Mathematics-in-Industry Case Studies Journal, Volume 1, pp. 49-65 (2009)

Wavelength Assignment in Optical Network Design

Brendan Farrell * Yi Huang † Mark Iwen ‡ Ting Wang § Lisa Zhang ¶ Jintong Zheng ‖

Abstract. We consider a flexible greedy approach to wavelength assignment in an optical network with the goal of minimizing the cost incurred by wavelength conversions and fiber deployment. The greedy approach processes demands one by one in a certain order and makes a locally optimal choice for each demand. We address several heuristics for creating desirable demand orderings, including a random ordering, as well as a hybrid method that begins with a graph coloring algorithm. One of the primary strengths of our heuristics is that they are both simple and flexible. Hence, additional practical engineering and cost constraints can be easily incorporated into the approach. An empirical evaluation shows that our greedy approach works well on real-world networks under realistic demand loads.

Keywords. optical network design, discrete optimization, wavelength assignment, greedy algorithm, graph coloring

1 Introduction

Modern optical networks provide efficient information transport on the order of terabits. This is realized by the cutting-edge technology of Dense Wavelength Division Multiplex (DWDM). In this setting, an optical fiber is partitioned into a large number of wavelengths, and traffic demands sharing a common fiber are transported on distinct wavelengths. Typically, abundant dark fibers are already installed underground. However, activating or deploying these fibers to create high-speed and high-capacity networks is expensive. Optical equipment for backbone networks in the US cost in the order of hundreds of millions of dollars. Hence, there is a great deal of interest in modeling and exploring algorithmic solutions for designing such networks, with the objective of cost optimization.

*Institut für Mobilkommunikation, Technische Universität, Berlin, Germany, brendan.farrell@mk.tu-berlin.de
†Liquid Crystal Institute, Kent State University, nwong@lci.kent.edu
‡Institute for Mathematics and its Applications, University of Minnesota, Minneapolis, iwen@ima.umn.edu
§Department of Mathematics, University of Michigan, Ann Arbor, wting@umich.edu
¶Mathematical and Algorithmic Sciences Research Center, Bell Labs, ylz@research.bell-labs.com
‖Department of Mathematics, University of Delaware, zhengj@math.udel.edu

Unfortunately, optical network design is highly complex from the perspective of combinatorial optimization as well as software engineering. These problems were addressed in the practical context in [7]. In particular, [7] presents a collection of design tools developed at Bell Labs for the purpose of optimizing backbone networks from major US and European carriers.

We begin with an overview of optical network design. There are many levels of abstraction to model an optical network. This includes the entire spectrum from the high-level design (i.e. specification of traffic routes and wavelengths) to the simulation of optical signals (i.e. the computation of optical quantities such as signal-to-noise ratio, dispersion, and non-linear phase shift). All components are closely interconnected. However, since each component requires drastically different techniques (e.g., discrete optimization for the high-level design and solving partial differential equations for simulating optical signals) a commonly accepted approach is to decompose the entire problem into relatively independent and more manageable pieces.

We focus on the algorithmic issues that arise from the high-level design, which addresses two basic issues, routing and wavelength assignment. Routing aims to find a path of fibers between the source and destination nodes of each demand; wavelength assignment aims to find available wavelengths within each fiber to carry the requested bandwidth of the demand. Both routing and wavelength assignment can be formulated as graph-theoretic problems in their cleanest mathematical abstraction. However, even in these simple forms, routing contains classic NP-hard problems such as edge-disjoint paths, congestion minimization and buy-at-bulk network design. As we shall see later, wavelength assignment has a close association with vertex coloring. All these problems are not only hard to solve optimally, but also hard to approximate.

To make things worse, the practical aspect of routing and wavelength assignment has an overwhelming amount of details in engineering constraints and cost specification. The source of the engineering constraints can be both service providers and equipment vendors. For example, fiber connections in the network may be of heterogeneous types, some of which may be able to carry more wavelengths than others; a demand may request specific nodes, edges or wavelengths to use or to avoid; a node may have a degree bound (i.e. an upper bound on the number of incident fibers that actively carry traffic); a demand path may be restricted by hop count or by distance limit; a link may only be able to carry a certain subset of wavelengths; a node may impose restrictions on the wavelengths going through it due to the equipment the node has. In addition to these constraints that affect the feasibility of a solution, how to price a design can also be messy. Roughly speaking, the total equipment cost for activating a set of fibers to carry traffic is proportional to the *number* of such fibers. However, the exact cost also depends on the fiber type, its physical length, the number of wavelengths in use, and the highest active wavelength in use, etc. These additional factors have the second-order effect on the total equipment cost. Therefore, to a first approximation, the number of active fibers is linearly proportional to the total equipment cost of the network.

In this study, we separate routing from wavelength assignment and focus on the latter in isolation. This separation is motivated by the algorithmic difficulty of both problems alluded to before and is consistent with the tools perspective of [7]. We choose to work on a clean formulation of wavelength assignment that captures the core of the problem. We also choose a simple heuristic approach to solve the core problem. This approach gives us the flexibility to easily accommodate the messier engineering requirements as needed, and it allows the ever-changing pricing of optical components to drive the optimization.

We have considered other approaches to tackle the problem. Here are several examples. We could formulate the core problem as an integer linear program (ILP) and attempt to solve it with a commercial solver like CPLEX. This approach works well on very small problem instances. Thus, optimal solutions obtained by CPLEX can sometimes be used to judge the quality of heuristic solutions (as we do in Section 4). However, drawbacks of an ILP approach include both lack of scalability and inflexibility in accommodating extra constraints. Other approaches have similar drawbacks. Algorithms for wavelength assignment/demand routing on specific types of networks [20] generally either don't successfully extend to different classes of networks, or don't easily accommodate additional constraints. Likewise, viewing wavelength assignment in a planning context and encoding it as either a boolean satisfiability [10] or constraint satisfaction [4] problem was considered. Existing solvers [8, 5] could then be used to generate wavelength assignments. However, as before, these encoding-based approaches have drawbacks related to both scalability and to difficulties handling realistic engineering constraints. Hence, we focus on investigating simple, scalable, and flexible greedy solution methods.

We now state the wavelength assignment problem in terms of two abstracted models. To begin, we are given a network modeled by a simple undirected graph $N = \langle V, E \rangle$, and a set of demands $P_1, \ldots, P_d$. Each demand d_i sends one wavelength of traffic along a specified path P_i. The task of wavelength assignment is to assign these paths to wavelengths from the range $[1, \mu]$ where μ is the fiber capacity. Given the demand paths, the minimum number of fibers necessary on link e is $f(e) = \lceil L(e)/\mu \rceil$, where $L(e)$ is the number of demand paths traversing link e.

In the first of our two models, which we call **Min-Fiber**, each demand path is assigned *one* wavelength from beginning to end, with no wavelength conversion. If we let $N_e(w)$ denote the number of times wavelength w is used on link e due to an assignment, then $\max_w N_e(w)$ is the number of fibers link e would have to deploy. We denote this quantity by $F(e)$. Note that $F(e)$ is necessarily at least $f(e)$. The objective of **Min-Fiber** is then

$$\min \sum_e F(e).$$

Note that a more general objective function could be minimizing a weighted sum $\sum_e \ell(e)F(e)$ for edge-dependent weights $\ell(e)$. As we shall see, our greedy heuristic is directly applicable to this generalization. However, for this paper we stick to minimizing the total *number* of fibers. For reasons alluded to before, minimizing the fiber count directly implies cost minimization.

In the second model, which we call **Min-Conversion**, we allow wavelength conversion along a demand path. In particular, we can potentially partition each demand path P_i into several subpaths and assign each subpath a distinct wavelength. The number of conversions is then the number of subpaths minus one. We refer to this quantity as $C(i)$. In this model, we require exactly $f(e)$ fibers deployed on link e, i.e. no extra fibers. The objective of **Min-Conversion** is then

$$\min \sum_{i=1}^{d} C(i).$$

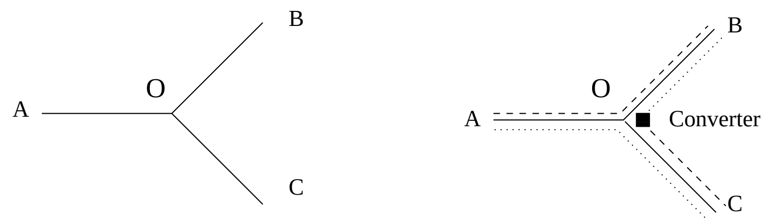

Figure 1: *Network solution for* **Min-Conversion**

Figure 1 illustrates a very simple example. There are four network nodes: A, B, C and O; three demands (i.e. routes): AOB, AOC and BOC; and we are allowed two wavelengths per fiber. We begin with all wavelengths available. If the first demand is AOB, it is routed completely on one wavelength. Then AOC is routed completely on the second wavelength. There is now one wavelength available on both AO and OB, but these are different wavelengths. A converter is, therefore, placed at O for the demand BOC.

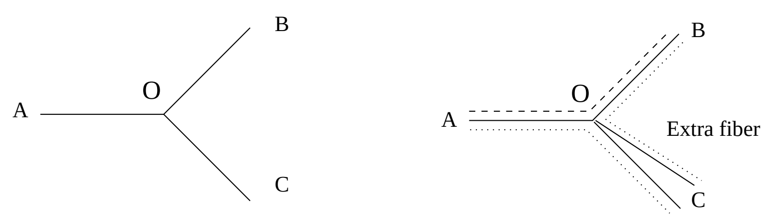

Figure 2: *Network solution for* **Min-Fiber**

Figure 2 shows the **Min-Fiber** solution for the same simple network as in Figure 1. Here, however, rather than adding a converter for the demand BOC, additional fiber is added to the link OC.

To be useful in practice a wavelength assignment algorithm must be both robust and easily adaptable to additional unforeseen physical, economic, and engineering constraints imposed by the service provider. For example, a service provider may wish to prioritize the use of some wavelengths over others, or to use wavelengths in bundles. Therefore, we purposefully seek a solution technique that is both simple and adaptable. In this paper we introduce and empirically evaluate several heuristic greedy solution techniques for **Min-Fiber** and **Min-Conversion**. Our solution techniques are not only simple and adaptable, but also quite accurate for instances that come from real-world networks, as we shall see.

2 Preliminaries and Related Work

In this section we give a review of related work. In doing so we both: (*i*) explain why **Min-Fiber** and **Min-Conversion** are hard, and (*ii*) motivate our choice of greedy solution techniques.

2.1 Difficulty of Min-Fiber and Min-Conversion

For a graph $G = \langle V, E\rangle$ defined on a vertex set V and an edge set E, a **coloring** for G is an assignment of colors to the vertices such that no vertices sharing a common edge are assigned the same color. The **chromatic number** of G, $\chi(G)$, is defined to be the smallest number of colors that a coloring of G must have. Determining $\chi(G)$ for an arbitrary graph G is NP-hard. Furthermore, even approximating $\chi(G)$ to within a factor of $n^{1-\epsilon}$ is also hard, where n denotes the number of vertices.

To demonstrate why **Min-Fiber** and **Min-Conversion** are difficult, we relate them to vertex coloring. Given a wavelength assignment instance defined on a network $N = \langle V_N, E_N\rangle$ and a set of demand paths $P_1, \ldots, P_d$, we create a **demand graph** $D = \langle V_D, E_D\rangle$. The vertices in D correspond one-to-one to the demand paths. In particular, each vertex $v_i \in V_D$ corresponds to the demand path P_i. Furthermore, $(v_i, v_j) \in E_D$ if and only if $P_i \cap P_j \neq \emptyset$. An example demand graph is shown in Figure 3.

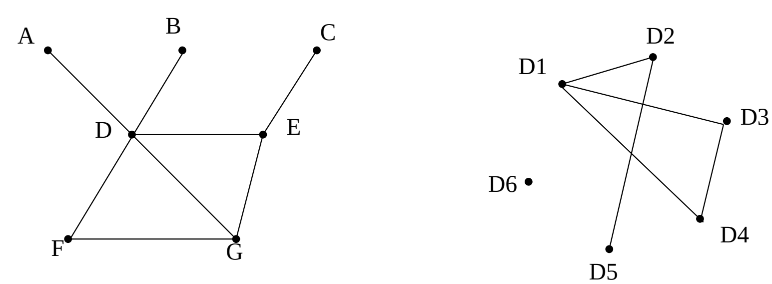

Figure 3: *The left graph is the network with demand paths $D1 = ADE$, $D2 = ADG$, $D3 = FDE$, $D4 = FDA$, $D5 = BDG$ and $D6 = CEG$. The right graph is the demand graph.*

Consider the **Min-Per-Link** problem: assuming all links are given the same number of fibers, how many fibers are required for an optimal wavelength assignment? The **Min-Per-Link** problem is clearly easier than the **Min-Fiber** problem, and solving it is equivalent to finding the chromatic number of the associated problem's demand graph. Standard hardness results for the chromatic number problem [11] (i.e. determining $\chi(D)$) suggest the difficulty of **Min-Per-Link**. Given that

Min-Per-Link is hard, it is not surprising that **Min-Fiber** is also difficult. For a formal proof of the inapproximability of **Min-Per-Link** see [1] and for the inapproximability of **Min-Fiber** see [2].

The inapproximability of **Min-Conversion** is less studied, though the problem is NP-hard even for simple topologies such as rings and trees [2]. The following informal argument indicates the NP-hardness of the conversion problem. Recall that d is the number of demands we must satisfy. We begin by noting that we can use a solution method for **Min-Conversion** to solve **Min-Per-Link**. Suppose we can solve **Min-Conversion** in polynomial-time as long as each edge has at least $\max_{e \in E_N} f(e)$ fibers (i.e. we allow additional fibers per link – in the next paragraph we will demonstrate that this does not influence our argument). Clearly no conversions are required to solve such a **Min-Conversion** instance if the number of fibers per link is chosen to be large enough. Thus, we can solve **Min-Per-Link** by solving $O(d)$ **Min-Conversion** problems (with the same network demands), where the number of fibers per link is varied from $\max_{e \in E_N} f(e)$ to d. The smallest number of fibers per link requiring zero conversions is our answer.

We finish by noting that we may solve any **Min-Per-Link** problem by considering an equivalent demand graph derived network with $O(d^3)$ links. Furthermore, by adding an additional $O(\mu \cdot d^4)$ trivial (i.e. single-link) demands we can increase its minimum number of required fibers per link without influencing the number of conversions necessary to solve the same problem. Therefore, we can indeed test any given number of fibers per link as a solution to a given **Min-Per-Link** problem by considering a conversion-equivalent demand graph derived network **Min-Conversion** problem. The upshot is that we can solve the **Min-Per-Link** problem in polynomial-time if we have a polynomial-time solution method for **Min-Conversion**. The difficulty of **Min-Conversion** follows.

2.2 When the Going Gets Tough, the Tough Get Greedy

The inapproximability and hardness of optimally solving **Min-Fiber** and **Min-Conversion** mean that we have to consider a potentially suboptimal approach. In the absence of polynomial time algorithms with a constant approximation guarantee, we fall back on heuristic greedy solution techniques. Greedy heuristic solution methods have been fruitfully applied to many NP-hard problems including vertex coloring [19], SAT solving [16, 3], set covering [17], strip packing [13], and generalized planning [9] problems.

Generally greedy techniques perform surprisingly well. Many greedy/heuristic techniques have been applied to the graph coloring problem with favorable results [12, 18, 14, 6]. See [15] for a survey of many more. Perhaps most inspiring for us is the work of Turner [19], who demonstrated that a simple greedy vertex coloring algorithm performs optimally on most graphs. The strong empirical results associated with greedy SAT solvers [16] are also encouraging, especially considering that propositional satisfiability is an NP-complete problem. For these reasons we employ greedy priority wavelength assignment methods along the lines of [13].

3 Our Greedy Approach

Without assuming any additional properties for the network, finding the least expensive **Min-Fiber/Min-Conversion** network design that satisfies all demands currently necessitates the use of a superpolynomial-time algorithm. Of course, such algorithms are generally computationally infeasible. Thus, we have taken a *greedy* approach to solving **Min-Fiber/Min-Conversion**.

Generally, a greedy optimization algorithm iteratively makes locally optimal assignments in the hope of reaching a globally optimal solution. In this case, we order our demands according to several heuristics and then seek to make locally optimal wavelength assignments for them, one at a time. Our general algorithm is the following.

Algorithm 3.1 *Optimize Design:*

1. *Let* $P_{\pi(1)}, \ldots, P_{\pi(d)}$ *be an ordering of the demand paths.*
   - *For* $i = 1, \ldots, d$*: find locally optimal solution for* $P_{\pi(i)}$.
2. *Randomly perturb the ordering* π *and let* $P_{\pi'(1)}, \ldots, P_{\pi'(d)}$ *be the resulting ordering.*
   - *For* $i = 1, \ldots, d$*: find locally optimal solution for* $P_{\pi'(i)}$.

In our simulations, the random perturbation of step 2 was implemented by randomly permuting sets of adjacent indices. Note that both steps 1 and 2 can be repeated multiple times. If so, the best ordering of the trials is recorded.

The Optimize Design algorithm has essentially two components: ordering and locally optimal solution. The latter component is greedy as it does its best for each demand in the currently considered order. We discuss this greedy step in Section 3.1. Of greatest importance here is that as each ordered demand is treated by the greedy algorithm, assignments are made that change the set of available wavelengths for subsequently ordered demands. Therefore, the order in which the demands are treated is crucial for a good final design. Indeed, it can be shown that there is always some ordering which leads to an optimal solution [21]. Of course, there is no known efficient method which is guaranteed to determine such an ordering (see Section 2.1 above). Thus, we order our demands heuristically and/or randomly.

As we can see, the above framework is extremely flexible. For example, it can be tuned to accommodate constraints such as avoiding forbidden wavelengths or enforcing desired wavelengths at a link/node, etc. Different objective functions can also be swapped in, as needed.

3.1 Greedy Solution Given Ordering

While there are possible combinations of networks, demand sets and demand orderings for which the greedy approach would return a result far inferior to an optimal solution, we are considering realistic networks and always consider a handful of orderings. Under such conditions, the greedy approach generally yields a good solution. Importantly, the greedy algorithm does so relatively quickly. We have two greedy routines, *Greedy-Conversion* and *Greedy-Fiber*, which respectively minimize the number of conversions and the number of deployed fibers. Each takes a demand as input and returns a wavelength assignment path for that demand. In the algorithms that follow, a demand path P consists of links $\{e_1, ..., e_n\}$.

Algorithm 3.2 *Greedy-Conversion(P)*

1. *Start at link* e_1*. Assign the wavelength* $w \in [1, \mu]$ *that is available on the greatest number of consecutive subsequent links. If this is* k *links on wavelengths* $\tilde{w}$*, assign* $\tilde{w}$ *to the subpath* $\{e_1, e_k\}$*.*

2. *Repeat (1), starting at* e_{k+1} *until all links are treated.*

We note that *Greedy-Conversion* in fact finds a wavelength assignment with a minimum number of conversions for each demand path as ordered. But it happens that this assignment is also determined by treating the path greedily.

Algorithm 3.3 *Greedy-Fiber(P)*

1. *Determine the wavelength that is available on the greatest number of* $\{e_1, ..., e_n\}$*. Suppose it is* $\tilde{w}$*.*

2. *Add a fiber on the links where* $\tilde{w}$ *is not available. Update the network.*

3. *Assign wavelength* $\tilde{w}$ *to path* P*.*

3.2 Ordering

We propose four different orderings for our demands. We then process the demands in the given order using our greedy Section 3.1 methods. The first approach uses the *length* heuristic: demands are ordered according to the number of links that they travel, and the longest are prioritized. The second uses the *load* heuristic: each link is given a weight according to the number of demands that travel that link. The weights along each demand's route are summed to give the load, and demands with higher route loads are prioritized. The third ordering approach is *random*, in which the ordering is a random permutation of all the demands. The fourth approach uses graph coloring. This approach is more involved and is discussed in Section 3.3. We note here, though, that while its output is a set of assignments, it may also be viewed as returning the ordering that produces those assignments when the algorithm is applied.

Each of these four methods may be used, as well as in combinations, to create an ordering for the demands.

3.3 Demand Preprocessing via Vertex Coloring

Network fibers are assumed to each support μ wavelengths. Therefore, if it is possible to cover a problem's demand graph with μ colors, neighboring demands will be assigned different wavelengths (i.e. colors) and will not conflict on their common link(s). Since, in general, μ colors will not suffice to color the demand graph, we color as many demands as possible with μ colors using a fast coloring algorithm known to perform well "on most graphs" [19]. Each demand that is assigned a color is then routed, and we turn to the unrouted demands. These fall into two groups: those with only one fiber on each link and those with more than one fiber on each link. We set the first group aside, and treat the second group by creating its demand graph. This process is repeated as long as demands remain with at least one unused fiber. In this way we are able to assign a large number of wavelength routes using established graph coloring methods before turning to our demand ordering heuristics. Unprocessed demands are then handled globally by the length, load, or random approaches.

4 Empirical Evaluation

The problem instances that we study closely resemble those from major carriers in the US and Europe. Each instance specifies a network topology and a demand matrix which reflects the traffic estimate by the carriers. The networks are sparsely-connected and consist of a collection of inter-connected rings. The ring-based topology is prevalent in practice and is well documented. For example, [22] studies wavelength minimization over a tree of rings and proves constant approximation guarantees. The practical motivation comes from the need for failure protection. In the presence of a failed node or link, affected demands can then be routed along a backup path. We do not focus on the issue of protection in our study. For the purpose of wavelength assignment, we fix routes for the demands by computing either shortest paths or some longer paths that would pack in more demands. We also scale the entries in the traffic matrix to create varying demand loads on the active fibers. We remark that the purpose of this empirical evaluation is to test the greedy approach on "common" instances and see how it would help with real-life designs. Our focus is not on creating hard obscure instances.

Although for proprietary reasons, we cannot present real data exactly, Figure 4 is a perturbed network topology of one small instance we tested. As we can see, it is a planar graph with 20+ nodes and an average node degree slightly over 2. It consists of 6 inter-connected rings. The traffic demands originate from about 80 node-pairs. Nodes corresponding to large cities typically have more traffic originating from or terminating at them. The fiber capacity μ ranges from 20 to 50 in our tests.

We divide the networks that we considered into three groups, each with 3 instances. The first consists of small networks. For the most part, we know the optimal solution for these networks, because for sufficiently small instances either a commercial solver like CPLEX can yield an optimal

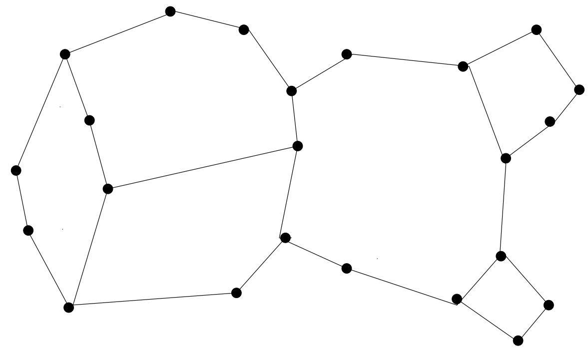

Figure 4: *A perturbed topology of an instance tested.*

solution or we were able to reason what an optimal solution looked like. We find here that all of our approaches quickly find the optimal solution. See Figure 5 for results on small and lightly loaded networks.

The second group is networks with a small number of nodes, but a heavy demand load on most links (i.e. such as about 80% required usage of available wavelengths). For some instances in this group we know the optimal solution, and for these cases we observe that taking the best result from our set of methods has returned the optimal solution. For instances in this group, for which we don't know the optimal solution, we find that the random ordering yields the best solution, where a few hundred orderings were tried. See Figure 6 for results on small networks with heavy demand loads.

For the group of networks consisting of a large number of nodes and with heavy saturation of most of the links, we find that the length and load approaches and the vertex coloring approach result in better solutions than the random ordering, where we significantly increased the number of orderings for this group than the previous two groups. However, since we do not know any of the optimal solutions for these networks, we do not know how close our solution is to optimal. The results are summarized in Figure 7 below.

The key observation is that for a small number of demands, random ordering outperforms our other heuristics. Yet for larger demand sets the length, load and vertex coloring approaches outperform the random approach. A plausible explanation is that in the smaller demand sets, all demands are of comparable size, and, therefore, there is not an inherent scale of complexity for the demands. For the large, heavily loaded networks, however, there are clear distinctions between heavy or light and long or short. Therefore the heuristic is more effective in the large network setting.

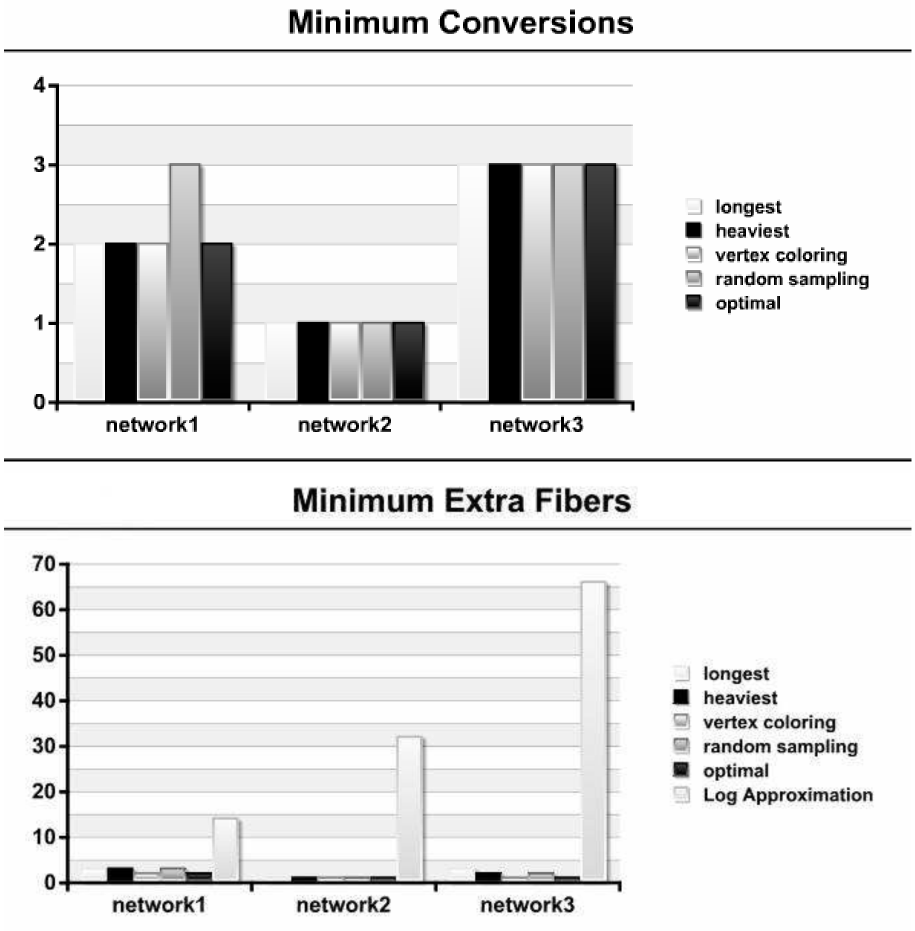

Figure 5: *Small Networks with Light Demand Loads.*

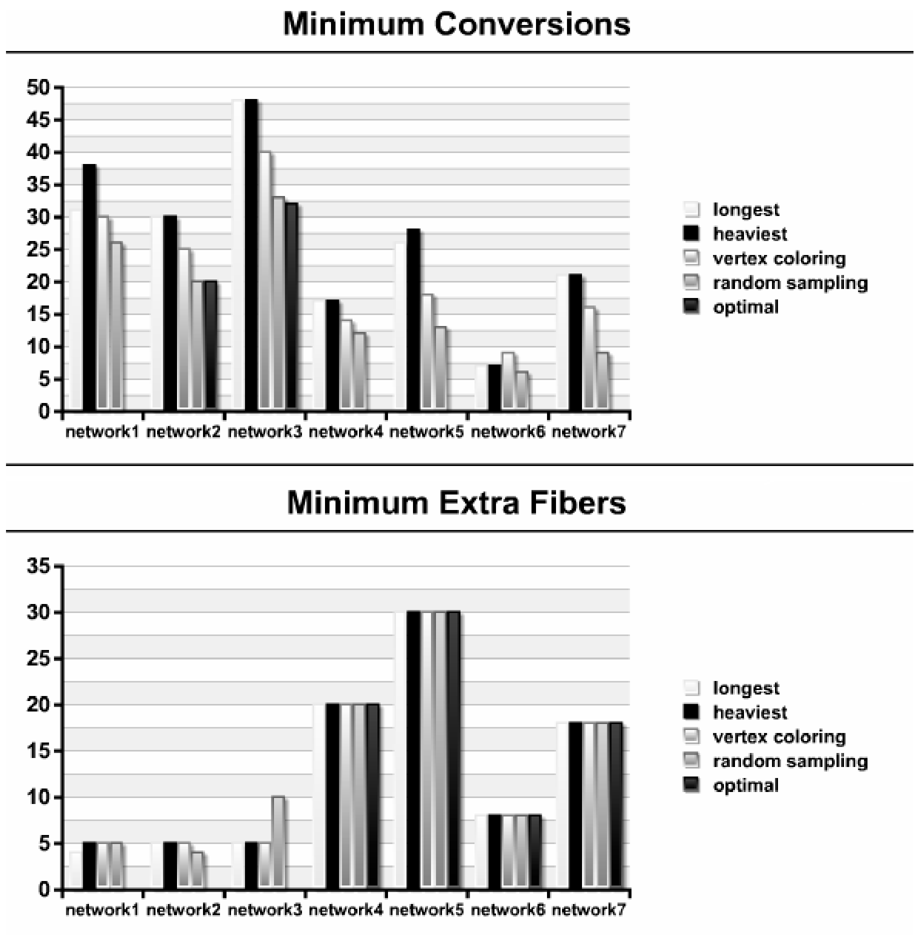

Figure 6: *Small Networks with Heavy Demand Loads.*

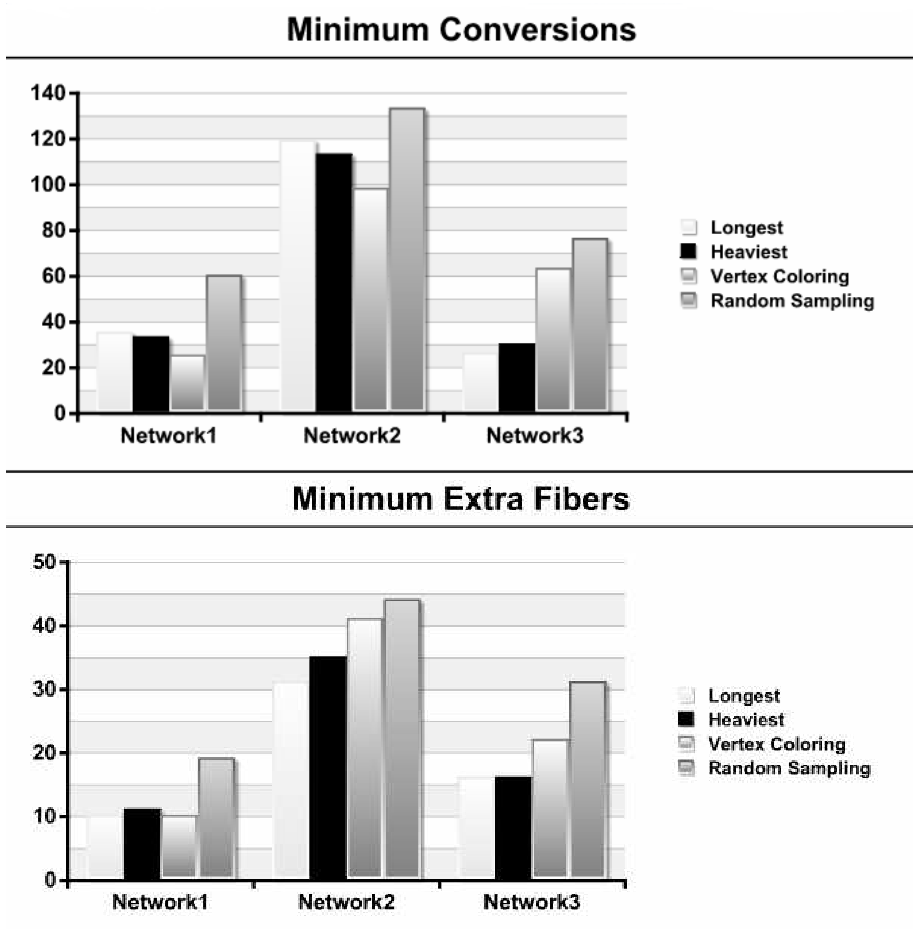

Figure 7: *Large Networks with Heavy Demand Loads.*

5 A New Direction

We have presented heuristics to find the minimum number of fibers or conversions. In the final section of this paper we consider a new direction that uses a combination of both objectives. A natural way to do this is to check how many fewer conversions are required if one fiber is added. To apply this idea, we list all the extra fibers we need to add according to **Min-Fiber**. We add one of them and determine the minimum number of conversions needed. We try each fiber on the list, and select the one that results in the greatest decrease in conversions. We repeat this routine until all fibers have been added. Then we view the results to make an allocation based on the cost of each outcome. Table 1 shows the effect of this algorithm for a large ring network.

The tradeoff between the number of deployed fibers and the number of conversions is dictated by the ratio of the equipment cost for activating a fiber and the cost of a conversion. This ratio is obviously product dependent, and a reasonable number can be from the range of $[10, 20]$. In the following we show our preliminary observation assuming that deploying 1 fiber is as expensive as 10 conversions. The last row of the table gives a cost estimate for different combinations. It clearly suggests that, in terms of cutting the cost, we should add 5 fibers to the specific links and use another 49 conversions to solve the remaining conflicts.

6 Conclusion

In this paper we considered the **Min-Fiber** and **Min-Conversion** optical network wavelength assignment problems. These problems deal with assigning wavelengths to network user demands in a fashion that minimizes the total number of deployed fibers or conversions, respectively. Given the expense of laying additional fibers or adding conversion equipment, these problems are of great commercial interest to optical network service providers.

Given the inherent difficulty involved with obtaining optimal solutions to either **Min-Fiber** or **Min-Conversion**, we propose using well-proven priority-based greedy heuristics as solution methods. Not only are greedy methods flexible to future constraints, but they have also been shown to work well on a plethora of other difficult (i.e. NP-hard) problems. Indeed, in keeping

Number of extra fibers	0	1	2	3	4	5
Number of conversions	129	109	90	79	60	49
Total cost	129	119	110	109	100	99
Number of extra fibers	6	7	8	9	10	11
Number of conversions	40	30	20	20	20	0
Total cost	100	100	100	110	120	110

Table 1: *Allocation and cost results for a Large Ring Network*

with the good reputation of greedy heuristics, our solution methods also perform well empirically on realistic wavelength assignment instances. As seen in Section 4, our greedy solution techniques all perform nearly optimally (i.e. within a factor of two or better) on all assignment problems for which the optimal solution is known. However, our study has not allowed us to draw a definitive conclusion as to which ordering is superior under what circumstances.

Perhaps most interestingly, we introduce a new mixed **Min-Fiber/Conversion** cost reduction model in Section 5. Preliminary tests indicate that our new mixed fiber/conversion cost reduction scheme creates wavelength assignments, conversion nodes, and additional fiber installations more cost effectively then either **Min-Fiber** or **Min-Conversion** solutions alone. Hence, our new mixed fiber/conversion model is potentially useful and can benefit from further exploration.

Acknowledgements

The authors are grateful to the IMA for the excellent work environment and support it provided during the workshop. The authors also wish to thank anonymous referees for insightful comments.

References

[1] M. Andrews and L. Zhang, *Minimizing maximum fiber requirement in optical networks*, Journal of Computer and Systems Sciences, Volume 72 Issue 1, pp. 118-131 (2006). 54

[2] M. Andrews and L. Zhang, *Complexity of wavelength assignment in optical network optimization*, INFOCOM 25th IEEE International Conference on Computer Communications Proceedings, Barcelona, Spain, pp. 1-12 (2006). 54

[3] G. Audemard and L. Simon, *GUNSAT: A Greedy Local Search Algorithm for Unsatisfiability*, International Joint Conferences on Artificial Intelligence, pp. 2256-2261 (2007). 54

[4] M. B. Do and S. Kambhampati, *Planning as constraint satisfaction: Solving the planning graph by compiling it into CSP*, Artificial Intelligence, Volume 132 Issue 2, pp. 151-182 (2001). 51

[5] S. Bouveret, F. Heras, S.de Givry, J. Larrosa, M. Sanchez and T. Schiex *Toolbar: a state-of-the-art platform for wcsp* http://www.inra.fr/mia/T/degivry/ToolBar.pdf, 2004. 51

[6] M. Caramia and P. Dell'Olmo. *Coloring graphs by iterated local search traversing feasible and infeasible solutions*, Discrete Applied Mathematics, Volume 156 Issue 2, pp. 201-217 (2008). 54

[7] C. Chekuri, P. Claisse, R. Essiambre, S. Fortune, D. Kilper, W. Lee, N. Nithi, I. Saniee, B. Shepherd, C. White, G. Wilfong and L. Zhang. *Design tools for transparent optical networks*, Bell Labs Technical Journal, Volume 11 Issue 2, pp. 129-143 (2006). 50, 51

[8] N. En and N. Srensson, *The MiniSat Page*, http://minisat.se/ 51

[9] J. Hoffmann and B. Nebel, *The FF Planning System: Fast Plan Generation Through Heuristic Search*, Journal of Artificial Intelligence Research, Volume 14, pp. 253-302 (2001). 54

[10] H. A. Kautz and B. Selman, *Planning as Satisfiability*, Proceedings of the Tenth European Conference on Artificial Intelligence (ECAI '92), Vienna, Austria, pp. 359-363 (1992). 51

[11] S. Khot and A. K. Ponnuswami. *Better inapproximability results for maxclique, chromatic number, and min-3lin-deletion*, Proceedings of the 33rd International Colloquium on Automata, Languages and Programming. Lecture Notes in Computer Science, pp. 226-237 (2006). 53

[12] M. Laguna and R. Mart, *GRASP for coloring sparse graphs*, Computational optimization and applications, Volume 19 Number 2, pp. 165-178 (2001). 54

[13] N. Lesh and M. Mitzenmacher, *Bubblesearch: a simple heuristic for improving priority-based greedy algorithms*, Information Processing Letters, Volume 97 Issue 4, pp. 161-169 (2006). 54

[14] T. Ono, M. Yagiura and T. Hirata. *A vector assignment approach for the graph coloring problem*, Proceedings of the Learning and Intelligent OptimizatioN Conference (LION 2007 II), Trento, Italy, 2007. 54

[15] P.M. Pardalos, T. Mavridou and J. Xue, *The graph coloring problem: A bibliographic survey*, Handbook of Combinatorial Optimization, D.-Z. Du and P.M. Pardalos (Editors), Volume 2, pp. 331-395 (1999). 54

[16] B. Selman, H. Levesque and D. Mitchell, *A new method for solving hard satisfiability problems*, Proceedings of the 10th National Conference on Artificial Intelligence (AAAI'92), pp. 440-446, San Jose, California (1992). 54

[17] P. Slavík, *A Tight Analysis of the Greedy Algorithm for Set Cover*, Proceedings of the 28th annual ACM Symposium on Theory of Computing (STOC '96), pp. 435-441, ACM Press, New York, NY, USA, 1996. 54

[18] M. A. Trick and H. Yildiz, *A large neighborhood search heuristic for graph coloring*, Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, Lecture Notes in Computer Science, P. V. Hentenryck, L. A. Wolsey (Editors), Volume 4510, pp. 346-360, Springer Verlag, (2007). 54

[19] J. S. Turner, *Almost all k-colorable graphs are easy to color*, Journal of Algorithms, Volume 9 Issue 1, pp. 63-82 (1988). 54, 57

[20] G. Wilfong and P. Winkler, *Ring Routing and Wavelength Translation*, Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 333-341 (1998). 51

[21] H. Zang, J.P. Jue and B. Mukherjee, *A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks*, Optical Networks Magazine, 2000. 55

[22] Z. Bian, Q. Gu and X. Zhou, *Tight bounds for wavelength assignment on trees of rings*, 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05), 2005. 57