Physically-Constrained Lightpath-Setup in Dynamic WDM Networks

D. A. Schupke[1], H. Bülow[2], S. De Patre[3], M. Gunkel[4], H. Haunstein[5], A. Lord[6], J. Martensson[7], C. Matrakidis[8], B. Niven-Jenkins[6], C. Politi[8], M. Potenza[9], A. Rafel[6], A. Stavdas[8], S. Sygletos[8]

[1] Siemens AG, Corporate Technology, 81730 Munich, Germany (contact author)
Tel: +49 89 636-43518, Fax: -51115, E-mail: dominic.schupke@siemens.com

[2] Alcatel SEL, Stuttgart, Germany [3] Pirelli, Milano, Italy [4] T-Systems, Darmstadt, Germany
[5] Lucent Technologies, Nuremberg, Germany [6] BT Exact, Ipswich, UK [7] ACREO, Kista, Sweden
[8] National Technical University of Athens, Athens, Greece [9] TILabs, Torino, Italy

This paper provides a high-level overview of network planning and operation processes for lightpath computation in WDM networks. Lightpath demands occur dynamically and the lightpaths are constrained by physical effects. We classify different approaches to connection setup and describe algorithms suitable for this purpose.

1 Introduction

We provide a high-level overview of network planning and operation processes, in which physically-constrained lightpath computation is embedded. A lightpath in this context means an optically transparent connection within a network domain. Our assumption is that we deal with networks that receive random lightpath demands over the operation time. We also summarise different approaches to connection management (in particular set-up) and describe viable algorithms for this purpose. This paper presents the results of the discussion within Work Package 5 of NOBEL, an EU 6th Framework project.

After characterising computation time requirements in Section 2, we provide the high-level structure in Section 3. Section 4 describes and discusses a class of typical algorithms. Section 5 presents our conclusions.

2 Timing Requirements

The *computation* timing requirements for connection set-up/tear-down depend on the protection architecture deployed in the system, see Table 1. If re-routing after failure is deployed in the system and if it demands quick recovery (R1), then the computation has to be very quick as well (some 10 ms). Otherwise, a value around 1 second (R2) seems reasonable for computation.

	Computation for a connection (1 path)	**Computation for a connection pair (1 working path, 1 backup path)**	**Computation for recovering connection after failure (1 path)**
Requirement	~1 second	~1 second	R1: some 10 ms R2: ~1 second
Unprotected, pre-emptible	X		
1+1 protected, shared-protected (preplanned restoration)		X	
Rerouting after failure (on-demand restoration)	X		X

Table 1: Timing requirements for connection set-up/tear-down computation.

3 High-Level Structure of Network Planning and Operation

A general effect in the network planning and operation process is that one depends on the other, see Figure 1. The performance of a connection set-up algorithm depends on the capacities obtained from the network planning process, while network planning depends on the way connections are chosen by the connection set-up algorithm later on.

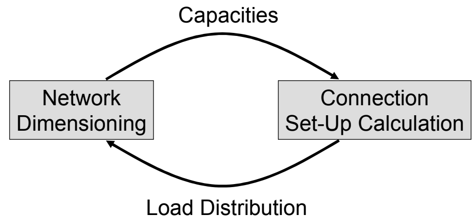

Figure 1: Mutual dependency between network planning and operation.

We distinguish between an upgrading model and a steady-state model, see Figure 2. The upgrading model reflects the actual planning and operation of typical networks, where demands are set up as they arrive (and are rarely torn down); network capacity (links, elements, etc.) are inserted from time to time, in order to accommodate the dynamically and (on average) growing demand. The steady-state model is often used in the literature for simulations as it assumes demands are randomly set up and torn down, both with the same rate. The network capacity is constant and the dynamic demand has also a constant average. Typically, the start-up period required to fill the network is disregarded in the steady-state model.

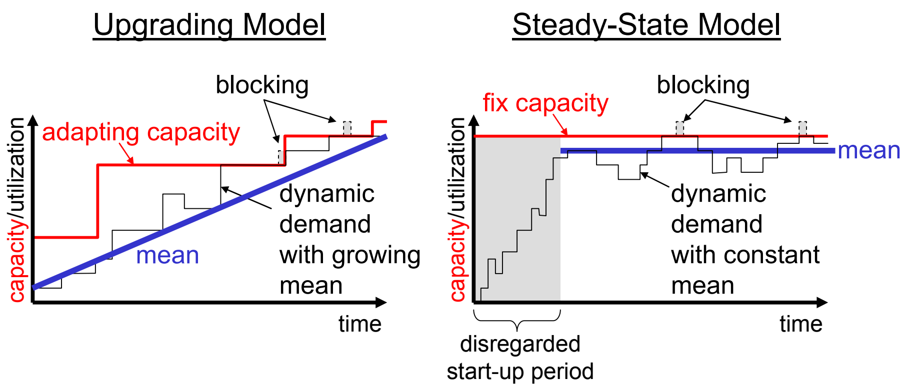

Figure 2: The upgrading model and the steady-state model.

In this paper we present several options for operating a connection set-up algorithm after the network has been planned. The connection set-up algorithm includes computation of the route (fibres used), the wavelengths, the transponders, and the regenerators, all based on a topology and available resources. Hence, a connection set-up algorithm includes routing and wavelength assignment (RWA). As a first step, we consider only non-regenerated connections in this paper, i.e., a connection is a lightpath. In addition, we focus on single unprotected demands presented to the connection calculation.

Besides centralised implementations, distributed or hybrid implementations of the processes are realisable. Hence, network nodes (and possibly a centralised system) can go through the following flow diagrams jointly, using signalling and routing protocols of a control plane.

3.1 Network Planned Such That Any Connection is Physically Feasible

In this approach (Figure 3) the network is planned, such that any connection can be established by physically feasible transparent sub-paths. The connection set-up algorithm is simple, since it does not need to consider physical constraints (except for the wavelength continuity constraint). In this process, for a transparent network, a feasible connection refers to a free wavelength on a route.

A common method is to form transparent islands which are delineated by opto-electro-optical (O-E-O) conversion, i.e., connections terminated at or passing through the borders of sub-networks undergo O-E-O conversion. Network planning has to ensure that any sub-path within a transparent island is physically feasible. The routing can be enhanced by multi-domain routing to account for the transparent sub-networks and their interconnection points.

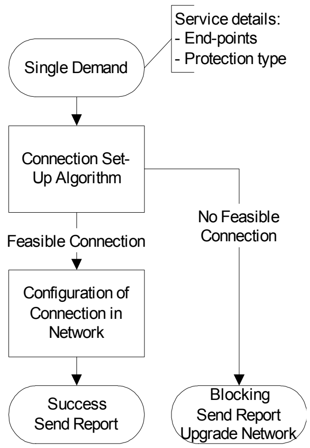

Figure 3: Physical constraints unaware connection set-up.

Qualitatively, the simplicity of this approach makes it the most attractive for use during network operation, both for static and dynamic networks, when short computation time is advantageous. In addition, the computation time can be kept very short. However, since this approach requires O-E-O conversion at the edges of transparent islands, the network can become expensive. For methods to design transparent islands see [1] and the references therein. The use of transparent islands also raises issues of flexibility and scalability.

3.2 Network Planned Such That Between All Node Pairs Some Connections are Physically Feasible – Reactive Process

In this approach the network is planned, such that between all node pairs some number of connections are physically feasible. This is to ensure that the connection set-up algorithm can find solutions initially. As a reactive process, the connection set-up algorithm is invoked after a connection demand arrives.

Figure 4 shows the first variant. The connection set-up algorithm is more complex, since it can consider physical constraints [2] and can run optical verification simulations as a sub-procedure of the connection set-up algorithm. The inclusion of physical constraints can facilitate optical verification simulation (simpler simulation or better hit ratio) or even avoid it. In this context, a particular proposal is the abstraction of the physical constraints by normalised sections [3] without simulation during operation.

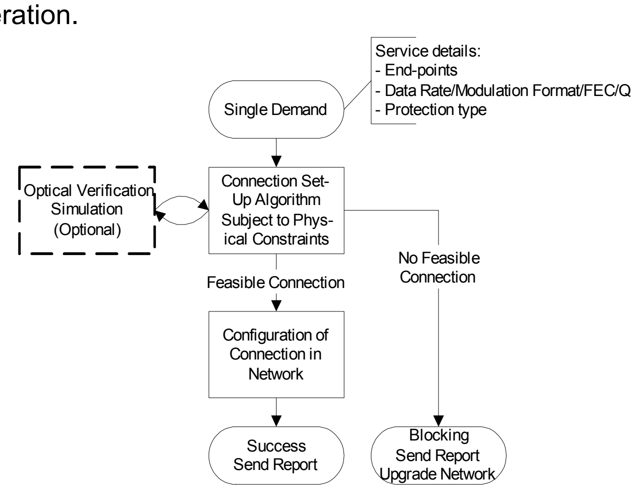

Figure 4: Physical constraints aware connection set-up (computation-oriented, reactive process).

Qualitatively, this approach can achieve close-to-optimal performance for the individual connections (e.g., O-E-O sparing) and for the network (e.g., low load). However, because of the complex calculations (for physically-constrained routing and/or simulation), centralised network controllers may be needed. Long computation delays can make real-time connection set-up impossible. The optimal performance may not be reached if (for computational speed) approximate calculations are employed. Network planning needs to consider that approximate calculations or normalised sections are used, and it can therefore involve sub-optimal design results.

Figure 5 shows the second variant. The connection set-up algorithm can again be kept simple, since as the only additional constraint is to avoid forbidden connections. The feasibility of the connection (and possibly of all existing connections, see the state dependence discussion in Section 4) is verified by performance probing. A connection is marked as forbidden if it cannot achieve the target performance as determined by the performance monitor (see Figure 5). Connections calculated by the algorithm are configured and probed iteratively, until a feasible connection (passing the performance monitoring check) can eventually be configured for operation.

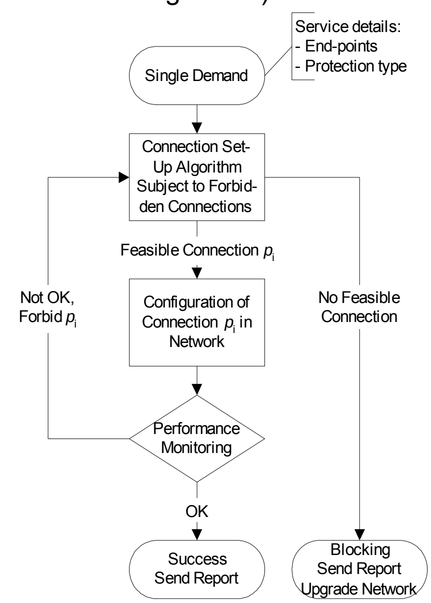

Figure 5: Physical constraints aware connection set-up (probing-oriented, reactive process).

Qualitatively, this approach is quite simple and can achieve optimised network performance. While it is an exact method to check connection feasibility, it needs performance monitoring equipment. Long measurement delays can make real-time connection set-up impossible. Moreover, since the connection set-up algorithm does not directly produce a feasible path, the approach can involve long probing iterations. In addition, if new connections affect other connections, this approach can make set-up for future connections too restrictive.

3.3 Network Planned Such That Between All Node Pairs Some Connections are Physically Feasible – Proactive Process

This proactive approach is based on the first variant in the previous section, which is enhanced by a pre-computation process. After the connection for a requested demand is established in the network, the process calculates (pre-computation) and stores eligible connections for all possible future demands that can be anticipated. This approach is feasible if the pre-computation time is lower than the demand inter-arrival times.

Qualitatively, this approach (as the first variant of Section 3.2) can achieve the most optimal network performance, but may need centralised network controllers for computation. Unlike the approaches in Section 3.2, real-time connection set-up is possible, since the connection details are pre-stored and only a simple look-up has to be done. However, a dual process and storage for connection details are needed. It is also necessary to ensure that the computation time does not exceed the demand inter-arrival times and that all anticipated demands are covered (but, as a fall-back, a reactive process is still possible).

4 Algorithm Descriptions

For the routing problem there are generally three approaches that are used in the literature depending whether the paths are pre-calculated or not. The **fixed** routing assumes a specific path for each connection. Obviously, this method can lead to high blocking probabilities, if the resources along the path are tied up, and it cannot directly handle faults in the network. The **fixed-alternate** approach considers multiple alternative routes in the network between each source node and its destination node and a list of possible routes is stored in the corresponding source node. **Adaptive** routing is preferred over the above two methods because the route between the source and the destination is chosen dynamically for each lightpath that is requested, depending on the state of the network at that epoch in time. When a connection request appears, an appropriate path is calculated. With adaptive routing a connection is blocked, only if there is no feasible route between the source and destination nodes. It must be noted that in order to perform adaptive routing in a dynamic network, there is an overhead at the control and management layers. However, enhanced network performance is expected.

Most connection set-up algorithms for calculating a path between two nodes use or are based on a standard **shortest path algorithm**. The well-known Dijkstra algorithm [4] is often used; it can be implemented to compute with low computation complexity. A shortest path algorithm minimises the route weight (which is the sum of its link weights). The main motivation for this is that the weights usually represent something that is a measure of cost. Note that the terms weight, length, cost, and metric are often used synonymously.

There are many problem dependent algorithms:

1. ***k*-Shortest paths algorithms** (see [5]) compute paths in the following way. Suppose w_1 is the minimum weight achieved by some path between the end nodes, w_2 the next larger weight, etc. The algorithm calculates all paths with w_1, all paths with w_2, … until w_k is reached. Note that for a specific w_i, multiple paths can exist. *k*-Shortest paths algorithms are used, for example to produce a set of path options for iterative algorithms.
2. ***k*-Shortest edge-disjoint paths algorithms** and ***k*-shortest node-disjoint paths algorithms** (see [6]) compute *k* paths (i) whose overall weight sum is at minimum and (ii) which are mutually edge-disjoint and node-disjoint, respectively. This is used, for example with *k*=2 for 1+1 protection.
3. **Shortest path algorithms subject to constraints** [7] compute paths which are minimal in weight and which fulfil a set of further constraints. Common constraints are expressed by additive, multiplicative, and concave functions of further link-values. In general, we can use these constraints to directly include physical effects in the computation.
4. **Shortest pair of disjoint paths algorithms** subject to constraints combine 2 and 3.

The above algorithms are concerned with the path computation problem. For transparent WDM networks, we have to supplement it by the wavelength assignment problem, surveyed in [8]. For a static network with a given set of paths, the wavelength assignment problem is subsequently directed to assign a wavelength to each path in a way that no two paths share the same wavelength on the same fibre link. For the dynamic cases, however, there are plenty of heuristics that have been proposed and can be combined with a path calculation algorithm. Here, minimising the blocking probability is the main objective. The calculations are performed online and make use of the current state information.

An important issue is the state dependence. At best, a connection to be routed depends only on the availability of wavelengths in the network. If, however, the physical performance of a connection depends also on the network state, then the interrelation of a new connection brought into the network with the existing and future connections may be tested by: (a) verifying the feasibility of all connections individually (the new one, and the existing and future ones), or by (b) verifying the feasibility of the new connection using a collapsed constraint from all existing and future connections. Moreover, connection tear-down becomes an issue, since removing an existing lightpath can potentially affect other currently existing lightpaths.

The following subsections describe the structure of three classes for computing the connection details. The algorithms differ in their computational complexity, used format of (input, internal, and output) data, flexibility, and last but not least in their network performance (e.g., blocking probability and utilisation).

4.1 Iterative Algorithms

Iterative algorithms require a priority order, i.e., an order in which multiple iterative loops are nested and an order for the iterations of the loops. As an example, consider the two nested loops in Figure 6. The outer loop considers a (pre-computed) set of routes (e.g., the *k* shortest ones) and the inner loop considers the system wavelengths. Because both loops stop when the first feasible solution is found ("first fit"), route assignment has higher priority than wavelength assignment. This priority order keeps connections short and can thereby minimise the average per-connection resources in the network. The loops can also be swapped such that wavelength assignment precedes route assignment. This priority order tries to maximise the number of wavelengths available in the overall network.

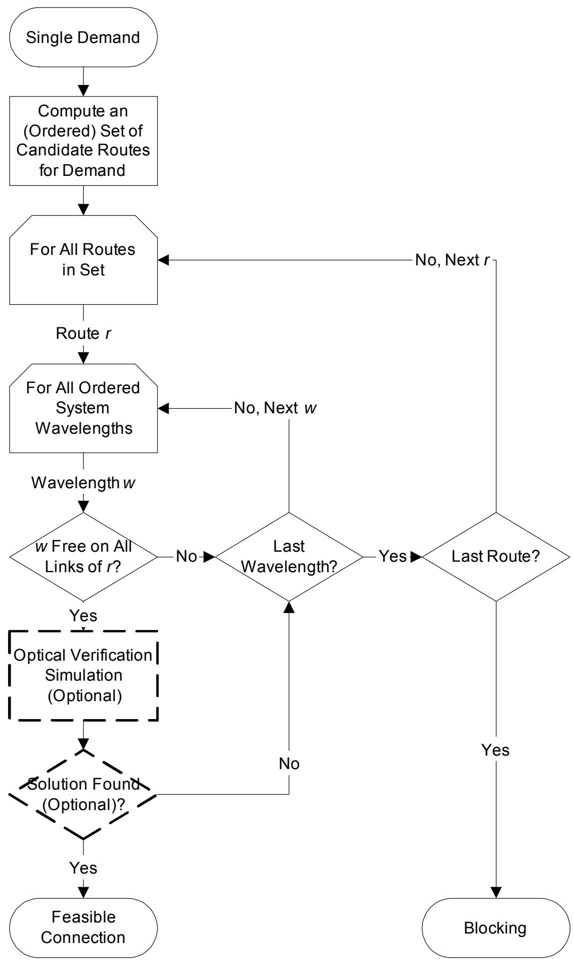

Figure 6: Example of an iterative algorithm.

To check that a solution is feasible, all wavelengths on the route have to be free and, optionally, the optical verification has to give a feasible result. The ordering of both routes and wavelengths according to suitable measures ensures that the solution is optimal in the loop priority order. For example, if routes are ordered by increasing lengths and wavelengths are ordered by increasing transparent reach (see [9]), longer routes will only be tested after all wavelengths on the shorter routes are tested as not free or optically verified as not feasible. If eventually wavelengths are feasible for a route, a feasible wavelength with the minimal transparent reach will be taken.

If a different measure for a connection is adopted, we can choose other loop configurations. As an alternative, we can modify the existing loop stop criterion to make the loops consider all routes and all wavelengths. During running through the loops, the best solution so far is kept and finally taken. This "brute force"-like method can become very time-consuming when the optical verification simulation is used as sub-process.

As depicted in the example, the optical verification simulation is easily introduced into the iteration approach. The iterative structure becomes more complex (e.g., by more loop levels) if it additionally accounts for protection/restoration and regeneration.

4.2 Implicit Algorithms

Implicit algorithms integrate the physical and network constraints into a graph model on which the computations are done. A common approach is to transform the fibre topology graph (e.g., as in Figure 7 (a)) into a more complex graph (Figure 7 (b)) on which a standard routing algorithm is invoked. The result of this algorithm is then used to configure the connection.

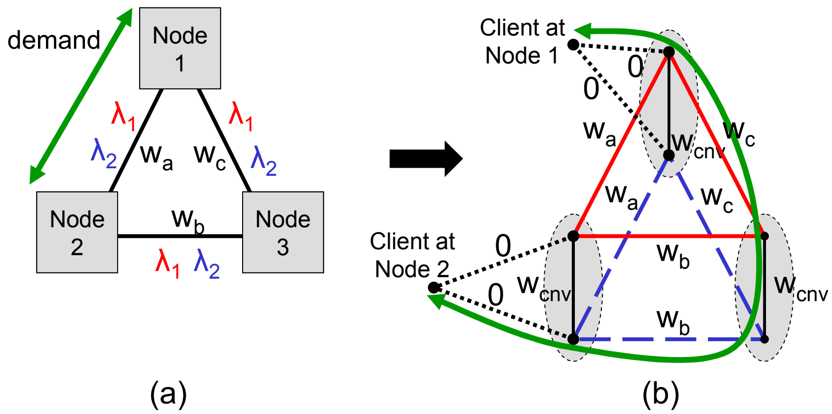

Figure 7: Example of a model for an implicit algorithm.

In more detail, the example Figure 7 (a) shows a simple network with W=2 wavelengths, routing weights per link, and a demand between Node 1 and 2. This network is transformed, into the graph Figure 7 (b); in effect the network is copied W times (W wavelength layers), each of these layers is linked to the two end-points of the demand (at its nodes, dotted black links), and if arbitrary wavelength conversion (or regeneration) is possible at a node, the layers are further fully meshed at a node (this is done in the example, where only one link is needed per node, see the black links labelled w_{cnv}). The routing weights from the original network apply on each layer, routing weights for client connection are naught, and routing weights for conversion (w_{cnv}) can be set to a desired value (typically to a value higher than the sum of all network weights such that conversion is only used if required). The weight of a layer-link can be set to infinity if the corresponding wavelength on the fibre-link is used. In this transformed graph, a suitable algorithm can be invoked (e.g., the Dijkstra algorithm). A feasible path between node 1 and 2 is shown in Figure 7 (b). For this path, links b and c are used with wavelengths 2 (blue dashed line) and 1 (red solid line) respectively; wavelength conversion is done in node 3.

This approach can easily reuse existing algorithms. However, if these algorithms are invoked on a more complex network they could result in longer computation times. The inclusion of constraints is possible, but it seems hard to directly include physical constraints if regeneration points are possible (e.g., as modelled above), since these constraints apply to transparent subsections between regeneration points.

4.3 Optimisation Approaches (LP, MILP)

In general, an optimisation model to solve an optimisation problem comprises variables, constraints, and an objective. The (decision) variables reflect the parameters which should be determined. The objective (function) is a function of these variables which should be minimised or maximised in the problem. The set of constraints are equalities and inequalities which restrict the variables to values that represent an admissible solution to the problem. We call a feasible solution a setting of variables, which fulfil all constraints, and an infeasible solution a setting of variables, that does not fulfil one or more constraints.

Usually, the optimisation models are linear, i.e., the objective and the constraints are linear. We can employ linear optimisation models with continuous variables, called *linear programs (LPs)*, and integer linear optimisation models additionally with integer variables, called *mixed integer linear programs (MILPs)*.

After formulating a problem as optimisation model, specialised optimisation algorithms can calculate feasible optimal (or sub-optimal) solutions of the optimisation model. These algorithms can also determine that a problem has no feasible solution. Well-known algorithms are simplex algorithms for LPs and branch-and-bound algorithms for MILPs. Commercial software tools, e.g., CPLEX [10], or free software tools, e.g., lpsolve [11], incorporate these algorithms.

Such optimisation approaches are typically used for network planning, typically based on link-flow models (e.g., in [12], [13]) and path-flow models (e.g., in [12]). Recent literature has shown that these models can also be feasible for (centralised) connection set-up calculation[14].

5 Conclusions

A transparent optical network is a complex system, with complex interdependencies between entities and functions (e.g., the wavelength selected for a new lightpath may adversely affect existing lightpaths in the network). Accordingly routing and wavelength assignment algorithms designed to accurately consider physical constraints and calculate the optimal route and wavelength assignment also likely exhibit significant complexity. While it is relatively quick and easy to calculate lightpath routing and wavelength assignments using very simple heuristics, these may not sufficiently model the physical constraints inherent in a transparent optical network and may not produce the required performance (in terms of blocking).

Therefore, when producing a routing and wavelength assignment algorithm for physically constrained lightpath setup, it is necessary to trade off the computational complexity (and therefore run-time) of the algorithm against the accuracy with which the algorithm models the physical aspects and constraints of the network. The exact trade-off required is likely to differ depending on many factors, including (but not limited to): The operator deploying the algorithms (and their specific commercial and operational pressures), the geographic region the network is situated in, and the services/applications that will run over the network. It is therefore necessary to see lightpath setup within the general context of network planning and operation. A rich set of methods is available for accomplishing lightpath calculation. A viable solution combines operational requirements (e.g., a given computation time) and good network performance (e.g., low blocking probability), given that the physical performance requirements are met.

In further work, Work Package 5 of NOBEL is going to assess approaches for lightpath setup using simulation of network case studies.

References

[1] M.C. Scheffel, “Regenerator Allocation Strategies for Optical Transparency Domains Considering Transmission Limitations,” IEEE International Conference on Communications (ICC), Seoul, Korea, May 16-20, 2005.

[2] J. Strand; A.L. Chiu; R. Tkach, “Issues For Routing In The Optical Layer,” IEEE Communications Magazine, February 2001.

[3] Hanik, N.; Gladisch, A.; Lehr, G., "An effective method to design transparent optical WDM-networks", Europ. Conf. On Networks and Optical Commun. (NOC), Manchester, UK, 1998.

[4] J. L. Gross; J. Yellen, “Graph Theory and Its Applications, “ CRC Press, Boca Raton, 1998.

[5] D. Eppstein, “Finding the k Shortest Paths,” IEEE Symp. Foundations of Computer Science., Santa Fe, 1994.

[6] R. Bhandari, “Survivable Networks,” Boston: Kluwer, 1999.

[7] S. Chen; K. Nahrstedt, „An Overview of Quality of Service Routing for Next-Generation High-Speed Networks: Problems and Solutions,” IEEE Network, Nov./Dec. 1998.

[8] H. Zang; J.P. Jue; B. Mukherjee, “A Review of Routing and Wavelength Assignment Approaches for Wavelength-Routed Optical WDM Networks,” Opt. Net. Mag., vol. 1, Jan. 2000.

[9] G. Lehmann; M. Schuster; D.A. Schupke; S. Duhovnikov; G. Göger, “Benefits of wavelength dependent reach in transparent optical networks”, Invited Talk, Asia-Pacific Optical Communications Conference (APOC), Shanghai, China, November 6-10, 2005.

[10] ILOG Inc., CPLEX, www.ilog.com, 2005.

[11] Netlib, lpsolve , http://www.netlib.org/ampl/solvers/lpsolve/, 2005.

[12] N. Wauters and P. Demeester, “Design of the Optical Path Layer in Multiwavelength Cross-Connected Networks,” IEEE Journal in Selected Areas on Communications, vol. 14, pp. 881–892, 1996.

[13] Schupke, D.A., "Off-line Lightpath Routing in WDM Networks with Different Wavelength Converter Configurations," IEICE/IEEE Workshop on High Performance Switching and Routing (HPSR), Kobe, Japan, May 26-29, 2002.

[14] I. Tomkos; D. Vogiatzis; C. Mas; I. Zacharopoulos; A. Tzanakaki; E. Varvarigos, "Performance Engineering of Metropolitan Area Optical Networks through Impairment Constraint Routing," IEEE Communications Magazine, Optical Supplement, August 2004.