Algorithmic Aspects of Optical Network Design

(Invited Paper)

E. A. Varvarigos and K. Christodoulopoulos

Computer Engineering and Informatics Department, University of Patras, Greece, and
Research Academic Computer Technology Institute, Patra, Greece
{*manos*, *kchristodou*}@ *ceid.upatras.gr*

Abstract—**Optical network design problems fall in the broad category of network optimization problems. We give a short introduction on network optimization and general algorithmic techniques that can be used to solve complex and difficult network design problems. We apply these techniques to address the static Routing and Wavelength Assignment problem that is related to planning phase of a WDM optical network. We present simulation result to evaluate the performance of the proposed algorithmic solution.**

Keywords- ***Network optimization, Linear Programming (LP), Integer Linear Programming (ILP), LP-relaxation, Routing and Wavelenth Assignment***

I. Introduction (Heading 1)

Wavelength routed WDM is the most common architecture used for establishing communication in optical transport networks [1]. In wavelength routed WDM, data is transmitted through lightpaths; that is, all-optical WDM channels that may span multiple consecutive fibers. From the network perspective, establishing a lightpath for a new connection requires the selection of a route (path) and a free wavelength on the links that comprise the path. Since the lightpaths are the basic switched entities of a wavelength routed WDM network, their effective establishment and usage is crucial. Thus, it is important to propose efficient algorithms to select the routes for the requested connections and to assign wavelengths on each of the links along these routes, so as to optimize a certain performance metric. This is known as the *routing and wavelength assignment* (abbreviated RWA) problem. The constraints are that paths that share common links are not assigned the same wavelength (*distinct wavelength assignment*). Also a lightpath, in the absence of wavelength converters, must be assigned a common wavelength on all the links it traverses (*wavelength continuity constraint*).

The RWA problem is usually considered under two alternative traffic models. When the set of connection requests is known in advance (for example, given in the form of a static traffic matrix) the problem is referred to as *offline* or *static* RWA. In this case the network is configured based on the traffic it is predicted to handle and thus this phase is related to the planning phase of the WDM network. In the operational phase, new lightpath requests arrive dynamically, at random times, and they have to be established upon their arrival, one by one, taking into account the current utilization state of the network, that is, the previously established lightpaths. This problem is referred to as *online* or *dynamic* RWA.

Offline and online RWA problems fall in the general and broad category of network optimization problems. We will focus our study on the planning phase of the WDM network, on offline RWA, which is known to be a NP-hard problem [2]. Offline RWA is more difficult than online RWA, since it aims at jointly optimizing the lightpaths used by the connections, in the same way that the multicommodity integer flow problem is more difficult than the shortest path problem in general networks. Network optimization problems ranges from simple problems such as shortest-path, max-flow, minimum spanning tree, etc. up to more complicated problems, such as multicommodity integer flow, graph coloring, traveling salesman, etc. With respect to optical networks, apart from the typical and pure RWA problem [3], there are also a number of other optimization problems, such as the traffic grooming problem [4], the impairment-aware RWA problem [5], the time scheduling of connections problem [6], and many more.

For a large number of network optimization problems there have been developed specific algorithms that can solve efficiently the specific problem under study [7]. However, for the more complex network optimization problems it is rather difficult to find and develop good algorithms. In those cases a solution can be found by using the general algorithms that have been developed in optimization theory.

In this paper we will present some general algorithms and techniques that can be used to solve a large number of network optimization problems and in particular we will try to focus on complex and difficult problems. Then we will apply these techniques to solve the planning problem of a WDM network, that is, we will address the offline RWA problem.

II. Network Optimization

A. General Optimization Problem

The general optimization problem is defined as follows:

$$\text{minimize (or maximize)} \quad f_0(\boldsymbol{x})$$
$$\text{subject to} \quad f_i(\boldsymbol{x}) \le b_i,\ i = 1,\dots,m,$$

where $\boldsymbol{x} = (x_1,\dots,x_n) \in \mathrm{R}^n$ are the optimization variables, $f_0 : \mathrm{R}^n \rightarrow \mathrm{R}$ is the optimization function, and $f_i : \mathrm{R}^n \rightarrow \mathrm{R}$, $i = 1,\dots,m$ are the constraint functions.

In general, optimization problems are hard to solve, but there are certain problems that we know algorithms that can be used to find the optimal solution in efficient time. Also, even for some hard problems we know good algorithms that can help us to optimally solve small to medium size problems. Note that, in the context of this paper we will assume that an algorithm is efficient if it can find the solution in polynomial time. We say that an algorithm runs in polynomial time if the number of elementary steps taken by the algorithm on any instance I of the input is bounded by a polynomial on the size of I. On the contrary, we will assume that algorithms that run in exponential time on the size of the input are not efficient. A problem is provably difficult if it belongs to the class of NP-complete problems, for which no polynomial time algorithms are known.

In the following we will examine some specific cases of the general class of optimization problem that can be used to formulate a large number of network optimization problems.

We will comment on the algorithms and techniques that can be used to solve these specific classes of problems and we will then apply these techniques and present our solution to the offline routing and wavelength assignment (RWA) problem.

B. *Linear Programming*

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

$$\text{minimize (or maximize)} \quad c^{T} \cdot \boldsymbol{x}$$

$$\text{subject to} \quad A \cdot \boldsymbol{x} \leq b,\ \boldsymbol{x} = (x_1,\ldots,x_n) \in \mathrm{R}^n,$$

where A is a mxn matrix, and c and b are vectors of size n and m respectively. With respect to the general optimization problem presented in Section II.A we can see that optimization function f_0 and constraints f_i, i=1,…,m are all linear on $\boldsymbol{x}$.

Figure 1 displays an example of a linear optimization problem with two variables and the corresponding geometrical representation. A linear constraint corresponds to a line in the two dimensional plain, and the set of constraints define the feasible region that is colored grey in the geometric representation. The optimal solution is the point (2.5,2.5) that corresponds to a vertex 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 forms a convex n-dimensional polyhedron. Since the objective function is also linear, all local optima are automatically global optima. The linear objective function also implies that an optimal solution can only occur at a boundary point of the feasible region, so there is an optimal solution on a vertex of the feasible region polyhedron [9].

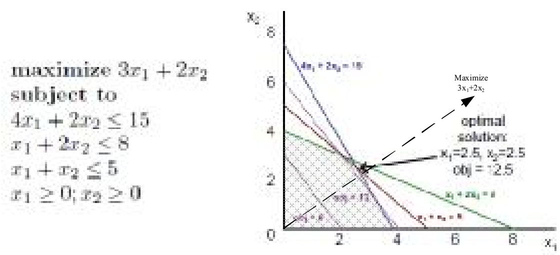
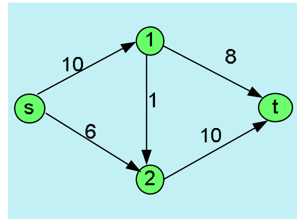


Figure 1. A LP problem example.

A number of algorithms have been developed to solve the general LP problem. Simplex algorithm developed in 1947 by Dantzig, moves from vertex to vertex of the feasible region (the n-dimensional polyhedron), going always to a vertex with a better or equal objective cost (higher or lower for maximization or minimization problems, respectively), and certain extensions have been developed to avoid circles. Although Simplex algorithm is efficient for the majority of inputs, problems have been constructed that Simplex takes exponential time to solve. Algorithms such as the Elliptic and Interior Point that were later developed can provably solve in polynomial time any LP input instance, that is, even in the worst case their running time is polynomial bounded on the size of the input. Still Simplex is used in many cases since it has good average running times, while at certain cases Inferior-Point is also very efficient.

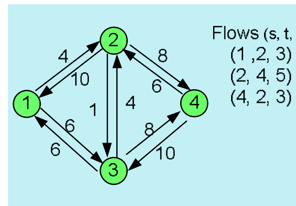
Many known network optimization problems can be formulated as LP problems and solved using Simplex or another LP algorithm. Figures 2 and 3 display two such examples. Note that the second example that presents the multicommodity flow problem becomes difficult to solve if we require the flows that are used to be integers. We will address such problems in the next paragraph.



Input: Demand (s,t), links capacities u_{ij}
Variables: x_{ij} flow over link (i,j)

Maximize v
Subject to
$\sum_j x_{sj} = v$
$\sum_j x_{ij} - \sum_j x_{ji} = 0$, for all $i \neq s$ or t
$\sum_i x_{it} = -v$
$0 \leq x_{ij} \leq u_{ij}$, for all links (i, j)

Figure 2. Maximum flow problem.



Input: Demand flows $f \rightarrow (s_f,t_f,d_f)$, links capacities u_{ij}
Variables: x_{ij}^f flow of flow demand f over link (i,j)

Minimize 0
Subject to
$\sum_f x_{ij}^f \leq u_{ij}$ for all links (i, j)
$\sum_j x_{s_f j}^f = d_f$ for all flows f
$\sum_i x_{i t_f}^f = -d_f$ for all flows f
$\sum_i x_{ij}^f = \sum_k x_{jk}^f$ for all $j \neq s_f, t_f$

Figure 3. Multicommodity flow problem.

C. *Integer Linear Programming*

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

$$\text{minimize} \quad c^{T} \cdot \boldsymbol{x}$$

$$\text{subject to} \quad A \cdot \boldsymbol{x} \leq b,\ \mathbf{x} = (x_1,\ldots,x_n) \in Z^n,$$

where A is a mxn matrix, and c and b are vectors of size n and m respectively. Compared to the general linear optimization problem presented in Section III.B we can see that the only difference is that the variables $\boldsymbol{x}$ are now constrained to take only integer values, instead of real values. Figure 4 presents the corresponding ILP problem of Figure 1.

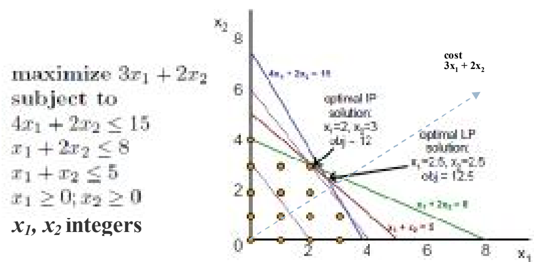


Figure 4. ILP problem that corresponds to the example of Figure 1.

The general ILP problem is difficult to solve. The optimal solutions are not anymore at the vertices of the feasible region, as is the case for LP problems, and LP algorithms such as Simplex or Interior Point 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 are not known efficient algorithm to solve these problems. Note that if only some of the variables are required to be integers, then the problem is called a mixed integer programming (MIP) problem. These are generally also NP-complete.

Branch-and-bound and cutting plane techniques can be used to solve small and medium size ILP problems. However these techniques do not scale well, since their running time is exponential to the size of the input. The branch-and-bound technique is based on a sophisticated enumeration of all integer solutions, which explores the solution space in a tree-like structure. Typically, branch-and-bound stops the exploration of a certain area of the solution space, if this area would not produce a better solution than the one already found. Cutting-plane technique reduces the feasible solution space in a way that the optimal integer solutions are not discarded, by adding constraints, and stops when an optimal solution is located at a vertex of the feasible region. More information about these techniques can be found in [9]. Commercial ILP solvers utilize both these techniques in branch-and-cut variations so as to combine their advantages and provide effective solutions to small and medium size ILP problems.

D. Connection of ILP and LP

1) LP-relaxation

Assume an ILP problem which is, as stated before, difficult to solve (no known polynomial time algorithms exist). Instead of solving this ILP problem we can solve the same problem without constraining the variables to take integer values (for example instead of solving the problem presented in Figure 4 we solve the problem of Figure 1). The corresponding problem is called the LP-relaxation of the ILP problem. As shortly discussed previously, the related 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, that is, if all the optimization variables 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 specifications), there are certain techniques and rules to wirte ILP formulations that can increase this probability. We will discuss this more latter in the paper. 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 (a solution to the initial ILP problem), it would have been found, since it would also be an optimal solution for the corresponding LP-relaxation problem. The branch-and-bound technique uses the LP-relaxation to calculate the objective lower (or upper) bound of a subtree/branch of the solution tree (a feasible solution subarea). It considers these bounds in order to decide the search ordering and also stops the exploration of a certain subtree/branch if its lower (upper) bound is higher than the optimal solution already found by the algorithm. Finally, given the solution of the LP-relaxation problem we can use rounding methods, such as randomized rounding, to obtain good and approximate solutions for the initial ILP problem.

2) Convex Hull

Assuming an ILP optimization problem, the same set of integer solutions can be described by different sets of constraints. Having in mind the geometrical representation of the feasible region we can visualize the set of integer solutions to be included in different-shaped *n*-dimensional polyhedrons. Figures 5 (a) and (b) show an example of an ILP problem that has the same integer solutions but is described by a different set of linear constraints. The convex hull is the minimum convex set that includes all the integer solutions of the problem (see Figure 5(c)). If we have the convex hull we can use LP algorithms to optimally solve the related ILP problem in polynomial time. However, it is difficult to define the set of constraints that would give us the convex hull of an ILP problem. Moreover, the transformation of a general *n*-dimension polyhedron to the corresponding convex hull is difficult (and is the process that is used in cutting plane techniques).

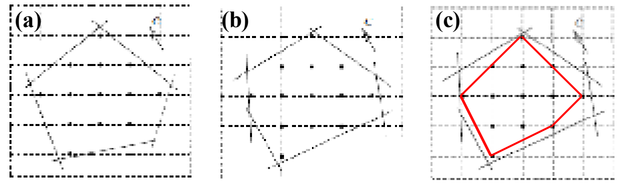


Figure 5. Integer solution set and Convex Hull.

3) Good ILP formulations

Based on the above, we can say that an ILP formulation is good if its feasible region that is defined by its linear constraints is close (tight) to the corresponding integer convex hull. If this happens then the branch-and-bound and cutting plane techniques can help us solve quickly and efficiently the corresponding ILP problem. Moreover, an ILP formulation with feasible region that is tight to the convex hull would have a large number of vertices that consist of integer variables. This could increase the probability of obtaining an integer solution when solving the LP-relaxation of the initial ILP problem. For general rules on how to write good ILP formulations the reader is referred to [10]. In what follows we present such a formulation to solve the offline RWA problem.

III. ROUTING AND WAVELENTH ASSIGNMENT PROBLEM

In this section we are going to apply the general algorithms and techniques that were presented in the previous section in order to solve the offline RWA problem.

A network topology is represented by a connected graph $G=(V,E)$. V denotes the set of nodes, which we assume not to be equipped with wavelength conversion capabilities. E denotes the set of (point-to-point) single-fiber links. Each fiber is able to support a common set $C=\{1,2,\ldots,W\}$ of W distinct wavelengths. The static version of RWA problem assumes an a-priori known traffic scenario given in the form of a matrix of non-negative integers Λ, called the traffic matrix. Then, Λ_{sd} denotes the number of requested wavelengths from source s to destination d.

The algorithm takes as input a specific RWA instance; that is, a network topology, the set of wavelengths that can be used, and a traffic matrix. It returns the RWA instance solution, in the form of routed lightpaths (paths and wavelengths), as well as the blocking probability, in case the connection requests cannot be served for the given set of wavelengths.

A. RWA LP-based algorithm

The proposed RWA algorithm consists of four phases [5]. The first (pre-processing) phase computes a set of candidate paths to route the requested connections. RWA algorithms that do not use any set of predefined paths, but allow routing over any feasible path have also been proposed in the literature. These algorithms are bound to give at least as good solutions as the algorithms that use pre-calculated paths, such as the one presented here, but use a much higher number of variables and constraints and do not scale well. In any case, the optimal solution can be also found with a RWA algorithm that uses pre-calculate paths, given a large enough set of paths. In particular, k candidate paths for each requested connection are calculated. After a set P_{sd} of candidate paths for each commodity pair $s-d$ is computed, the total set $P=\bigcup_{s-d}P_{sd}$ is inserted to the next phase of the algorithm. The pre-processing phase clearly takes polynomial time. The second phase of the proposed algorithm formulates the given RWA instance as a linear program (LP).

The LP is solved using the Simplex algorithm that is generally considered efficient for the great majority of all possible inputs, and has additional advantages, as we will see, for the problem 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 not acceptable, since a connection is not allowed to bifurcate between alternative paths or wavelength channels. Finally, phase 4 handles the infeasible instances, so that some (if all is not possible) requested connections are established.

In what follows we focus on the second and the third phase of the algorithm.

B. RWA LP formulation using piecewise linear cost function

The proposed LP formulation aims at minimizing the maximum resource usage, in terms of wavelengths used on network links. Let $F_l=f(w_l)$ denote the flow cost function, an increasing function on the number of lightpaths w_l traversing link l (the used formula is presented in the next subsection). The LP objective is to minimize the sum of all F_l values. The following parameters, constants and variables are used:

Parameters:

- $s,d \in V$: network nodes
- $w \in C$: an available wavelength
- $l \in E$: a network link
- $p \in P_{sd}$: a candidate path

Constant:

- Λ_{sd}: the number of requested connections from node s to d

Variables:

- x_{pw}: an indicator variable, equal to 1 if path p occupies wavelength w, that is if lightpath (p,w) is activated, and equal to 0, otherwise
- F_l: the flow cost function value of link l

$$\textbf{Minimize}: \sum_l F_l$$

subject to the following constraints:

- Distinct wavelength assignment constraints,
$$\sum_{\{p|l\in p\}} x_{pw} \le 1, \text{ for all } l\in E \text{ and all } w\in C$$
- Incoming traffic constraints,
$$\sum_{p\in P_{sd}} \sum_{w} x_{pw} = \Lambda_{sd}, \text{ for all } (s,d) \text{ pairs}$$
- Flow cost function constraints,
$$F_l \ge f(w_l) = f\left(\sum_{\{p|l\in p\}} \sum_{w} x_{pw}\right), \text{ for all } l\in E$$
- The integrality constraints are relaxed to
$$0 \le x_{pw} \le 1 \text{ for all } p\in P \text{ and all } w\in C$$

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

<u>*Flow Cost Function*</u>

The variable F_l expresses the cost of congestion on link l, for a specific selection of the routes. We choose F_l to be a properly increasing function $f(w_l)$ of the number of lightpaths $w_l = \sum_{\{p|l\in p\}} \sum_{w} x_{pw}$ crossing link l. $F_l = f(w_l)$ is chosen to also 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, in terms of overall network performance, to serve an additional unit of flow using several low-congested links than using a link that is close to saturation. In particular, we utilize the following flow cost function:

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

The above (nonlinear) function is inserted to the LP in the approximate form of a piecewise linear function; i.e., a continuous function, that consists of W consecutive linear parts (Figure 6). Since the LP objective is to minimize the cost $\sum_l F_l$, for a specific value of w_l, one of these W linear cost functions, and in particular the one that yields the highest $F_l(w_l)$, is satisfied with equality at the optimal solution of the LP. All the remaining linear functions are de-activated, that is, they are satisfied as strict inequalities at the optimal solution.

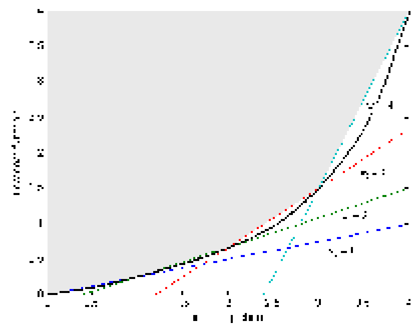


Figure 6. The set of linear constraints that are inserted in the LP formulation. We use inequality constraints to limit our search in the colored area. Since the objective that is minimized is the flow cost, we finally search for solutions only at its lower bounds, which identify the piecewise linear approximation of the flow cost function $F_l=f(w_l)$ (black line).

This piecewise linear function is equal to the nonlinear function $F_l=f(w_l)$ at integer argument values ($w_l=1,2,\ldots,W$) and greater than that at other (fractional) argument values. Inserting such a piecewise linear function to the LP objective, results in the identification of integer optimal solutions by Simplex, in most cases [8]. This is because the vertices of the polyhedron defined by the constraints tend to correspond to the corner points 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).

C. Random Perturbation

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 [11]. To be more precise, assume a general multicommodity minimization problem

$$\text{minimize } D(\boldsymbol{x}),$$
$$\text{subject to } \boldsymbol{x}\in \mathrm{X},$$

where $\mathbf{x}=(x_1,x_2\ldots,x_n)$ is a solution consisting of n flow variables, and D is a differentiable convex function. Now assume that we have an optimal solution $\boldsymbol{x^*}$ and let $x_p \neq 0$ and $x_{p'} \neq 0$ be two variables carrying fractional flows that both serve the same source-destination pair. If we move a small fraction of flow $\delta > 0$ from x_p to $x_{p'}$, so as to obtain $x_p - \delta$ and $x_{p'} + \delta$ for the corresponding flow values, the increase ΔD in the objective cost would be

$$\Delta D = \delta \cdot \left(\frac{\partial D(\boldsymbol{x^*})}{\partial x_{p'}} - \frac{\partial D(\boldsymbol{x^*})}{\partial x_p} \right).$$

For $\boldsymbol{x^*}$ to be optimal, ΔD should be greater than or equal to zero, so that such a shifting of flow from one path to the other does not increase the objective cost. A similar argument can be made by assuming that flow δ is moved from $x_{p'}$ to x_p. Therefore, the following relation must hold at the optimal solution when both flows x_p and $x_{p'}$ are nonzero

$$\frac{\partial D(x^*)}{\partial x_{p'}} = \frac{\partial D(x^*)}{\partial x_p},$$

indicating that at an optimal solution, a flow that is served by more than one paths must have equal sums of their first derivative lengths over the corresponding paths.

Now if we turn our attention to the RWA problem that we examine, a flow variable corresponds to a candidate lightpath (p,w). The objective function $D(x)$ that we utilize in our RWA formulations sums the flow costs of the links that comprise a lightpath, and thus a request served by more than one lightpath has equal sums of first derivates over the links of these lightpaths. The derivative of the cost on a specific link is given by the slope of the linear or piecewise linear flow cost function that we utilize. To make this more precise, let two lightpaths $x_{p,w}$ and $x_{p',w'}$ serve a connection request. Also let a_l be the slope of the flow cost $f(w_l)$ on link l for a given solution. At an optimal solution where $x_{p,w}$ and $x_{p',w'}$ are both nonzero and both serve the same source-destination pair, the following holds:

$$\sum_{l \in p} \frac{\partial f(w_l)}{\partial x_{pw}} = \sum_{l \in p'} \frac{\partial f(w_l)}{\partial x_{p',w'}} \Rightarrow \sum_{l \in p} a_l = \sum_{l \in p'} a_l.$$

To increase the number of integer solutions obtained, we use the following random perturbation technique. To make the situations where two lightpaths have equal first derivative lengths over the links that comprise them less probable, and thus obtain more integer solutions, we multiply the slopes on each link with a random number that differ to 1 in the sixth decimal digit. Thus, we defining different slopes $a_{l,pw}$ for each lightpath (p,w) and for each link $l \in E$.

D. Iterrative Fixing and Rounding Teqniques

If even with the piecewise linear cost function and the random perturbation technique 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 from the previous solution to a solution with equal or more integers with the same cost. Since the objective cost does not change, if after successive fixings we reach an all-integer solution we are sure that it is an optimal one. On the other hand, fixing variables is not guaranteed to return an integer optimal solution, if one exists, since the integer solution might consist of different integer values than the ones gradually fixed. When we reach a point beyond which the process of fixing 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 value of the objective cost, rounding makes us move to higher objective values and search for an integer solution there. Rounding is inevitable when there is no integer solution with the same objective cost as the LP-relaxation of the RWA instance. 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.

E. IA-RWA

In transparent or translucent WDM networks, where lightpaths remain in the optical domain for more than one link, *physical layer impairments* affect the quality of transmission (QoT) of a lightpath. The interdependence between the physical and the network layers makes the RWA problem in the presence of physical impairments a cross-layer optimization problem. To address this, a number of approaches are emerging, usually referred to as impairment-aware (IA)-RWA algorithms. The reader is referred to [5] for a detailed description of the offline IA-RWA problem.

IV. Performance Results

To evaluate the performance of the proposed RWA algorithm we carried out a number of simulation experiments. We implemented the algorithm in Matlab and used LINDO [12] to solve the corresponding LP and ILP problems. We evaluate the integrality and optimality performance of the proposed LP-relaxation algorithm and the random perturbation technique (Section III.C). The network topology used in our simulations was the generic Deutsche Telekom network (DTnet), shown in Figure 7.

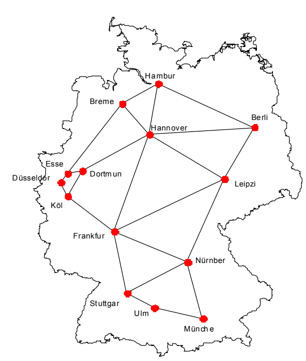


Figure 7: Generic DT network topology. 14 nodes and 46 directed links.

To have a reference point, we also executed the same experiments using a typical min-max formulation (a formulation whose objective is to minimize the maximum number of wavelengths used), which was optimally solved using the ILP algorithm of [12]. Note that the maximum number of wavelengths used is the actual objective that we want to minimize. The piecewise linear cost function used in the proposed LP RWA algorithm (Section III.B) tries to approximate the min-max objective, while also being continuous and piecewise linear, so as to exhibit a good integrality performance when the Simplex algorithm is used. Thus, the ILP-min_max algorithm sets the criterion in terms of optimality. We also used the same min-max formulation and solved its LP-relaxed version followed by iterative fixing and roundings. This LP-min_max algorithm sets a comparison criterion in terms of integrality and execution time, since its difference to our proposed LP algorithm lies on the piecewise linear cost function that we utilize and the random perturbation technique. For all algorithms we have used k=3.

The results were averaged over 100 experiments corresponding to different random static traffic instances of a given traffic load (for the definition of load ρ please see Section IV.D). More specifically, we have performed experiments for loads ranging from 0.5 up to 2 with 0.5 step.

To evaluate the performance we used the following metrics:

(a) The number of used wavelengths averaged over all experiments. This is the objective we want to minimize.
(b) The fraction of instances for which we obtained an integer solution by the LP execution (without any fixing and rounding iterations).
(c) The number of "fixings" and "roundings" required to obtain integer solutions that are guaranteed to be optimal, averaged over all experiments; this is the average number of fixing and rounding iterations performed to move from (b) to (d).
(d) The fraction of instances that we are sure to have found an optimal solution (corresponding to instances for which there was no increase in the objective cost of the LP).
(e) Average number of fixing and rounding iterations for the cases that we are not sure to obtain an optimal solution; this is the average number of fixing and rounding iterations performed to move from (d) to (f).
(f) The fraction of instances that we have found an integer solution after fixing and rounding iterations irrespective of the optimality ((f) is always 1 since we always succeeded in obtaining integer solutions).
(g) Average running time (in sec): the average running time of the simulation experiments, including the tableau creation, the LP (or ILP) execution and the fixing and rounding iterations until we obtain integer solutions (when applicable).

Table 1 presents the corresponding results. From this table we can see that the proposed LP-piecewise algorithm finds solutions (column (a)) that are closer to the optimal ones (as expressed by column (a) of the ILP-min_max algorithm) than those obtained by the LP-min_max algorithm. The random perturbation technique seems to improve the performance of the algorithm, being able to find in some cases better solutions that use a smaller maximum number of wavelengths. This is because the random perturbation technique yields more integer solutions without fixings (metric (b)) and solutions that are guaranteed to be optimal (column (d)) than the LP-piecewise algorithm without it. When using the random perturbation technique, for all the experiments performed, the optimality was lost only for one instance (of load ρ=1). The random perturbation technique reduces the number of fixing and rounding iterations (column (c) and (e)) that are performed and has a slightly better running time (column (g)). The execution time of the LP-min_max algorithm is higher than that of the proposed LP-piecewise algorithm due to its bad integrality performance and the high number of fixing and rounding iterations it has to perform to obtain an integer solution.

Thus, the proposed LP-piecewise algorithm has superior overall performance than the other algorithms examined. It finds with high probability an optimal solution while maintaining low execution times. The random perturbation technique makes the proposed LP-piecewise algorithm even better, since it increases its integrality performance, and also reduces the running time. The good optimality performance of the proposed algorithm is maintained for high loads.

TABLE 1: PERFORMANCE OF THE PURE RWA ALGORITHMS

Load	Cost Function	a	b	c	d	e	f	g
0.5	ILP-min-max	7.70	1	n/a	n/a	n/a	n/a	7.18
	LP-min-max	7.76	0	9.76	0.61	16.71	1	16.72
	LP-piecewise	7.72	0.43	1.29	0.87	6.38	1	2.3
	LP-piecewise + random perturbation	7.70	0.94	1.1	1	0	1	1.27
1	ILP-min-max	14.01	1	n/a	n/a	n/a	n/a	91.93
	LP-min-max	14.03	0	9.78	0.63	16.77	1	22.5
	LP-piecewise	14.02	0.08	2.14	0.88	5.92	1	7.72
	LP-piecewise + random perturbation	14.02	0.75	1.18	0.98	3	1	5.17
1.5	ILP-min-max	20.53	1	n/a	n/a	n/a	n/a	840.2
	LP-min-max	20.56	0	9.74	0.59	15.38	1	29.85
	LP-piecewise	20.54	0.06	2.1	0.87	7.31	1	17.03
	LP-piecewise + random perturbation	20.53	0.53	1.17	0.98	4	1	14.52
2	ILP-min-max	26.66	1	n/a	n/a	n/a	n/a	3992
	LP-min-max	26.68	0	10.75	0.46	14.14	1	38.44
	LP-piecewise	26.68	0.02	2.46	0.88	3.45	1	30.85
	LP-piecewise + random perturbation	26.66	0.48	1.28	0.98	4	1	30.84

V. CONCLUSIONS

We outlined some general techniques that can be used to solve complex and difficult designing problems for optical networks. We applied these techniques and proposed an algorithm to solve the planning problem of a WDM network that corresponds to the offline or static RWA problem. Our RWA algorithm is based on LP-relaxation formulation that uses piecewise-linear cost function to increase the probability of obtaining optimal integer solutions. We also presented a random perturbation technique that can be used to enhance the performance of the algorithm and iterative fixing and rounding methods. Through simulation experiments we verified that the proposed LP-relaxation algorithm has superior overall performance and is able to find with high probability an optimal solution while maintaining low execution times.

REFERENCES

[1] T. Stern, G. Ellinas, K. Bala, "Multiwavelength Optical Networks: Architectures, Design and Control", Cambridge Press, 2nd Edition, 2008.
[2] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high-bandwidth optical WANs," IEEE Transactions on Communications, vol. 40, pp. 1171–1182, 1992.
[3] H. Zang, J. P. Jue, B. Mukherjee, "A Review of Routing and Wavelength Assignment Approaches for Wavelength-Routed Optical WDM Networks", Optical Networks Magazine, vol. 1, 2000.
[4] R. Dutta, A. Kamal, G. N. Rouskas, "Traffic Grooming for Optical Networks: Foundations, Techniques and Frontiers", Springer, 2008.
[5] K. Christodoulopoulos, K. Manousakis, E. Varvarigos, "Offline Routing and Wavelength Assignment in Transparent WDM Networks", IEEE/ACM Transactions on Networking, Vol. 18, Is. 5, October 2010.
[6] R. Guerin, A. Orda, "Networks with Advance Reservations: The Routing Perspective", IEEE INFOCOM, 2000.
[7] R. Ahuja, T. Magnanti J. Orlin, "Network Flows: Theory, Algorithms, and Applications", Prentice Hall, 1993.
[8] A. Ozdaglar, D. Bertsekas, "Routing and Wavelength Assignment in Optical Networks", IEEE/ACM Transactions on Networking, 11(2), 2003.
[9] C. Papadimitriou, K. Steiglitz, "Combinatorial Optimization: Algorithms and Complexity", Dover publications, 1998.
[10] D. Bertsimas, R. Weismantel, "Optimization over integers", Dynamic Ideas, 2005.
[11] D. Bertsekas, R. Gallager, "Data Networks", 2nd ed., Prentice Hall, 1992.
[12] LINDO: http://www.lindo.com/