IMA Final Report: Wavelength Assignment in Optical Network Design

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

Lisa Zhang ‖(Mentor)

August 17, 2007

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. Such high-speed and high-capacity networks are expensive. Typically, the optical 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.

Unfortunately, optical network design is highly complex, from messy engineering constraints to difficult combinatorial optimization issues. For example, to carry a traffic demand over the network, a network designer must specify a physical route as well as the particular wavelength that carries the demand. Both routing and wavelength assignment are nontrivial in its own right. 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 close

*Department of Mathematics, University of California, Davis. Email: farrell@ math.ucdavis.edu.

†Liquid Crystal Institute, Kent State University. Email: nwong@ lci.kent.edu.

‡Department of Mathematics, University of Michigan, Ann Arbor. Email: markiwen@ umich.edu.

§Department of Mathematics, University of Michigan, Ann Arbor. Email: wting@umich.edu.

¶Department of Mathematics, University of Delaware. Email: zhengj@math.udel.edu.

‖Mathematical and Algorithmic Sciences Research Center, Bell Labs. Email: ylz@research.bell-labs.com.

association with vertex coloring. For this project, we assume that the routes are given as input and we study wavelength assignment in isolation.

Specifically, we consider the wavelength assignment problem in the following two models. We are given a network, modeled by a simple undirected graph $N = \langle V, E \rangle$, and a set of demands $P_1, \ldots, P_d$ where 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 model, 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

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

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

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

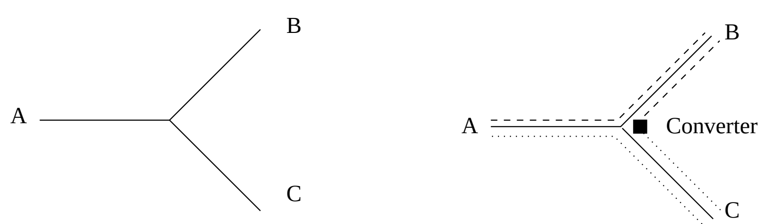

Figure 1: Network solution for **Min-Conversion**

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,

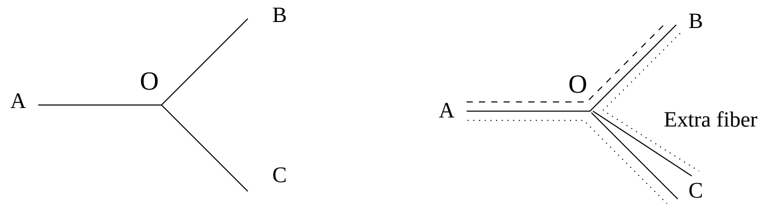

Figure 2: Network solution for minconversion

placed at O for the demand BOC. Figure 1. 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.

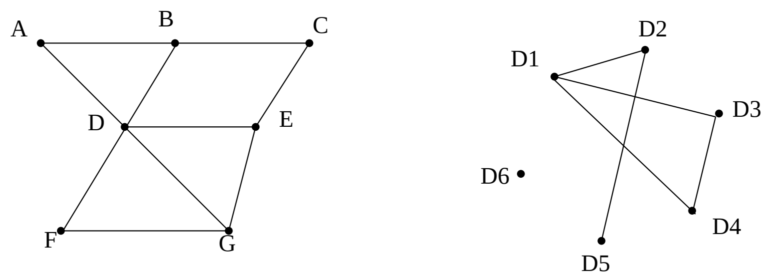

Figure 3: The left graph is the network with demand paths $D1 = ADE$, $D2 = ADG$, $D3 = FDE$, $D4 = FDA$, $D5 = BDG$ and $D6 = CEG$. The right graph is the demand graph.

Consider the **Min-Per-Link** problem: assuming all lings are given the same number of fibers, how many fibers are required for an optimal wavelength assignment? Standard hardness results for the chromatic number problem [5] (i.e., determining $\chi(D)$) suggest the difficulty of **Min-Per-Link**. For a formal proof of inapproximability along these lines see [2]. Given that **Min-Per-Link** is hard, it is not surprising that **Min-Fiber** is also difficult [1]. The formal complexity of **Min-Conversion** is less studies. However, the following argument gives certain indication of the hardness of the conversion problem. Even if an oracle provides us with optimal conversion placements on every link, finding the associated demand wavelength assignment will still require the associated demand graph to be colored optimally. The end result is that, unless we're willing to use exponential time algorithms, solving either **Min-Fiber** or **Min-Conversion** optimally is generally a hopeless task. In fact, even coming up with good approximations is too much to expect.

2.2 When the Going Gets Tough, the Tough Gets 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 guaranteed accuracy, we fall back on heuristic greedy solution techniques. Greedy heuristic solution methods have been fruitfully applied to many NP-hard problems including vertex coloring [9], SAT solving [7, 3], set covering [8], strip packing [6], and generalized planning [4] problems.

Generally greedy techniques perform surprisingly well. Perhaps most inspiring for us is the work of Turner [9], who demonstrated that a simple greedy vertex coloring algorithm performs optimally on most graphs. The strong empirical results associated with greedy SAT solvers [7] 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 [6].

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 requires calculating the cost of each possible wavelength assignment for the given demands. Of course, considering each possible design is infeasible, as the number of designs grows exponentially.

We have taken a *greedy* approach to solving **Min-Fiber/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. *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* $P_{\pi'(i)}$.

Note that both steps 1 and 2 can be repeated multiple times. If so, the best ordering of the trials are 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. We discuss this greedy step in Section 3.2. But when each demand is treated by the greedy algorithm, assignments are made that change the set of available wavelengths for subsequent demands. The order in which the demands are treated is, therefore, crucial for a good final design.

3.1 Ordering

We propose to try four different orderings. 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 3.4. 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 combination, to create an ordering for the demands.

3.2 Locally optimal solutions

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 additional 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 minimum number of conversions for each demand path. 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.*

3.3 Trade off

Once we have the algorithms to find the minimum number of fibers or conversions, we consider using a combination of the two. A natural way to do this is to check how many fewer conversions are required if one fiber is added. To apply this idea,

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

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. The table shows the effect of this algorithm for the "ring-50-70" network.

The last row of the table gives a cost estimate for different combinations with a reasonable assumption that 1 fiber is as expensive as 10 conversions. 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.

3.4 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. 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 there remain demands with at least one unused fiber. In this way we are able to assign a large number of wavelength routes before turning to the greedy algorithm. Unprocessed demands are then handled globally by the length, load or random approaches.

4 Empirical Evaluation

We divide the networks that we considered into three groups. The first consists of small networks of mostly active links. For the most part, we know the optimal solution for these networks. We find here that all of our approaches quickly find the optimal solution.

The second group is networks with a small number of nodes, yet still with a heavy demand load on most links, such as about 80%. 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.

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. 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 observations are summarized in the figure 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.

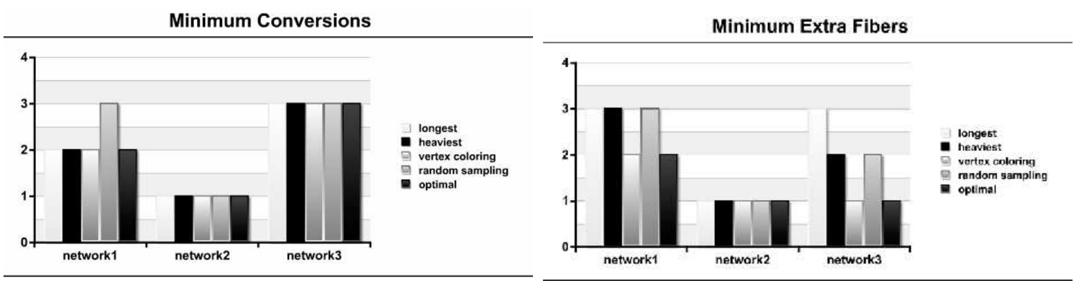

Figure 4: Networks in group 1.

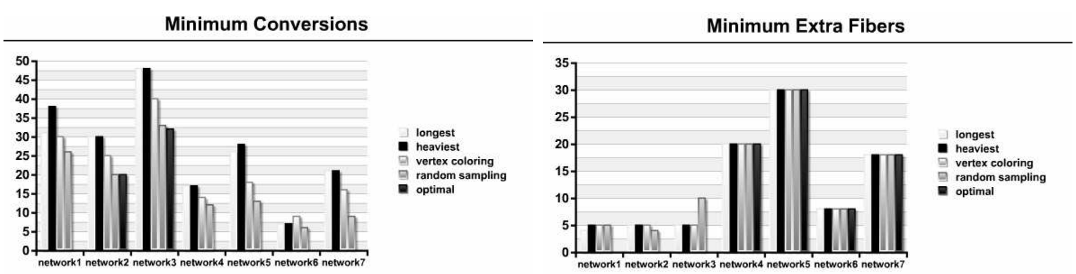

Figure 5: Networks in group 2.

5 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 which minimizes the total number of additional network fibers or conversions, respectively. Given the expense of laying

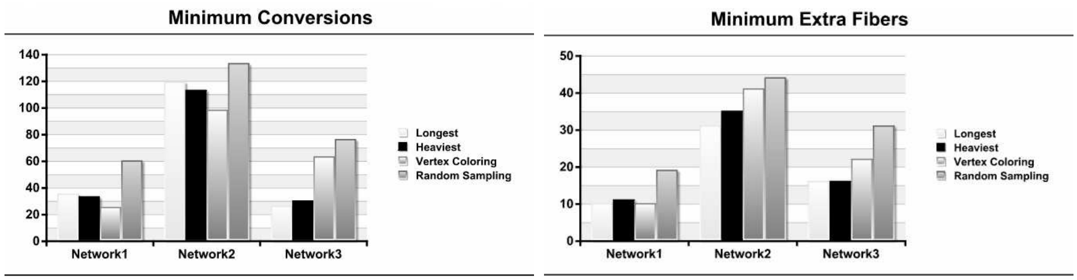

Figure 6: Networks in group 3.

additional fibers or adding conversion equipment, these problems are of great commercial interest to optical network service providers.

Given the inherent difficulty involved with guaranteeing good approximations to either **Min-Fiber** or **Min-Conversion** [1, 2], 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 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 is known.

Perhaps most interestingly, we introduce a new mixed **Min-Fiber/Conversion** cost reduction model in Section 3.3. 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 of potentially useful.

References

[1] M. Andrews and L. Zhang. Complexity of wavelength assignment in optical network optimization. *IEEE INFOCOM*, 2006.

[2] M. Andrews and L. Zhang. Minimizing maximum fiber requirement in optical networks. *Journal of Computer and Systems Sciences*, 72:118–131, 2006.

[3] Gilles Audemard and Laurent Simon. Gunsat: A greedy local search algorithm for unsatisfiability. In *IJCAI*, pages 2256–2261, 2007.

[4] Jörg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. 14:253–302, 2001.

[5] 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, pages 226–237, 2006.

[6] N. Lesh and M. Mitzenmacher. Bubblesearch: a simple heuristic for improving priority-based greedy algorithms. *Inf. Process. Lett.*, 97(4):161–169, 2006.

[7] Bart Selman, Hector Levesque, and David Mitchell. A new method for solving hard satisfiability problems. In *AAAI'92*, pages 440–446, 1992.

[8] Petr Slavík. A tight analysis of the greedy algorithm for set cover. In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 435–441, New York, NY, USA, 1996. ACM Press.

[9] Jonathan S. Turner. Almost all k-colorable graphs are easy to color. *J. Algorithms*, 9(1):63–82, 1988.