Estimating Network Throughput with an Adaptive Routing and Wavelength Assignment Algorithm

Robert J. Vincent, David J. Ives, and Seb J. Savory

Electrical Engineering Division, University of Cambridge. 9 JJ Thomson Avenue, Cambridge, CB3 0FA, UK.

rjv36@cam.ac.uk

Abstract: We propose an adaptive sequential loading algorithm that approaches the ILP throughput. Of 2,000 network realizations tested, shortest path routing achieves >90 % of the maximum throughput in 341 cases c.f. 1,904 with the proposed algorithm. © 2018 The Author(s)

OCIS codes: (060.0060) Fiber optics and optical communications, (060.4251) Networks, assignment and routing algorithms

1. Introduction

Routing and wavelength assignment is a well known optimization problem in optical communications [1]. An NP-hard problem, it is further complicated by the increased flexibility of transparent and elastic optical networks [2]. This complexity severely limits the size, and number, of networks that can be optimized with reasonable computing power, even with an efficient mixed integer linear program, MILP. Sequential loading offers a fast method to model network activity that, despite providing no guarantee of reaching optimal throughput, could replace traditional optimization heuristics for large networks where an approximate solution is better than none at all.

2. Methodology

Optimal network capacity was found using the MILP described in [3]. This models a transparent, fixed-grid network and assigns modulation formats according to signal-to-noise ratio, SNR, estimates for the worst-case, central WDM channel. We consider polarization multiplexed *m*-ary quadrature amplitude modulation (PM-*m*-QAM) at 25 GBd on a 32 GBd signal. This overhead was used for forward error correction capable of recovering signals with bit-error rates below 1.5 %. The highest-order modulation format considered was PM-1024-QAM which transmitted 20 bits per symbol but required an SNR of above 30.3 dB. PM-BPSK, at 2 bits per symbol, was the simplest format used and needed SNR of just 3.7 dB; routes with an SNR below this limit were deemed infeasible. Nonlinear noise from both self- and cross-phase modulation was modeled while four wave mixing was neglected. The fiber, $\alpha = 0.22$ dB/km, was assumed to be the only lossy component in the network and these losses were ideally compensated by Erbium doped fiber amplifiers with a noise factor 5, spaced 80 km apart. Uniform traffic was required between all node pairs and each demand had up to 15 route options, calculated using Yen's algorithm [4]. The resulting MILP was solved using IBM's CPLEX optimization suite.

The layouts of the NSF [5] and DTAG [6] networks, with 14 and 9 nodes respectively, were the basis of continental- and national-scale test networks. The nodes of each network are shown in Fig. 1.
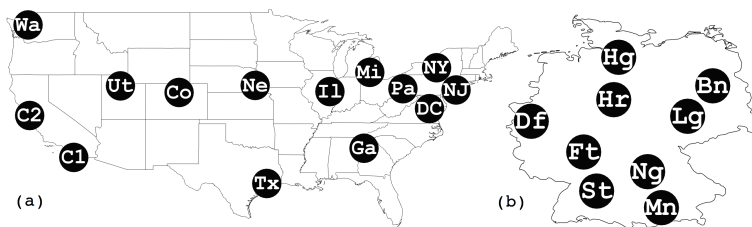

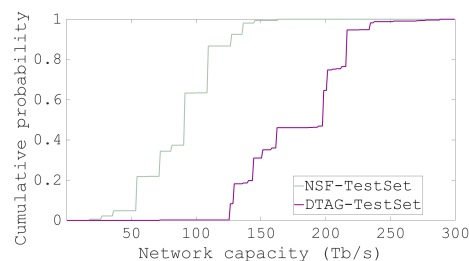Fig. 1: Node positions of (a) NSFnet and (b) DTAG

Fig. 2: Distributions of network capacity for NSF-TestSet (green) and DTAG-TestSet (purple).

We created the test networks by selecting the $N(N-1)/2$ possible links on each layout using a genetic algorithm, GA. To ensure these networks were geographically sensible, the GA sort to minimise the product of total fiber length and longest shortest path; node degree was restricted to between 2 and 5, inclusive. 1,000 networks for each layout were selected and were labeled NSF-TestSet and DTAG-TestSet. The mean node degree was 3.1, with standard deviation 0.2, and 2.6 (s.d. 0.2), respectively; the longest shortest path was 6,300 km (s.d. 400) and 1,100 km (s.d. 100). The

capacities were found using the MILP described above and the results are shown in Fig. 2. Uniform traffic constrains throughput on NSF-TestSet and DTAG-TestSet to multiples of 9.1 Tb/s and 3.6 Tb/s, respectively. Higher capacities were achieved on DTAG-TestSet due to its smaller size allowing higher-order modulation formats.

3. The algorithms

Simulation of uniform traffic for sequential loading typically uses a truly random traffic sequence where each demand is chosen with equal probability irrespective of previous demands. This rarely results in a uniform distribution for the final traffic configuration. Consequently, we suggest a pseudo-random approach which sees the $N(N-1)/2$ possible demands as cards in a pack, shuffles, deals, and repeats until a demand cannot be routed: this guarantees the traffic matrix is uniform each time the entire pack has been dealt and allows a direct comparison to the MILP solution.

We introduce two basic routing algorithms that take the K shortest paths, as described above, and search them for viable wavelengths in order of their cost. These costs are path length for K-Shortest Path routing, K-SP, and the number of links used by each path for K-Fewest Hops, K-FH. Both algorithms search wavelengths ordered from most to least congested and select the first viable route and wavelength pairing they encounter, terminating when no such pairing exists for a given demand. By repeatedly running these algorithms, with pseudo-random demand sequences, we can record the maximum uniform throughput for a given network however, as these repetitions are not connected, their number is an arbitrary choice. Each demand is PM-BPSK and if the route SNR allows for a higher-order format, subsequent demands are routed along the same route and wavelength until the capacity of the lightpath has been reached, at which point a new route and wavelength pair is searched for in the manner described.

A congestion adaptive algorithm, CA shown in Alg. 1, repeatedly calls 15-SP but, by changing the link weights based on prior congestion, the order in which these K routes are searched changes: a longer, but less congested, route may supersede a shorter, more congested route. We use this template to define CA-SP and CA-FH, which differ only in their inputs: the former defines G using link lengths, whilst the latter uses equal weights for all links, the first iteration for these algorithms are therefore 15-SP and 15-FH, respectively.

Algorithm 1: CA, congestion adaptive sequential loading based on shortest path routing

---
Inputs: Link Weights Matrix, G; Route Information, R; MaxReps
Outputs: Maximum Throughput, T_{max}; Optimal Weights Matrix, G_{maxT}

---
1. $G_{opt} = G$; $G_{new} = G$; Reps = 0
2.   **while** Reps $<$ MaxReps **do**
3. $[T$, LinkUtilization$]$ = **15-SP**(G_{new}, R); $G_{old} := G_{new}$
4.   **if** LinkUtilization $<$ 70 **then** LinkUtilization $:= 0$
5. $G_{opt} := G_{opt} + ($LinkUtilization$/80) * G$;
6. $G_{new} :=$ **normalize** G_{opt}; **update** R **from** G_{new}; $\delta := ||G_{old} - G_{new}||^2$
7.   **if** $\delta < \delta_{th}$ **do**
8. $G_{new} := G_{maxT}$; $R := R_{maxT}$
9.   MaxReps $:=$ Reps $+ 250$
10.   Reps$++$

Changes in the route search order are dictated by lines 4 and 5 of Alg. 1. We first define two weight matrices G_{opt} and G_{new}: the former grows in size with each iteration and is then normalized to make the latter. After initializing both with the original links weights G, we iteratively increase G_{opt} along links using more than 70 WDM channels, of the possible 80, by the fractional link usage multiplied by its original weight. This increases the cost of traversing links that were fully utilized slightly more than it does those that were merely mostly utilized. The scale of the increase is roughly constant but, if a certain link is consistently congested, the corresponding entry in G_{opt} grows with each iteration: this mimics simulated annealing, as the effective step size gets smaller with each iteration. After each iteration, we normalize G_{opt} to create G_{new}, the elements of which must sum to the number of links of the network and it is this that determines the search order for the next iteration. Normalization decreases the cost of traversing under-utilized links relative to the previous iteration in addition to providing a solution of constant magnitude which simplifies comparisons. We track the change in G_{new} using δ and reset the algorithm to the weights that gave the maximum throughput thus far once this change is smaller than a threshold value: it was found that 10^{-3} was a reasonable value for δ_{th}. After being reset, the algorithm was allowed 250 more iterations to maximally exploit this solution.

4.   Results and Discussion

K-SP and K-FH, using K values of 1 and 15, were repeated on each network 1,000 times; CA-SP and CA-FH were given up to 1,000 iterations per network. The maximum throughput achieved with each algorithm was recorded. Figs. 3a and 3b show the number of networks reaching optimal throughput for NSF-TestSet and DTAG-TestSet, respectively: we see very little difference between CA-SP and CA-FH but both see better performance on the larger NSF-TestSet networks (844 and 841) than on DTAG-TestSet (665 and 647). Figs. 3c and 3d show the number of networks that achieved better than 90 % of maximum throughput for each algorithm on NSF-TestSet and DTAG-TestSet, respectively. For DTAG-TestSet, this metric shows CA providing a reasonable estimate of network capacity in 98.1 % and 98.3 % of cases with CA-SP and CA-FH, respectively. For NSF-TestSet we see slightly worse performance but these metrics remain above 90 % for both CA strategies. Interestingly, although 1-SP and 1-FH perform identically, 15-FH outperforms its shortest path counterpart on both test sets.
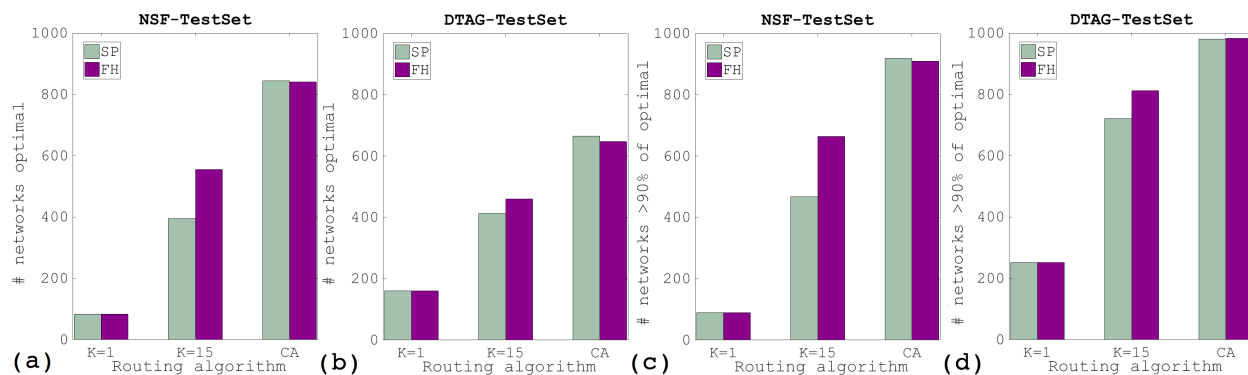

Fig. 3: (a) and (b) show the number of solutions that are optimal for various algorithms on NSF-TestSet and DTAG-TestSet, respectively. (c) and (d) show the number of networks achieve at least 90 % of optimal throughput NSF-TestSet and DTAG-TestSet, respectively.

On these small networks, the advantage of this method is minimal because of the high efficiency of traditional optimization heuristics: 481 of the 1,000 DTAG-TestSet solved faster using the MILP than with CA-SP, for instance. And on NSF-TestSet, the MILP was less than an order of magnitude slower in over 70 % of cases. However the variance in solve time was very much higher: the slowest network took over 7 hours compared to just 72 seconds to run *both* CA-SP and CA-FH for that same network. The sub-optimal solution, in this case 93 % of capacity, now seems a reasonable trade-off, especially if the number of nodes was increased, say greater than 20, when each MILP would take, on average, several hours to solve. The increased predictability of solve times allows many networks to be simulated without worrying about incredibly slow outliers and this enables, for instance, a heuristic to find the optimal network topology on an arbitrary set of nodes. While the current algorithm, as presented, assumes uniform traffic preliminary investigations have found that it can be extended to non-uniform traffic when required and likewise it can be extended towards flex-grid networking.

5.   Conclusion

We have presented a new sequential loading algorithm for fast routing and wavelength assignment that learns from previous iterations to delay the onset of congestion and hence increase network throughput. This method was able to achieve at least 90 % of maximum throughput in over 95 % of cases, thereby providing a fast estimate of network throughput.

References

1. J. M. Simmons, *Optical Network Design and Planning*, (Springer, 2014), 2nd ed.
2. O. Gerstel *et. al.*, "Elastic optical networking: a new dawn for the optical layer?," IEEE Commun. Mag. **50**, s12–s20 (2012).
3. D. J. Ives *et. al.*, "Routing, modulation, spectrum and launch power assignment to maximize the traffic throughput of a nonlinear optical mesh network," Photonic. Netw. Commun. **29** 3 244–256 (2015).
4. J. Y. Yen, "Finding the K Shortest Loopless Paths in a Network," Manag. Sci. **17** 712–716 (1971).
5. R. Ramaswami and K. N. Sivarajan, "Design of logical topologies for wavelength-routed all-optical networks," IEEE J. Sel. Areas Commun. **14** 840–851 (1996).
6. A. Betker *et. al.*, "Comprehensive Topology and Traffic Model of a Nationwide Telecommunications Network," J. Opt. Commun. Netw. **6** 1038–1047 (2014).