A Machine Learning Framework for Scalable Routing and Wavelength Assignment in Large Optical Networks

(Article begins on next page)

A Machine Learning Framework for Scalable Routing and Wavelength Assignment in Large Optical Networks

Davide Andreoletti[1], Cristina Rottondi[2], Andrea Bianco[2] and Silvia Giordano[1]

[1]*Networking Laboratory, University of Applied Sciences of Southern Switzerland, Viganello, Switzerland*
[2] *Politecnico di Torino, Italy,* [3] *Politecnico di Milano, Italy*

{*davide.andreoletti,silvia.giordano*}*@supsi.ch,*{*cristina.rottondi,andrea.bianco*}*@polito.it*

Abstract: We perform a machine-learning-based network pruning that significantly reduces the complexity of routing and wavelength assignment in large optical networks. A significant computational time reduction is achieved by accepting a minor deterioration of the obtained solution.

1. Introduction

Today's Internet is a complex ecosystem composed of a large number of interconnected Autonomous Systems (ASs) that transport, mainly over optical networks, massive amounts of traffic generated by bandwidth-hungry and latency-sensitive services. In this scenario, effective network management strategies must be implemented to meet the increasingly-stringent performance requirements of current and upcoming services. This management might require, for instance, the quick resolution of common network planning problems. Unfortunately, exact optimization methods like the mixed integer linear programming suffer from severe scalability issues, that prevent their application to large real-world networks, especially when timely decision-making is required. Such issues are commonly tackled by means of heuristic approaches, which are capable to provide sub-optimal solutions in shorter time [2]. In recent years, Machine Learning (ML) has been used to further reduce these scalability limitations, either by directly replacing exact optimization methods, or by enhancing existing heuristic approaches.

In this work, we consider the Routing and Wavelength Assignment (RWA) problem in large optical networks. This problem was proven to be NP-hard, and ML-based approaches have been proposed to reduce its complexity. In [4] a model that maps a given traffic demand to a sequence of links along a specific wavelength is learned by means of supervised classification. In [5], a deep neural network is employed to select the best RWA strategy for a given traffic demand, under preliminary calculation of a predefined number of shortest paths. Reinforcement learning frameworks for routing, modulation format and spectrum assignment have also been devised for network scenarios with dynamic traffic [6, 7]. Despite these emerging ML-based solutions, the most common approach to tackle the RWA problem is a simple heuristic that first computes the shortest path over which the lightpath is routed and then assigns an optical channel. This strategy makes it possible to solve the RWA in a reasonable amount of time, but still suffers from scalability issues that prevent its real-time application on very large networks. To handle this problem, we develop a ML-based pruning system that, given an input graph and a source/destination pair, returns a sub-graph formed only by the nodes that most likely belong to the shortest path. In this way, the size of the original graph is dramatically reduced, and the computational time of the RWA task significantly lowered (of up to 35%), as observed from extensive simulations performed over a network with tens of thousands of nodes and links. This improvement is achieved with an acceptable increment in the blocking probability, and a negligible increase of the average length of the computed shortest paths.

2. Machine Learning Framework for Scalable Routing in Large Optical Networks

First, let us remark that the complexity of the RWA for a given lightpath can be significantly reduced if this task is executed on a sub-portion of the original network that contains the shortest path between source and destination of the considered lightpath. Starting from this observation, we propose a ML algorithm that *i*) estimates the likelihood that each node of the network belongs to the shortest path between a source and a destination and *ii*) prunes the network accordingly, i.e., by selecting the nodes (and the links interconnecting such nodes) whose likelihood is higher than a pruning threshold. Then, the RWA task is executed on the obtained reduced network, as depicted in Fig.1. As common in the literature, we also assume that the routing phase is performed using the Dijkstra algorithm, while the wavelength assignment phase is executed with either the First-Fit (FF) or the Random Wavelength Selection (RS) strategy. Given a network with $|V|$ nodes, $|E|$ links with $|W|$ optical channels

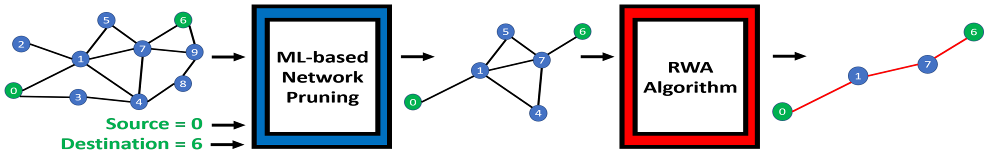


Fig. 1: Representation of the overall ML-aided RWA strategy

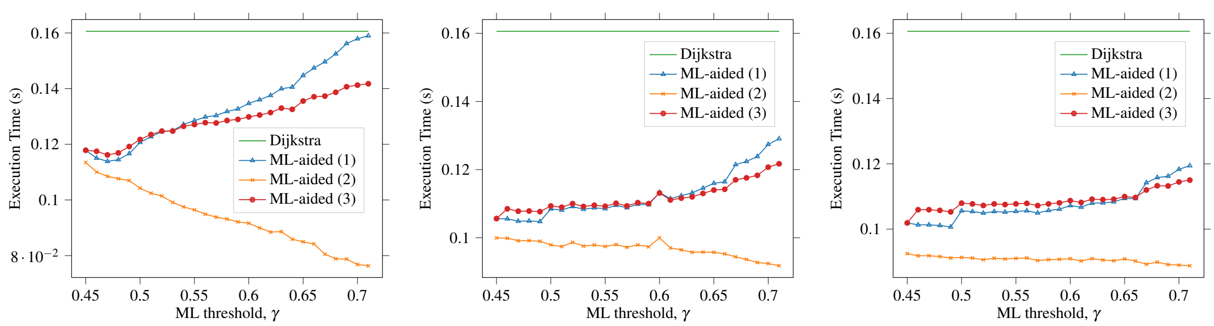
and diameter $|D| << |E|$, the complexity of the Dijkstra algorithm is $O(|V|\log|V| + |E|)$, while the complexity of both FF and RS is $O(|WD|)$. In real-world large networks, the number of optical channels is below 100, but the number of nodes and links might easily reach tens of thousands. Therefore, an effective network pruning system has the potential to drastically reduce the complexity of the RWA problem.

More in detail, the proposed network pruning system is based on a ML inference system that computes the likelihood p^i_{sd} that the generic node n_i belongs to the shortest path between a source node n_s and a destination node n_d. To train and then query this inference system, a suitable nodes representation strategy is required. We argue that the nodes belonging to a shortest path likely share some topological properties (e.g., similar positions with respect to a particular node), and we propose a representation that aims to capture this property. We randomly select K *anchor nodes*, and we represent the generic node n_i as a vector $\mathbf{v_i}$ whose j-th component is the length of the shortest path between n_i and the j-th anchor, with $j \in \{1,\ldots,K\}$. The training process is then performed as a regression to learn how to map the set of representations of nodes n_i, n_s and n_d onto a real value p^i_{sd} within the range $[0,1]$, which indicates the probability that n_i belongs to the shortest path between n_s and n_d. More specifically, the ML algorithm is fed with a vector of size $3K$ obtained by juxtaposing $\mathbf{v_i}$, $\mathbf{v_s}$ and $\mathbf{v_d}$ (i.e., the vectors representing n_i, n_s and n_d, respectively). The probabilities are then "binarized" as a function of a pruning threshold parameter γ: if $p^i_{sd} \leq \gamma$, n_i is pruned, otherwise n_i is kept in the reduced network. The considered ML algorithm is a Decision Tree trained using a dataset formed by shortest paths computed between randomly-selected pair of nodes. The size of the training set is equal to the 0.4% of the total number of shortest paths between any pair of nodes in considered network (which are $|V| \cdot (|V|-1)/2$). Note that our ML framework is developed having scalability as a primary design objective. For this reason, we have chosen a nodes representation strategy and an inference ML algorithm that only involve low-complexity operations. After the training phase, the ML-based inference system can be used to implement the network pruning system. As the ML prediction is prone to errors, nodes belonging to the shortest path might be mistakenly pruned, and this can eventually lead to sub-optimal solutions of the RWA, or even to the impossibility of finding a path between source and destination. In this latter case, we envision a re-execution of the RWA task either on the original graph, or on a graph pruned considering a lower node selection threshold. This re-execution introduces a time overhead, as further discussed in Sec.3.

3. Results

We assess the performance of the RWA task executed with the proposed ML-aided network pruning system by comparison to those obtained without the pruning phase, considering a network of $|V| = 50166$ nodes and 103252 links with $|W| = 80$ optical channels per link and formed by interconnecting 1000 ASs. Both the intra-ASs and inter-ASs networks are generated according to the model described in Ref. [1].

Fig. 2 shows the average time required to compute 3000 shortest paths for $\gamma \in [0.45, 0.7]$. Specifically, we compare the computational times of the "Dijkstra" algorithm executed on the original graph with the "ML-aided" on the pruned graph under three scenarios: (1) if the path is not found, it is re-computed executing Di-



(a) k = *3 anchor nodes* (b) k = *5 anchor nodes* (c) k = *7 anchor nodes*

Fig. 2: Execution time of the routing algorithms for several values of the pruning threshold γ when shortest paths are computed on the original graph (Dijkstra), the pruned graph (ML-aided (1)), the pruned graph only when a path is found (ML-aided (2)) and the pruned graph with optimized threshold selection (ML-aided(3))

jkstra on the original graph; (2) only when a path is found and (3) if the path is not found, the shortest path is re-computed over a graph pruned setting a lower node selection threshold γ and, if also this re-computation fails, Dijkstra is executed on the original graph. As for scheme (3), at the second iteration γ is chosen by means of an optimization strategy that we omit due to space limitations. Fig.2(a), 2(b) and 2(c) report the execution time considering $K = 3,5,7$ anchors nodes, respectively. Results show that network pruning leads to a significant gain in terms of computational time, and this gain increases with increasing K (e.g., from 29% to 35% when K goes from 3 to 7). Augmenting K also increases the robustness with respect to a variation of γ. Furthermore, increasing γ reduces the computational time on the pruned graph when the path is found but, conversely, this time increases if path re-computation is also considered (where a re-computation done through a second optimized threshold selection outperforms the selection directly on the original graph).

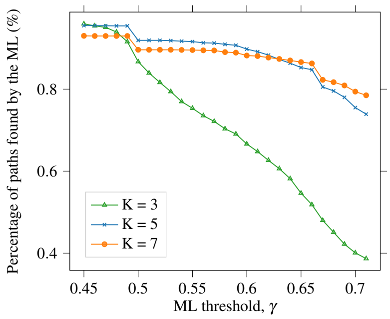


Fig. 3: Percentage of paths found on the pruned graph for several values of the node selection threshold

From now on, we focus on ML-aided (1) option. We measure the percentage of extra-length of the paths found on the pruned network with respect to the length of the shortest paths and the percentage of paths directly found on the pruned network. As for the former, we observe a negligible extra-length that ranges from a minimum of 0.7% (with $K = 3$ and $\gamma = 0.7$) to a maximum of 1.6% (with $K = 7$ and $\gamma = 0.6$). As for the latter, Fig.3 shows that the percentage of times a path is found directly on the pruned network decreases with increasing γ (as the graph might be more easily disconnected after network pruning). A higher robustness to the variation of γ is achieved by setting K to higher values. For instance, this percentage experiences a significant drop - from 96% to 39% - when $K = 3$, but only from 96% to 74% when $K = 5$.

We then evaluate the performance of the entire RWA process when the ML-aided (1) pruning system is employed. To this end, we simulate a dynamic traffic scenario where the arrival times of 10^4 requests are characterized by a Poisson distribution and their duration by a negative exponential distribution. Requests are routed on the shortest path between their sources and destinations using the Dijkstra algorithm and an optical channel is assigned using either the FF or the RS strategy. If no available spectrum slot is found, the request is blocked. This experiment is repeated in three different load scenarios characterized by blocking probabilities in the order of 10^{-4}, 10^{-3} and 10^{-2}. Each experiment is independently repeated 20 times and the average blocking probabilities and 95% confidence intervals are shown in Tab.1, where OG and PG stand for original and pruned graph, respectively. We observe a minimal increase of the blocking probability when the RWA task is executed on the graph pruned considering $\gamma = 0.5$. Finally, note also that the time required to pre-compute the shortest path between each node pair in the considered network takes around 100 minutes, while the ML algorithm can be trained in a much shorter time, which grows linearly with the training set. As an example, 6 seconds are sufficient to train a model achieving a prediction accuracy of 93% on unseen nodes, while a training of 60 seconds increases the accuracy to about 96% (note that prediction accuracy is the fraction of nodes correctly labelled as belonging/not belonging to the shortest path between a node pair). Moreover, though the path pre-computation strategy exploiting Dijkstra can be applied to static topologies, it is not suitable for topologies that change over time. Dynamic topologies will be considered in a future work.

	$\cdot 10^{-4}$	$\cdot 10^{-3}$	$\cdot 10^{-2}$
OG/FF	6.71 ± 0.95	5.87 ± 0.19	1.700 ± 0.024
OG/RS	8.15 ± 0.81	5.92 ± 0.19	1.710 ± 0.023
PG/FF	9.90 ± 0.98	6.80 ± 0.21	1.900 ± 0.024
PG/RS	11.70 ± 1.01	6.90 ± 0.20	1.900 ± 0.023

Table 1: Average Blocking Probability

In summary, the proposed ML-based network pruning strategy significantly reduces the computational complexity of the RWA task with a minor blocking probability deterioration.

References

1. A. Elmokashfi et al., "On the scalability of BGP: The role of topology growth. IEEE Journal on Selected Areas in Communications, vol. 28, no. 8, pp. 1250-1261, 2010.
2. B. Jaumard et al., "Efficient spectrum utilization in large scale RWA problems. IEEE/ACM TON, vol. 25, no. 2, pp. 1263-1278, 2017.
3. C. L. Gutterman et al., "Neural network based wavelength assignment in optical switching", Workshop on Big Data Analytics and Machine Learning for Data Communication Networks, 2017.
4. I. Martín et al., "Machine learning-based routing and wavelength assignment in software-defined optical networks," IEEE TNSM, vol. 16, no. 3, pp. 871-883,
5. J. Yu et al., "A deep learning based RSA strategy for elastic optical networks" IEEE ICOCN, 2019.
6. X. Chen et al., "DeepRMSA:A deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks," *IEEE/OSA JLT*, vol. 37, no. 16, pp. 4155–4163, 2019.
7. X. Luo et al., "Leveraging double-agent-based deep reinforcement learning to global optimization of elastic optical networks with enhanced survivability," Optics express, vol. 27, no. 6, pp. 7896–7911, 2019.