation algorithm has superior overall performance, being able to find with very high probability an optimal solution with low execution times. The proposed random perturbation technique is a key component, since it increases its integrality performance. The good performance of the proposed algorithm is maintained even at heavy loads. The detailed comparison between these two algorithms can be found in [21].

D. *RWA Problem Extentions for Realistic Networks*

In previous sections we considered a basic version of the RWA problem, to showcase the way general optimization techniques (ILP and LP-relaxation with sophisticated features) can be used to solve a theoretically hard problem. We now turn our attention to more practical problems that arise when planning real WDM networks. The list of such problems is long and difficult to elaborate on within the page limitations, so we outline them and give appropriate references.

All problems discussed in the following include the basic RWA problem and as such they have to assign paths and wavelengths at some point. So it is crucial to have a good basic RWA algorithm to extend and account for the additional practical considerations. Depending on the types of variables used, RWA ILP formulations are classified as link-, path-, or maximal independent set (MSI)-based. Path- and MSI-based formulations use path-precalculation and might miss the optimal solution, if it uses a path not included in the precalculated set, but they scale much better than link formulations. Since longer hop paths use more resources (wavelengths on links) and also the wavelength continuity constraint makes lightpath establishment over longer hop paths more difficult, the optimal solution will in most cases use relatively short paths. So a k-shortest paths algorithm with k being small (e.g., 3—depending on the network size of course) yields optimal solutions in most cases, and even when optimality is missed, it will be very close. Writing elegant and sleek ILP formulations is still an active research topic, and recent results find the optimal solution of fairly large problems in very low running time [19]. Several approaches based on LP-relaxation and various heuristics have been also proposed [18], [20], [21]. Heuristics scale better than ILP formulations and are more appropriate to be extended to capture the additional problems that we will discuss in the following.

In WDM networks PLIs, such as noise, dispersion, crosstalk and other interference, affect the quality of transmission (QoT) of a lightpath [22]. Since a lightpath optically bypasses intermediate nodes, PLIs accumulate and after a point the transmission becomes infeasible. A threshold on the bit-error-ratio or some other QoT metric is used to classify a lightpath as acceptable or not. The interdependence between the physical and the network layers makes RWA in the presence of PLIs a cross-layer optimization problem, usually referred to as the impairment-aware (IA)-RWA problem. A large variety of IA-RWA algorithms exist, ranging from those considering a worst-case PLIs scenario, where the network is assumed fully utilized and calculate the worst case interference, to actual PLIs approaches, which go into details and calculate PLIs' effects for the true utilization state of the network. Clearly, there is a trade-off between complexity and efficiency between these two approaches. The former approach simplifies interdependencies, so that the IA and the RWA problem are decoupled and addressed separately. For example, the RWA algorithm presented in Section III-B can be used as an IA-RWA algorithm, provided that the paths passed to it are acceptable under worst case interference (to guarantee this, we calculate PLIs effects under full network utilization, and prune paths with unacceptable QoT). On the other hand, in the latter approach intelligent RWA can avoid interference and make use of paths that would be discarded by the former approach. To do so, one has to account for the lightpaths' interdependence through additional interference-related constraints in the RWA formulation, performing a true cross-layer optimization of the problem. This approach has a larger solution space and gives better performance, but at a higher complexity (e.g., the related ILP formulation requires substantially more variables and constraints). The reader is referred to [21] to see how the RWA LP-relaxation algorithm of Section III-B can be extended to account for the actual (as opposed to worst-case) interference among lightpaths.

The use of regenerators substantially improves the performance of WDM networks that span large geographical areas. A long distance connection can be regenerated at intermediate nodes and established in a multi-segment manner with acceptable QoT. Thus, regeneration is interrelated to PLIs, but also affects wavelength assignment, since regenerators act as wavelength converters. The presence of regenerators increases the complexity of the RWA problem, but also the efficiency of the solutions obtained. Algorithms ranging from ILP formulations to heuristics have been proposed to address it [22]–[24]. Another interesting problem is WDM network survivability [25], which also becomes more complicated in the presence of PLIs [26].

Optical WDM networks are typically deployed in core and metro areas and serve aggregated traffic at their ingress points. So, in addition to the physical layer, discussed in previous paragraphs, the RWA problem can be extended to consider the grooming of traffic at the edges as a way to further improve the efficiency of the WDM system. Traffic grooming used to involve the SONET/SDH sublayer [27], but today it is mostly performed directly at the IP layer, referred to as the IP-over-WDM problem. Traffic grooming can be viewed as a virtual topology routing problem over the WDM network and can either be solved jointly with RWA, or the two problems can be decoupled and solved individually but suboptimally [28].

To address the rapid increase of traffic, WDM systems target the deployment of higher rate and transmission distance connections, with the use of advanced modulation formats and coherent detection. The past few years have experienced a move from 10 to 40 Gbps transmissions, as technology matured and 40Gbps equipment prices became competitive. Even higher rate connections, e.g. 100 Gbps, will become economically viable in the near future. Thus, some WDM networks will end up managing a variety of line rates, what is usually referred to as a mixed-line-rate (MLR) WDM system. Connection establishment in a MLR WDM network is somewhat more complicated, since for each connection we have to choose among the different types of transponders [29], [30]. Accounting for PLIs seems more difficult on the one hand, since different models have to be used for each modulation format, but, on the other hand, coherent detection used at high rate transponders compensates for some of the PLIs, making the IA problem less important. Optimizing the migration of the network from a single to a mixed-line-rate involves the forecasting of traffic and equipment cost so as to deploy higher rate transponders at the right time [31].

IV. Routing and Spectrum Allocation

We now turn our attention to flexible optical networks and the related resource allocation problem, referred to as the Routing and Spectrum Allocation (RSA) problem.

Flexible optical networks are built using bandwidth variable switches whose granularity is lower than that of standard WDM systems, and they can combine the spectrum units - slots to create channels as wide as needed [3]. Moreover, flexible optical networks assume the use of flexible transponders that can adapt a number of transmission parameters, including the baud rate (symbols per second), the modulation format (number of bits encoded per symbol), the forward error correction (FEC) used, the spectrum employed, and the useful bit rate [5]. Therefore, establishing a connection in a flexible optical network involves the choice of the transponders' configurations (transmission parameters) [4] and the switches' configuration along the path to allocate the appropriate spectrum slots. Network protocols also have to be extended to support these additional features [32]. We refer to an all-optical connection as a *flexpath*, a variation of the word lightpath used in standard WDM networks. Flexpaths utilize as many spectrum slots as actually needed and thus the huge, but still limited, optical bandwidth is allocated in a more efficient manner than in standard WDM networks.

Taking a different approach than in previous sections, where we presented algorithms for the basic RWA problem (Section III-B) and then discussed extensions to them to address PLIs and other issues (Section III-D), in this section we will directly present an algorithm for planning a realistic flexible optical network. In doing so, we formally describe the problem in a way that is general and captures the specifications of both flexible and fixed-grid WDM systems, trying to unify the resource allocation problems in both settings. Moreover, in our problem definition we formulate the transponders' configurations, and show how the RSA problem can also incorporate such decisions. We chose to present here a heuristic/meta-heuristic algorithm, which combined with the ILP and LP techniques used in the previous section are the most commonly used techniques for planning fixed-grid and flexible optical networks.

In the network setting studied here, demands are served for their requested rates by choosing the transmission configuration of the transponders, selecting the paths, placing regenerators, if needed, and allocating spectrum to the defined flexpaths. The above decisions are interrelated, and even though we can jointly optimize over all of them by formulating the problem as an ILP, such an approach scales badly [35], even worse than in standard WDM networks. So the approach we pursue is to simplify the interdependence among the involved parameters, decoupling the problems, reducing the solution space, but without losing good solutions. In particular, the algorithm to be outlined considers PLIs and regeneration placement in the preprocessing phase when it creates candidate *path-transmission tuple pairs* to serve the demands. Then the main phase of the algorithm selects a path-transmission tuple pair for each demand and allocates spectrum to the resulting flexpaths.

The objective of the RSA algorithm is to find a solution that serves the traffic and optimizes the maximum spectrum and the cost of transponders used. Even though this is a multi-objective optimization problem, we use the scalarization method outlined in Section II-F, and define our objective cost as a weighted combination of the two metrics of interest, so as to convert it into a single objective problem.

A. A Generalized Approach to Account for the Physical Layer and Transponders Tunability

To model physical layer effects and the transponders' tunability in a flexible network we assume that a flexpath has a specific optical reach, defined as the distance it can transmit at with acceptable QoT. Optical reach depends on the flexpath's transmission configuration, the adjacent interfering flexpaths, and their transmission configurations and guardbands. The number

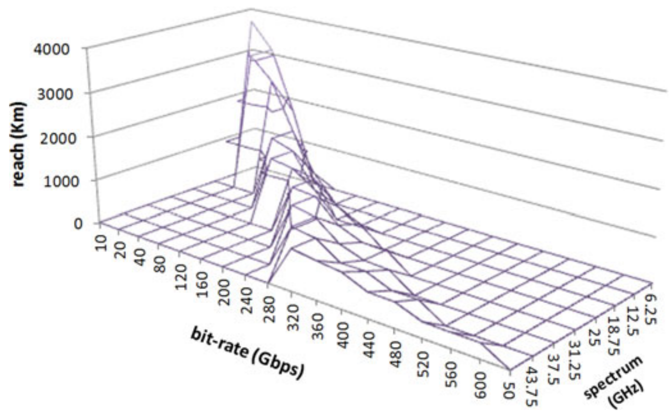

Fig. 7. Transmission reach as a function of the rate and spectrum used for a specific flex-grid transponder.

of configuration combinations can be huge; also, PLIs analytical models may not capture all effects, or experimental measurements may be limited to some options. So, the only viable solution seems to be some sort of simplification that captures PLIs in a coarse but safe manner, reducing the parameters and the solution space without eliminating good solutions. Note that calculating actual case interference for fixed grid networks is much simpler [21], since the parameters involved there are much fewer (non-tunable transponders).

A simplification we found fitting the above requirements is to calculate the transmission reach of a flexpath for a specific transmission configuration and guardband assuming worst case interference from the adjacent flexpaths. Specifically, assume that a flexible transponder of cost c can be tuned to transmit r Gbps using bandwidth of b spectrum slots and a guardband of g spectrum slots from its adjacent spectrum flexpaths to reach l km distance with acceptable QoT. This defines a *physical feasibility function* $l = f_c(r, b, g)$ that captures PLIs and can be obtained experimentally or using analytical models [32]. The guardband parameter is included, since increasing the space between flexpaths reduces interference and can increase the reach. Fig. 7 shows an example of a physical feasibility function without displaying (for illustration purposes) the guardband parameter g, assuming slot size of 6.25 GHz and a transmitter capable of transmitting up to 600 Gbps in 50 GHz. Note that defining the rate r and spectrum b incorporates the choice of the modulation format.

Using the functions f_c of the available flexible transponders we define (*reach-rate-spectrum-guardband-cost*) transmission *tuples*, corresponding to feasible configurations of the transponders. A feasible transmission tuple t is denoted by $(l_t, r_t, b_t, g_t, c_t)$. The term "feasible" is used to signify that the tuple definition incorporates PLI limitations, while the cost parameter is used when there are transponders of different capabilities and costs.

Note that the above definition is generic and can describe any type of flexible or even fixed-grid WDM optical network, with tunable or not transponders. For example, the mixed-line-rate (MLR) WDM network of [30] with 10/40/100 Gbps transponders can be represented with the following (*reach-rate-spectrum-guardband-cost*) transmission tuples:

- 10 Gbps: (1750 km, 10 Gbps, 50 GHz, 0 GHz, 1)
- 40 Gbps: (1800 km, 40 Gbps, 50 GHz, 0 GHz, 3)
- 100 Gbps: (900 km, 100 Gbps, 50 GHz, 0 GHz, 6),

where the last element corresponds to the cost, with the cost of the 10 Gbps transponder taken as the reference. A single-line-rate WDM network is even simpler and is described by a single tuple.

Using the above methodology we can enumerate the feasible transmission options, which incorporate in them the physical layer effects (PLIs), the tunability (or not) of the transponders and the different options for the type of transponders. It is then the role of the RSA algorithm to select the appropriate configuration among the available feasible ones to serve the traffic.

B. Pre-Processing Phase: Regenerator Placement and Generation of Candidate Solutions

The pre-processing phase computes the set of candidate *path-transmission tuple pairs* for serving the demands. The paths defined here include also the regeneration points. This is a symplifying assumption we made that limits the searched solution space and the optimal solution may be missed. Algorithms that do not assume a specific placement but include regenerator placement as part of the problem, or that calculate all possible regenerator placements, could also be devised, but would be computationally demanding and face scalability problems.

For each demand (s, d) we pre-calculate k paths, using a variation of the k-shortest path algorithm. Based on the link lengths of a candidate path p we identify the transponders' configurations (transmission tuples) that can be used over that path. In particular, for a transponder configuration, given as a (reach-rate-spectrum-guardband-cost) tuple $t = (l_t, r_t, b_t, g_t, c_t)$ to be acceptable, the reach l_t has to exceed the longest link of the path. For each acceptable path-transmission tuple pair (p, t), the path p is swept from left to right and a regenerator is placed whenever required, namely, at the last node before the transmission reach l_t is reached. With the placement of regenerators a *translucent* connection over path p is broken into sub-paths, each corresponding to a *transparent* flexpath. Let $R_{p,t}$ be the set of sub-paths comprising the translucent connection. If no regenerators are used, set $R_{p,t}$ contains one element (path p); otherwise, it contains paths $m_1, m_2, \ldots, m_{|Rp,t|}$, the first starting at source node s and ending at an intermediate regeneration node, the second starting at the previous regeneration node and ending at the destination or the next regeneration node, and so on, until destination d is reached.

For each path-transmission tuple pair (p, t) we calculate the cost of the transponders $C_{p,t}$, and the amount of spectrum $S_{p,t}$ required to serve the demand. These are the two metrics that we want to optimize in this planning problem. Given these we can remove path-transmission tuple pairs that will never be used in the solution. For a specific path p, suppose there is a tuple t that uses spectrum $S_{p,t}$ which is less than the spectrum $S_{p,t'}$ used by tuple t', and the transponders' cost $C_{p,t}$ is also less than the cost $C_{p,t'}$ of t'. Clearly, path-transmission tuple pair (p, t') cannot be part of the optimal solution, because we could always improve the solution by replacing t' with t. This is because we assume that the objective function to be minimized is monotonically increasing in each of these two parameters.

More formally, we will say that path-transmission tuple pair (p, t) *dominates* path-transmission tuple pair (p, t'), denoted as $(p, t) > (p, t')$, if the following holds:

$$(p, t) > (p, t') \text{ iff } C_{p,t} \leq C_{p,t'} \text{ and } S_{p,t} \leq S_{p,t'}$$

Dominated path-transmission tuple pairs are removed from the candidate solution space, since they will never be selected, reducing the solution space, without discarding good solutions, along with the execution time of the algorithms.

Based on the domination relation, we calculate for each demand (s, d) and each of its candidate paths $p \in P_{sd}$ the set of non-dominated path-transmission tuple pairs Q_{sd}. We pass these sets to the RSA algorithm, to be described next. We note here that this domination technique for pruning the search space and reduce complexity is a general technique used in other network optimization algorithms as well [34].

C. RSA Heuristic Algorithm—Sequential Serving of Traffic

Given as input the set of path-transmission tuple pairs, the RSA algorithm has to select one path-transmission tuple pair and allocate spectrum to it to serve each demand. Let C be the cost of the transponders and S the maximum number of slots used. The objective of the RSA algorithm is to find a solution that serves the traffic and optimizes both these metrics. We scalarize the multi-objective problem and optimize a weighted combination of these metrics, using the coefficient w:

$$\text{Minimize: } w \cdot S + (1 - w) \cdot C$$

We can formulate the RSA problem as an ILP and use an ILP commercial solver to obtain the solution [35]. Decision (Boolean) variables $x_{p,t}$ are used to choose the path-transmission tuple pair to serve each demand and starting frequency integer variables $f_{p,t}$ are used to formulate the spectrum allocation. Appropriate constraints have to be introduced to ensure the non-overlapping spectrum allocation.

Our experiments showed that the related ILP is tractable only for small problems, so in the following we present a heuristic approach that serves the demands sequentially, in a particular order. We keep track of the links spectrum utilization, updating them upon serving each demand. Since serving a demand is affected by previously made choices, the order in which the demands are considered plays an important role in the performance. We use simulated annealing meta-heuristic to search among different orderings and find better solutions.

The spectrum utilization of a link is represented by a three state vector, called the link slot utilization vector, with dimension equal to the maximum number F of slots supported in the system. A spectrum slot can be in one of the following states: (i) free (denoted by state u_f), (ii) used for data transmission (denoted by u_d), or (iii) used as guardband (denoted by u_g). The rules are that data slots cannot be used by new flexpaths, free slots can be used for data, while free and guardband slots can be used for guardband by new flexpaths. To enforce these rules, we calculate the slot utilization vector of a path using an (associative) 3-ary operator $\oplus$ for combining ("adding") the spectrum slots of the links that comprise it, defined as follows:

$$u_f \oplus u_d = u_d, u_f \oplus u_g = u_g, u_f \oplus u_f = u_f, u_g \oplus u_g = u_g, u_d \oplus u_d = u_d, u_d \oplus u_g = u_d$$

To serve a flexpath over a path using tuple $t = (l_t, r_t, b_t, g_t, c_t)$ requesting a specific number b_t of data spectrum slots and g_t guardband slots, we have to find b_t contiguous free (in state u_f) slots, and these slots need to have from each side g_t contiguous free or guardband (in state u_f or u_g) slots in the path utilization vector. Note, that we allow already assigned guardband slots to be used as guardband in a new flexpath, thus enabling the reuse of these slots, a feature that is useful when flexpaths require different amount of guardbands.

The sequential heuristic algorithm works as follows. We start with an empty network and all link utilization vectors initialized with u_f values for the slot states. We keep track of the cost C of transponders and the maximum number S of slots used by solutions obtained so far, initialized as $C = S = 0$. We serve the demands according to the ordering, examining for each of them all candidate path-transmission tuple pairs, decide the one to use, update the utilization vectors and costs, and move to the next demand. For a demand (s, d), for each path-transmission tuple pair (p, t) in Q_{sd}, and for each of its sub-paths m we compute its utilization vector, using the above equations to combine the utilization vectors of the links that comprise it. Then, we scan the sub-path utilization vector from left to right to find the first placement that can serve the flexpath. Spectrum continuity and non-overlapping spectrum allocation constraints are enforced over the sub-path by the definition of the slot addition among the links. After finding slots for all sub-paths of the path-transmission tuple pair (p, t) we calculate a temporary cost. We then move to examine the next path-transmission tuple pair in Q_{sd}, calculate the slots and cost of this option, and so on. Branch-and-bound techniques (stop examining path-transmission tuple pairs that have worse performance that the best found up to that point) speed-up the searching process. After doing this for all candidate path-transmission tuple pairs $(p, t) \in Q_{sd}$ we select the one that minimizes the objective. We establish the flexpaths, update the link utilization vectors and move to the next demand.

The algorithm finishes when it has served all demands, returning the final objective, the routes and spectrum allocated to each one. What is particularly important is that apart from the RSA solution the algorithm also chooses the transponders' configurations by selecting the transmission tuple to be used by each, and thus there two problems are jointly solved.

Since the algorithm's performance depends on the order in which demands are considered, we use simulated annealing meta-heuristic to find good orderings. We start with the highest demand first ordering. Neighboring orderings are created by interchanging two demands randomly chosen from a uniform distribution, and the sequential heuristic is run for these orderings and calculates the objective costs. Standard simulated annealing process [15] using the iteration count as stop criterion is employed. The solution of the ordering that yields the lowest objective is finally selected.

For a given ordering the sequential heuristic algorithm has to consider, in the worst case, for each demand all the non-dominated path-transmission tuple pairs, and find for each of them a valid spectrum allocation by manipulating the link and path utilization vectors. The number of operations performed depends linearly on the number of non-dominated path-transmission tuple pairs, the number of connections to be established, the number of links, and the size of the utilization vectors, and thus the proposed heuristic is polynomial to the size of the input. Simulated annealing is used so as to search among different orderings, remaining polynomial and being able to trade-off performance for running time by controlling the number of iterations (different orderings examined).

Note that the RSA algorithm outlined above can be also used for planning fixed-grid WDM networks. The definition of the path-transmission tuples is generic and can capture the specificities of fixed- and flex-grid systems with tunable or not transponders, of a single or more types, unifying in this way the static planning problem of these systems.

D. Performance of Proposed RSA Algorithm

We compared the above described heuristic algorithm to an ILP formulation using a commercial ILP solver to find optimal solutions for small scale problems [35]. Our results showed that the heuristic had very good performance in affordable running times. An important feature of simulated annealing is the trade-off it provides between the performance and the running time, by controlling the number of orderings that are examined.

As stated above, the proposed problem definition and algorithm are generic and can be used to optimize connection establishment in both flexible and fixed-grid WDM networks. We used this algorithm to compare the performance of a flexible and a MLR WDM network and observed that the former has significant performance benefits over the latter. We considered two optimization parameters, the maximum spectrum and the cost of transponders used. Defining the optimization objective as a weighted sum of these parameters and using the proposed heuristic algorithms we obtained the Pareto front of the solutions. We observed that in flexible optical networks with tunable transponders there is an interesting trade-off between these two parameters, so one can plan the network and save on transponders by using more spectrum, or the opposite.

E. Variations and Extensions of the RSA Problem

The spectrum allocation process in flexible optical networks is slightly more complicated than in traditional WDM networks. The difference is that the spectrum slots in flexible optical networks can be combined, while in WDM there is a one-to-one allocation of a spectrum unit (wavelength) to a single connection (lightpath). Spectrum allocation in flexible networks can be formulated in different ways. Assuming an ILP formulation, we can (i) define per slot decision (Boolean) variables similar to those used in the RWA problem and use additional constraints to regulate how slots are combined, or (ii) use starting frequency integer variables combined with appropriate non-overlapping spectrum assignment constraints [21], or (iii) precalculate all contiguous slot combinations for a connection and form decision variables with them [37]. The advantage of the second approach is that the number of variables is independent of the number of spectrum slots, and it can also formulate resource allocation even in gridless (as opposed to flex-grid) networks. The third approach is also very appealing, since although it results in a relatively high number of Boolean variables, these are easier to handle, as ILP solvers are optimized for such variables.

In flexible optical networks, powerful transponders that can be *sliced* and shared to serve more than one demand have started to be prototyped. These transponders are referred to as sliceable bandwidth variable transponders (S-BVT) or multiflow transponders. In a system with such transponders we need to account for the assignment of multiple flexpaths to transponders, requiring additional constraints to be formulated [39]. Thus, instead of point-to-point transmissions, algorithms that consider point-to-multipoint transmissions and tree-like routing have to be developed, and ideas from multi-cast routing could be used.

Researchers have started to examine the problem of jointly optimizing the IP layer and the flexible optical network to improve the overall performance. An ILP algorithm and a greedy randomized search meta-heuristic were presented in [40]. The IP-over-flexible networks problem is quite more complicated than the related problem in fixed-grid WDM networks, since by grooming traffic we change the transmission rate, which in turn interrelates to other transmission parameters of the tunable transponders. Grooming at the optical layer is also an interesting technology, and an ILP and a heuristic algorithm are presented in [41].

The RSA problem and its extensions, such as accounting for the PLIs, regenerator placement, grooming and survivability have to be considered in tandem with the choice of the transmission configurations of the transponders. Some of the algorithms in the literature do not consider transmission reach limitations [36]–[38] or they incorporate simple reach-modulation format constraints to capture physical layer effects [4], [21], [40]. The algorithm outlined in the previous sections, and extensively presented in [35], was designed with these issues in mind and jointly addresses the RSA, the choice of the transponders' configurations and the regeneration placement problems. Extensions to account for the IP layer, survivability and the use of sliceable transponders are underway.

Planning algorithms are expected to increasingly incorporate the choice of the transponders' configuration, as the tunable transponders find their way to the market. Once proper formulation of the transponders' configurations and capabilities are defined and incorporated in the various versions of the RSA planning problem, it will be possible to use traditional algorithmic techniques, such as ILP, heuristic/meta-heuristics, as discussed in Sections II, III, and IV, to solve them.

V. Conclusion

We outlined general techniques for solving optical network planning problems, emphasizing on linear optimization and heuristics. We then focused on the planning problem of fixed-grid and flexible optical networks, discussed their particularities, and presented algorithms to solve specific versions of these problems. The presented techniques can be applied to both network settings, and were chosen so as to span over different approaches widely used in the field. In particular, for the routing and wavelength assignment (RWA) problem in WDM networks, we presented an ILP formulation and an LP-relaxation algorithm that uses a piece-wise linear cost function, a random perturbation and iterative fixing and rounding techniques to obtain optimal integer solutions with high probability. We also presented a heuristic algorithm for the routing and spectrum allocation (RSA) problem in flexible optical networks. The algorithm captures the different configuration options for the flexible transponders and the effect of the physical layer in a preprocessing phase to calculate feasible options for serving the demands. A heuristic algorithm that serves demands in a particular order was also presented, with a simulated annealing meta-heuristic used to examine different orderings and improve the solution.

REFERENCES

[1] T. Stern, G. Ellinas, and K. Bala, *Multiwavelength Optical Networks: Architectures, Design and Control*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[2] A. A. M. Saleh and J. M. Simmons, "All-Optical networking, evolution, benefits, challenges, and future vision," *Proc. IEEE*, vol. 100, no. 5, pp. 1105–1117, May 2012.

[3] O. Gerstel, M. Jinno, A. Lord, and S. J. Ben Yoo, "Elastic optical networking: A new dawn for the optical layer?" *IEEE Commun. Mag.*, vol. 50, no. 2, pp. s12–s20, Feb. 2012.

[4] M. Jinno, B. Kozicki, H Takara, A. Watanabe, Y. Sone, T. Tanaka, and A. Hirano, "Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network," *IEEE Commun. Mag.*, vol. 48, no. 8, pp. 138–145, Aug. 2010.

[5] B. Teipen, M. Eiselt, K. Grobe, and J. P. Elbers, "Adaptive data rates for flexible transceivers in optical networks," *J. Netw.*, vol. 7, no. 5, pp. 776–782, May 2012.

[6] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high-bandwidth optical WANs," *IEEE Trans. Commun.*, vol. 40, no. 7, pp. 1171–1182, Jul. 1992.

[7] K. Christodoulopoulos, I. Tomkos, and E. A. Varvarigos, "Elastic bandwidth allocation in flexible OFDM-Based optical networks," *J. Lightw. Technol.*, vol. 29, no. 9, pp. 1354–1366, May 2011.

[8] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. New York, NY, USA: Dover, 1998.

[9] R. Ahuja, T. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, NJ, USA: Prentice Hall, 1993.

[10] D. Spielman and S. H. Teng, "Smoothed analysis: An attempt to explain the behavior of algorithms in practise," *Commun. ACM*, vol. 52, no. 10, pp. 76–84, 2009.

[11] O. Friedmann, "A subexponential lower bound for Zadeh's pivoting rule for solving linear programs and games," in *Proc. Conf. Integer Program. Combinatorial Optim.*, 2011, vol. 6655, pp. 192–206.

[12] IBM CPLEX. (2013). [Online]. Available. www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

[13] D. Bertsimas and R. Weismantel, "Optimization over integers," Charlestown, MA, USA: Dynamic Ideas, 2005.

[14] V. V. Vazirani, *Approximation Algorithms*. New York, NY, USA: Springer, 2001.

[15] M. Gendreau and J.-Y. Potvin, Ed., *Handbook of Metaheuristics*, 2nd ed. New York, NY, USA: Springer, 2012.

[16] J. Puchinger, G. R. Raidl, and S. Pirkwieser, "MetaBoosting: enhancing integer programming techniques by metaheuristics," *Matheuristics, Annals Inform. Syst.*, vol. 10, pp. 71–102, 2010.

[17] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice–Hall, 1992.

[18] A. Ozdaglar and D. Bertsekas, "Routing and wavelength assignment in optical networks," *IEEE/ACM Trans. Netw.*, vol. 11, no. 2, pp. 259–272, Apr. 2003.

[19] Z. Liu and G. N. Rouskas, "Link selection algorithms for link-based ILPs and applications to RWA in mesh networks," in *Proc. Conf. Opt. Netw. Design Modeling*, 2013, pp. 59–64.

[20] R. M. Krishnaswamy and K. N. Sivarajan, "Design of logical topologies: A linear formulation for wavelength-routed optical networks with no wavelength changers," *IEEE/ACM Trans. Netw.*, vol. 9, no. 2, pp. 186–198, Apr. 2001.

[21] K. Christodoulopoulos, K. Manousakis, and E. Varvarigos, "Offline routing and wavelength assignment in transparent WDM networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 5, pp. 1557–1570, Oct. 2010.

[22] J. S. Pareta, S. Subramaniam, D. Careglio, and S. Spadaro, "Cross-layer approaches for planning and operating impairment-aware optical networks," *Proc. IEEE*, vol. 100, no. 5, pp. 1118–1129, May. 2012.

[23] S. Pachnicke, T. Paschenda, and P. M. Krummrich, "Physical impairment based regenerator placement and routing in translucent optical networks," in *Proc. Opt. Fiber Commun. Conf.*, 2008, 1–3, paper OWA2.

[24] B. Garcia-Manrubia, P. Pavon-Marino, R. Aparicio-Pardo, M. Klinkowski, and D. Careglio, "Offline impairment-aware rwa and regenerator placement in translucent optical networks," *Lightw. Technol.*, vol. 29, no. 3, pp. 265–227, 2011.

[25] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," *J. Lightw. Technol.*, vol. 21, no. 4, pp. 2023–2030, 2003.

[26] S. Azodolmolky, M. Klinkowski, Y. Pointurier, M. Angelou, D. Careglio, J. Solé-Pareta, and I. Tomkos, "A novel offline physical layer impairments aware RWA algorithm with dedicated path protection consideration," *J. Lightw. Technol.*, vol. 28, no. 20, pp. 3029–3040, 2010.

[27] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks," *IEEE/ACM Trans. Netw.*, vol. 11, no. 2, pp. 285–299, 2003.

[28] R. Dutta, A. Kamal, and G. N. Rouskas, *Traffic Grooming for Optical Networks: Foundations, Techniques and Frontiers*. New York, NY, USA: Springer, 2008.

[29] A. Nag, M. Tornatore, and B. Mukherjee, "Optical network design with mixed line rates and multiple modulation formats," *J. Lightw. Technol.*, vol. 28, no. 4, pp. 466–475, 2010.

[30] O. Rival and A. Morea, "Cost-efficiency of mixed 10–40–100 Gb/s networks and elastic optical networks," in *Proc. Opt. Fiber Commun. Conf.*, 2011, pp. 1–3.

[31] C. Meusburger, D. A. Schupke, and A. Lord, "Optimizing the migration of channels with higher bitrates," *J. Lightw. Technol.*, vol. 28, no. 4, pp. 5002–5009, 2010.

[32] F. Cugini, G. Meloni, R. Paolucci, N. Sambo, M. Secondini, L. Gerardi, L. Potì, and P. Castoldi, "Demonstration of flexible optical network based on path computation element," *IEEE/OSA J. Lightw. Technol.*, vol. 30, no. 5, pp. 727–733, 2012.

[33] A. Klekamp, R. Dischler, and R. Buchali, "Limits of spectral efficiency and transmission reach of Optical-OFDM superchannels for adaptive networks," *IEEE Photonics Technol. Lett.*, vol. 23, no. 20, pp. 1526–1528, Oct. 2011.

[34] P. V. Mieghem and F. A. Kuipers, "Concepts of exact QoS routing algorithms," *IEEE/ACM Trans. Netw.*, vol. 12, no. 5, pp. 861–864, Oct. 2004.

[35] K. Christodoulopoulos, P. Soumplis, and E. Varvarigos, "Planning flexgrid optical networks under physical layer constraints," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 5, no. 11, pp. 1296–1312, 2013.

[36] Y. Wang, X. Cao, and Y. Pan, "A study of the routing and spectrum allocation problem in the spectrum-sliced elastic optical path networks," in *Proc. IEEE INFOCOM*, 2011, pp. 1053–1051.

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

[38] N. Patel, P. N. Ji, J. P. Jue, and T. Wang, "Routing, wavelength assignment, and spectrum allocation algorithms in transparent flexible optical WDM networks," *Opt. Switching Netw.*, vol. 9, no. 3, pp. 191–203, 2012.

[39] T. Tanaka, A. Hirano, and M. Jinno, "Performance evaluation of elastic optical networks with multi-flow optical transponders," in *Proc. Eur. Conf. Opt. Commun.*, 2012, paper Tu.3.D.2.

[40] O. Pedrola, A. Castro, L. Velasco, M. Ruiz, J. P. Fernández-Palacios, and D. Careglio, "CAPEX study for a multilayer IP/MPLS-Over-Flexgrid optical network," *J. Opt. Commun. Netw.*, vol. 4, no. 8, pp. 639–650, 2012.

[41] G. Zhang, M. De Leenheer, and B. Mukherjee, "Optical traffic grooming in OFDM-Based elastic optical networks [invited]," *J. Opt. Commun.*, vol. 4, no. 11, pp. B17–B25, 2012.

Authors' biographies not available at the time of publication.