Scheduling technology migration in WDM Networks

Andreas Bley[(1)], Fabio D'Andreagiovanni[(2)], Daniel Karch[(1)]

[(1)]Institut für Mathematik, Technische Universität Berlin, Straße des 17. Juni 136, 10623 Berlin,
{bley, karch}@math.tu-berlin.de
[(2)]Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Takustraße 7, 14195 Berlin,
d.andreagiovanni@zib.de

Abstract

The rapid technological evolution of telecommunication networks demands service providers to regularly update their technology, with the aim of remaining competitive in the marketplace. However, upgrading the technology in a network is not a trivial task. New hardware components need to be installed in the network and during the installation network connectivity may be temporary compromised. The Wavelength Division Multiplexing (WDM) technology, whose upgrade is considered in here, shares fiber links among several optical connections and tearing down a single link may disrupt several optical connections at once. When the upgrades involve large parts of a network, typically not all links can be upgraded in parallel, which may lead to an unavoidable longer disruption of some connections. A bad scheduling of the overall endeavor, however, can dramatically increase the disconnection time of parts of the networks, causing extended service disruption.

In this contribution, we study the problem of finding a schedule of the fiber link upgrades that minimizes the total service disruption time. To the best of our knowledge, this problem has not yet been formalized and investigated. The aim of our work is to close this gap by presenting a mathematical optimization model for the problem and an innovative solution algorithm that tackles the intrinsic difficulties of the problem. Computational experience on realistic instances completes our study.

Our original investigations have been driven by real needs of *DFN,* operator of the German National Research and Education Network and our partner in the BMBF research project *ROBUKOM* (http://www.robukom.de/).

1 Introduction

During the last years, telecommunications networks have faced an astonishing increase in traffic volume and have consequently grown in size and complexity. Major telecommunication companies, such as Nokia Siemens Networks (NSN) [5], predict that global data traffic will continue to grow exponentially, doubling its current volume in less than two years and reaching an exchange volume of more than 1000 Exabyte per year by the end of 2015 [7]. However, such growth will not be accompanied by an equivalent growth of the revenue. Providers will therefore have to continuously improve network efficiency and carefully consider the opportunity of implementing the latest hardware advancements to contain and reduce operational costs. Hardware upgrade for accommodating increase in traffic volume constitutes just one of the cases in which technology updates are required in a network. Novel technologies and updates of current technological standards are indeed frequently introduced to improve other key network features like resilience, security and energy efficiency. Neglecting important advances is simply not possible for a network provider, since this would deny relevant increases in operational efficiency and thus decrease in the business profit margin.

However, implementing technological upgrades is not a trivial task. Upgrades typically require the installation of new hardware components and, during the installation, network connectivity may be compromised, which in turn may lead to a temporary service disruption. Similarly, capacity expansion and maintenance operations often require the disconnection of the affected network elements and may thus lead to the disruption of user connections.

In the case of a major technological upgrade involving a large part of the network, it is likely that a considerable amount of time is required to perform all necessary operations, as the number of available technicians (or some other resource) is in general limited. This situation can get even worse if a complete nation-wide network must be upgraded and the team of technicians must visit places that are far away from each other. In such a case, a bad scheduling of the individual operations may produce prolonged disconnections and service disruption, causing, in the worst case, heavy violations of service level agreements and high penalization fees for the service provider.

In this work, we investigate the problem of scheduling the upgrade of a WDM network to a new link technology with the objective of minimizing the total disruption time of the optical connections that are established in the network. Surprisingly, to the best of our knowledge, this practically very relevant optimization problem has not yet been investigated. The aim of our work is to close this gap, proposing both an optimization model and an innovative solution algorithm for this problem.

Our work has been also inspired by the needs of our partner *DFN* in the BMBF funded research project *ROBUKOM* [1,3]. DFN is the no-profit organization operating the German National Research and Education Network X-WiN connecting the German universities and research institutions. DFN has recently faced the problem of upgrading the technology in the optical layer of X-WiN [6] and has evaluated the possibility of adopting an optimization tool to automate and improve the migration process.

2 The network migration problem

In this section, we formally define and model the fiber upgrade scheduling problem described above as a mathematical optimization model. For the sake of simplicity, we assume that the optical connections established in the network are not protected and that all fibers of the network need to be upgraded. The presented model can be easily modified to cope with protected connections and partial upgrades.

The underlying optical fiber network is represented as an undirected graph $G=(V,E)$, where V denotes the set of nodes and E denotes the set of fiber links. Given a subset of links $F \subseteq E$, we denote by $V(F)$ the set of nodes that are terminals of the links in F, i.e., $V(F) = \{v \in V: \exists (u,v) \in F\}$. Each link $e \in E$ is based on an old technology A and must be updated to a new technology B within a time horizon $T=\{0,1,...,|T|\}$. In the starting period $t=0$ all links have technology A and by the last period $t=|T|$ all the links must be upgraded to technology B.

Upgrading a link $e=(u,v) \in E$ requires a_u and a_v many technicians at the end nodes u and v of the link, respectively, and a_e many technicians along the link, replacing simultaneously the relevant components at the nodes and the optical signal repeater locations. Note that each link may be upgraded only as a whole in a single time period. Yet, regardless of the number of links incident to node u that are upgraded simultaneously in time period t, only a_u many technicians are required at node u in period t. These a_u technicians can handle the upgrade of all incident links. In other words, the number of technicians required at u in period t only depends on whether *some* link with end node u is upgraded in period t, but not on the number of links upgraded simultaneously.

The number of link upgrades that can be performed in each period is limited by the number K of technicians that are available. Formally, a subset of links $F \subseteq E$ may be upgraded simultaneously in a single period only if the overall number of required technicians does not exceed the K, i.e., if

$$\sum_{e \in F} a_e + \sum_{u \in V(F)} a_u \leq K$$

The network hosts several optical connections that are used to realize virtual connections or even virtual networks of clients. These connections are represented by a set P of simple paths in G. These paths are required to remain fixed during the network upgrade, that is, they must not be rerouted.

Due to technological incompatibilities, a path is operative only when its links either all have technology A or all have technology B. When some links in the path have A, while some others have B, the path is disrupted. More formally, given a path $p \in P$ and a time period $t \in T$, we say that p is *active* in period t if all of its links have the same technology (i.e., either A or B). Otherwise, we say that p is *disrupted* in t. Denoting by r_p is the earliest period in which all links of p have technology B and by l_p is the latest period in which all links of p have technology A, the overall disruption time of path p is equal to $d_p = r_p - l_p - 1$.

The *WDM Fiber Replacement Scheduling Problem (WDM-FRS)* consists in finding a schedule assigning each link (upgrade) to a period in the planning horizon T and respecting the workforce budget K in all periods, such that the sum of the disruption d_p over all paths is minimized.

It has been shown in [2] that this problem is strongly NP-hard even if the underlying fiber network G is as simple as a star. Even more, it is conjectured that even schedules exceeding the minimal disruption by only a constant factor cannot be computed in polynomial time in general.

2.1 An example

We provide a very simple example of the WDM fiber replacement scheduling problem to clarify some of the concepts that we have introduced so far. Suppose we must upgrade the technology of the network represented in Figure 1, made up of 4 nodes and 3 links, such that the total disruption of the 3 paths p_1, p_2, p_3 is minimal.

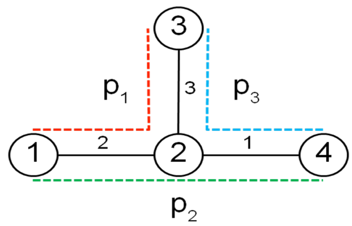

Figure 1 A simple network to be migrated

We assume that $K = 6$ technicians are available and that the numbers of technicians required to update the links are $a_{12}=2$, $a_{23}=3$, *and* $a_{24}=1$. Each node requires a single technician.

In Table 1, we represent two feasible schedules, denoted by S1 and S2. The schedule S1 is made up of 3 time periods, while S2 is made up of 2 periods. Schedule S1 updates link (1,2) in period 1, (2,3) in 2 and (2,4) in 3, while S2 updates (1,2) and (2,3) in 1 and (2,3) in 2. While S1 entails a total disruption of 4, S2 makes a better use of the technicians and reduces the disruption to 2. In the table, we show for each schedule: a) the technology installed on each link in each time period; b) the number of technicians required in each period; c) whether a path is disrupted; d) the overall disruption associated with the schedule.

Period	Technology on link			Workload	Path disruption		
	(1,2)	(2,3)	(2,4)		p_1	p_2	p_3
S1							
0	A	A	A	-	no	no	no
1	B	A	A	4	yes	yes	no
2	B	B	A	5	no	yes	yes
3	B	B	B	3	no	no	no
S2							
0	A	A	A	-	no	no	no
1	B	A	A	6	yes	no	yes
2	B	B	A	5	no	no	no

Table 1 Two examples of schedules

First, note that the overall network cannot be upgrade in a single time period, since this would require 10 technicians in total (2+3+1=6 technicians for the links and 4 technicians for the nodes), exceeding our budget of 6. Hence, we need at least two periods to complete all upgrades, which causes some unavoidable disruption.

Consider for example time period 1 of schedule S1: link (1,2) is upgraded to technology B, while the other two links maintain technology A. This means that the paths p_1 and p_2 are disrupted, because they both have one link with A and one link with B. The path p_3 is instead active, because all of its links have the same technology, namely A. In period 1, we thus experience two periods of disruptions: one for p_1 and one for p_2. We note that the upgrade of link (1,2) requires 4 technicians in total (2 for the link and 2 for the end nodes), keeping two technicians idle. As a better alternative, we could act like in schedule S2, where we update (1,2), (2,4) in the same period, using all the technicians (2+1 for the links and 3 for the nodes). This joint update reduces the overall disruption time of the schedule S2 by 2.

3 An integer linear programming formulation

In order to solve WDM-FRS, we propose an Integer Linear Programming formulation (ILP), based on time-indexed decision variables.

Variables. Our ILP formulation uses the following groups of binary variables:

- variables x_e^t to indicate whether link $e \in E$ is based on technology B in period $t \in T$,
- variables $y_p^{t\sigma}$ to indicate whether path $p \in P$ is active on technology $\sigma \in \{A,B\}$ in period $t \in T$, and
- variables z_i^t to indicate whether element (node or link) $i \in V \cup E$ requires technicians in period $t \in T$.

To simplify the notation, we also introduce additional (artificial) binary variables d_p^t indicating whether path $p \in P$ is disrupted in period $t \in T$.

Constraints. First of all, we need one constraint for each path $p \in P$ and time period $t \in T$ to express that p is either active in one of the two technologies or disrupted:

$$d_p^t + y_p^{tA} + y_p^{tB} = 1 \qquad p \in P, t \in T$$

Then we need two constraints for each link $e \in E$ to model that each link has technology A in the first period $t=0$ and must have technology B in the last period $t=|T|$:

$$x_e^0 = 0 \qquad e \in E$$
$$x_e^{|T|} = 1 \qquad e \in E$$

In order to establish the relation between the link technology variables and the path activation variables, we need two constraints for each path $p \in P$, each link $e \in p$ in the path, and each time period $t \in T$:

$$y_p^{tA} \leq 1 - x_e^t \qquad p \in P, e \in p, t \in T$$
$$y_p^{tB} \leq x_e^t \qquad p \in P, e \in p, t \in T$$

These ensure that path p is active in period t and technology A or B only if none or all of its links have been upgraded by period t, respectively.

Finally, to model the need for technicians, we introduce three groups of constraints. The first one models that technicians are required at link $e \in E$ in period $t \in T$ if and only if the technology of the e is upgraded in period t:

$$x_e^t - x_e^{t-1} = z_e^t \qquad e \in E, t \in T$$

The second group of inequality ensures that if a link $e \in E$ is upgraded in period $t \in T$, then technicians are required at its end nodes:

$$z_e^t \leq z_i^t \qquad e \in E, i \in e, t \in T$$

Finally, the third group of constraints ensures that the number of technicians required in each period does not exceed the given budget:

$$\sum_{i \in V \cup E} a_i z_i^t \leq K \qquad t \in T$$

Objective function. The objective is to minimize the total disruption of paths over the entire time horizon, i.e.:

$$\min \sum_{p \in P} \sum_{t \in T} d_p^t$$

3.1 Observations about the ILP

We remark that the artificial variables d_p^t can be eliminated from the model using equalities (2). Also, note that the constraints above imply that each link can change its technology only once and only from type A to type B during the planning horizon, i.e.,

$$x_e^t \geq x_e^{t-1} \qquad e \in E, t \in T$$

and that each path can be active in at most one technology in each period, i.e.,

$$y_p^{tA} + y_p^{tB} \leq 1 \qquad p \in P, t \in T$$

As presented so far, the above formulation is not useful in practice. Its linear programming relaxation allows for a trivial fractional solution that uniformly upgrades a fraction of *1/T* of each link in each time period, which leads to a total disruption of zero. This in turns makes the solution of larger problems hopeless, as it causes the generation of huge branch-and-bound trees.

However, there is a simple remedy: Exploiting the fact that the total disruption of a schedule does not change if we revert or translate the schedule in time, we may - without loss of generality - fix the upgrade of a single fiber link $e' \in E$ to a particular time period *t'*. The effect of this simple fixing is that it is no longer helpful to spread the upgrade operations evenly throughout the entire time horizon. Instead, it becomes beneficial to upgrade those links that have paths in common with *e'* in a time period as close as possible to *t'* to reduce the disruption of the common paths. In order to ensure that any feasible schedule can be transformed (via translation or reversion) into a schedule that upgrades link *e'* in period *t'*, it is sufficient to extend the planning horizon to at most *3/2 |T|* periods and fix the upgrade of the chosen link *e'* to period *t'*.

Our experiments show that this altered model is indeed useful in practice. Of course, the choice of the fixed link *e'* affects the strength of the linear relaxation of the model and, thereby, also the practical efficiency of the algorithm. Our experiments suggest that it is reasonable to use a link that belongs to many paths in *P*.

4 A branch-and-bound algorithm

We implemented a branch-and-bound algorithm (c.f. [4]) to solve the presented integer linear programming formulation of WDM-FRS.

In order to further strengthen the linear programming relaxation of the formulation and to speed up the overall solution process, we add several types of valid inequalities – so-called cutting planes [4] – to the model. One class of inequalities is based on analyzing the minimal disruption that is unavoidable on small subsets *P'* of the given paths *P* (such as all subsets of at most 3 paths). The inequalities of this class are added to the initial integer linear programming formulation before the start of the branch-and-bound process. Another class of inequalities is derived via projection from a very high-dimensional extension of the presented formulation, which employs additional variables to describe the order in which the links are upgraded. The inequalities in this class are so-called Benders cuts (c.f. [4]) and generated dynamically during the execution of the solution process. For a thorough discussion of these inequalities, their mathematical properties, and the corresponding generation procedures we refer the reader to [2].

In our initial experiments we observed that the standard heuristics and search strategies used commonly in branch-and-bound algorithms do not perform well for the integer linear programming formulations presented above. Thus, we also implemented a problem specific heuristic to quickly compute high quality integer solutions from the fractional solutions of the linear programming relaxation and a problem specific branching rule to decompose the solution space more effectively.

Our heuristic is applied multiple times during the algorithm's initial cutting plane phase and at each 5th node of the branch-and-bound tree. It works in two phases as follows: In the first phase, we build a good initial schedule from the current branch-and-bound node's fractional solution. Given the values x_e^t of this fractional solution, we determine for each threshold value *M* in *{0.2, 0.4, ... , 1}* and for each link $e \in E$ the so-called threshold period $t_e(M)$, which is the smallest period for which x_e^t is at least *M*. For each *M*, these values $t_e(M)$ define a (partial) order on the links. Scheduling the links in this order as early as possible, we obtain a feasible schedule for each *M*. The order leading to the best schedule defines our initial schedule. If this schedule is better than the best schedule known so far, we continue with the second phase.
Using local optimization techniques, we then try to iteratively improve this order (and the resulting schedule). In each iteration, we consider all orders that can be obtained from the current one by changing the position of a single link. The order leading to the best schedule (again, scheduling the links according to this order as early as possible) then becomes the reference order for the next iteration. This process repeats as long as improving orders are found. The best schedule found this way finally is adjusted to ensure that link *e'* is scheduled in period *t'* and returned.

Our problem specific branching rule works as follows. For each fractional variable x_e^t in the current branch-and-bound node's fractional solution, we compute a branch score value s_e^t. The formula for s_e^t favors links $e=(u,v)$ requiring many technicians $a_e+a_u+a_v$, and carrying many paths, time periods that are close to the reference time *t'*, and edges with fractional values x_e^t close to 0.5. The variable x_e^t with the highest score is then chosen as the branching variable and two new branch-and-bound nodes enforcing $x_e^t = 0$ and $x_e^t = 1$, respectively, are created. If all variables x_e^t are integer, our branching rule directly creates and evaluates the schedule defined by these variables and prunes the branch-and-bound tree, ignoring the fact that some of the other (artificial) variables still may be fractional. This way, we avoid many irrelevant branching operations that would be necessary to find fully integer feasible solution with general purpose branching rules in such cases.

5 Computational results

We have implemented the model given above using SCIP 3.0 [8] with CPLEX 12.4 [9] as LP solver. Additionally, we have added valid inequalities to strengthen the LP relaxation, as well as cutting planes obtained from a Benders decomposition, as described in [2]. All experiments have been conducted running on a single core of a PC with an AMD Phenom II X6 1090T processor and 8GB of RAM.

Table 2 lists the problem instances that we have considered in our experiments.

Instance	|V|	|E|	|P|	Solution
Han10	10	10	32	62
Fra20	20	22	51	59
Stu25	25	28	52	135
Des30	30	37	60	140

Table 2 Problem instances

In Table 3, we report the lower bound after the root node of the branch-and-bound tree, the number of nodes in the branch-and-bound tree, and the time needed until optimality was proven for the different test instances. The "Benders cuts" column indicates how many Benders cuts were added to the LP in the root node. All instances are subnetworks of the "full" instance comprising 64 nodes, 84 links, and 105 paths. The full instance is still too demanding for our solution approach, we therefore do not report results for it. The addition of the Benders cuts seems advisable for reasonably sized instances. For the instances Stu25 and Des30, only half as many nodes are explored in the branch-and-bound tree.

Instance	Benders cuts	Root bound	B&B nodes	Time (hh:mm:ss)
Han10	0	49.0	215	00:00:04.91
Han10	66	51.0	179	00:00:07.34
Fra20	0	45.7	409	00:00:41.11
Fra20	169	47.0	352	00:00:48.78
Stu25	0	78.2	15314	00:29:45.51
Stu25	628	84.6	8334	00:20:20.59
Des30	0	75.2	23976	01:04:58.17
Des30	555	81.8	11799	00:40:59.47

Table 3 Results for the non-uniform instances

Acknowledgement

This work was partially supported by the *German Federal Ministry of Education and Research* (BMBF) via the research project *ROBUKOM* [3].

6 References

[1] Bley, A.; D'Andreagiovanni, F.; Hanemann, A.: Robustness in Communication Networks: Scenarios and Mathematical Approaches. Photonische Netze 2011. Berlin · Offenbach: VDE-VERLAG, 2011

[2] Bley, A.; D'Andreagiovanni, F.; Karch, D.: Scheduling link replacement in networks. Preprint 2013-01, Institute of Mathematics, TU Berlin

[3] Koster, A.M.C.A.; Helmberg, C.; Bley, A.; Grötschel, M.; Bauschert, T.: BMBF Project ROBUKOM: Robust Communication Networks. Euro View 2012. Berlin · Offenbach: VDE-VERLAG, 2012

[4] Nemhauser, G.L.; Wolsey, L.A.: Integer and Combinatorial Optimization, Wiley, 1988

[5] Nokia Siemens Networks GmbH & Co KG, Munich, Germany, http://www.nokiasiemensnetworks.com/

[6] Piger, S.: Fliegender Wechsel, Migration der DWDM-Technik im X-WiN. DFN Mitteilungen. Vol. 82, pp. 8–11, 2012

[7] Theimer, T.: Towards a scalable and flat IP core network. ECOC 2009 Vienna. Berlin · Offenbach: VDE-VERLAG, 2009

[8] SCIP Optimization Suite, http://scip.zib.de

[9] IBM ILOG CPLEX, http://www.ibm.com/