Evaluation of Topology Optimization Objectives

Y. Sinan Hanay*, Shin'ichi Arakawa†* and Masayuki Murata†*
* Center for Information and Neural Networks (CiNet), NICT, Japan
Email: hanay@nict.go.jp
† Graduate School of Information Science and Technology, Osaka University, Japan
Email: {arakawa, murata}@ist.osaka-u.ac.jp

Abstract—**Two network-wide optimization contexts are traffic engineering and topology optimization. Various optimization objective functions and metrics have been proposed for both contexts. Yet, it is hard to evaluate the efficiency of those optimization objectives. Previously, a study analyzed the efficiency of some optimization metrics for traffic engineering by using linear programming (LP). On the other hand, in the topology optimization domain, there has not been any work on evaluation of different metrics. Because, it is hard to evaluate these metrics as the optimization algorithms are objective function tailored heuristics generally. As a result, a fair comparison of different objectives becomes hard. In this work, using machine learning we compare and analyze different traffic optimization objectives for topology optimization.**

I. Introduction

Traffic engineering (TE) and topology optimization are two domains in network optimization. TE focuses on routing optimization and load balancing. On the other hand, topology optimization focuses on which routers to connect. Both approaches try to optimize a performance goal such as minimizing maximum link utilization, average delay, weighted hop count, average queuing delay or maximizing available bandwidth. These are some well known network-wide optimization objectives.

In TE domain, most of the research have focused on optimizing the link weights to achieve an optimization objective [1], [2]. On the other hand, little has been done on the evaluation of how well optimization objectives do, such as the work in [3]. In that pioneering work, the researchers investigated the efficiency of different optimization objectives. They took linear programming approach for evaluating of different objectives while making some linear approximations on non-linear optimization objectives. Rightfully, they acknowledge the shortcoming of linear approximations. As a result, there remains a need for a fair comparison of different optimization objectives.

Optical communications is capable of carrying many channels simultaneously using wavelength-division multiplexing (WDM). This capability of optical medium allow establishment of many different *virtual topologies* on top of the very same physical topology. Selecting an efficient virtual topology (VT) is an important problem in autonomous systems (AS), such as metropolitan area networks (MAN).

In this work, we compare several topology optimization objectives, which has not been done before. However, the main contribution of our work is to provide a fair comparison

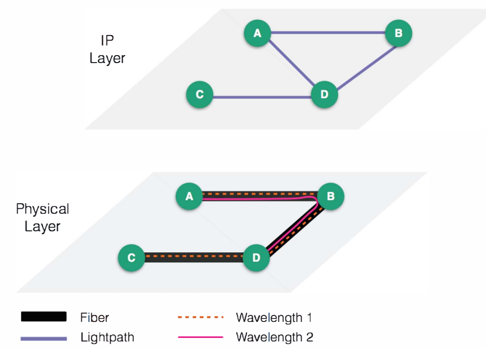

Fig. 1. Illustration of topology optimization problem. A physical topology with 4 routing nodes (A, B, C and D) and the corresponding virtual topology at the IP layer is shown.

for different optimization objectives. We strive to provide fair comparison by using a machine learning algorithm. Previously, non-linear objective functions were evaluated using linear approximations [3]. On the other hand, with machine learning such unfairness in the evaluation can be avoided. More importantly, we evaluate those objectives under realistic, dynamic traffic. This allows us to make our conclusions more comprehensive.

II. Motivation

There has been only a single work which considered different TE optimization objectives and the authors concluded that some objective functions are worthy than others [3]. However, as the authors acknowledge the use of linear approximation for nonlinear objective functions limits the scope of conclusion, and an alternative approach is crucial to understand performance of different optimization objectives.

To illustrate why the selection of objective function matters, let us look at a very common objective in topology optimization. One rule of thumb is to keep maximum link utilization under 50 percent or minimize it. However, minimizing maximum link utilization is overly sensitive to bottleneck links [3]. In other words, maximum link utilization is a very local metric, which may be far from capturing the global network performance. This is a common problem with objective functions based on min-max formulations. Yet, this has been the most common optimization metric.

Topology optimization takes place at the physical layer at the core of the Internet. Figure 1 illustrates the topology op-

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 27,2024 at 00:16:05 UTC from IEEE Xplore. Restrictions apply.

timization problem. In an all-optical, IP-over-WDM network, each router is equipped with a set of transmitters and receivers. Each fiber link can carry a certain number of wavelengths. Optical cross-connects serve as a switching device for optical signals, and associates and incoming link with an outgoing link. This allows the possibility of establishing various “virtual topologies” on top of a physical topology. Topology optimization problem is more specifically referred as virtual topology design (VTD) among the community.

The problem is called virtual topology reconfiguration (VTR), when the virtual topology is updated periodically. In this work, we focus on VTR problem which is illustrated in Figure 1. In the illustration, each fiber link can carry two wavelengths. Wavelength 1 was used to connect 1-hop nodes (i.e. A-B, B-D and C-D), and wavelength 2 was used to connect pair A-D. Thus, now A and D are connected, and in IP layer this connection seem as seamless, the edge connecting nodes in IP layer called “lightpaths”.

III. Preliminaries

The machine learning algorithm we use in this work is called Attractor Selection Based (ASB) topology control. In this section, we briefly review ASB. The details of ASB can be found in the prior work [4], [5].

ASB is built on neural networks. The learning type it utilizes can be regarded as reinforcement learning in a broad sense. ASB explores topologies randomly, it remembers good topologies by storing them in a memory. During an exploration, once a good performing topology has been found, it is stored in a list of “good topologies”. At any point in time, those good topologies attract the algorithm to converge a topology similar to themselves. Thus, we refer these good topologies as “attractors”. By similar, we mean two topologies having small Hamming distance.

A. ASB Algorithm

ASB algorithm utilizes neural memories to store found good topologies. The type of neural memory we use is auto-associative memories. Auto-associative memories can be used to correct noisy inputs by trying associate a given input to one of the stored patterns (e.g. topologies). ASB aims to find an optimal virtual topology (VT), and it changes topologies using the following equation [4]:

$$\frac{dx_i}{dt} = \underbrace{\left[f\left(\sum_{j=1}^{n} w_{ij} x_j \right) - x_i \right]}_{\text{auto-associative memory}} \alpha + \underbrace{\mathcal{N}(0,1)}_{\text{random walk}} \qquad (1)$$

where $\mathcal{N}(0,1)$ is the standard normal random variable, f can be sign or sigmoid function. The value α is the optimization metric to be maximized. For example, if we want to minimize u_{max}, then α should be inversely proportional to u_{max} (i.e. $\alpha \propto \frac{1}{u_{max}}$). For example, if $u_{max} = 0.1$, this means its a very good state since u_{max} is very low. Here, x_i represents the likelihood of establishing a path for node pair i. In each round, ASB makes changes to the present topology based on x_i values. For example, if x_i is greater than 0.5, then lightpath for pair i is established; otherwise the lightpath is terminated (if it exists). Of course, the lightpath is established only if corresponding resources are available (i.e. wavelength and ports).

IV. Related Work

A previous work compares the various traffic engineering objective functions [3]. The authors consider linear and non-linear objective functions, and find the optimal solution using linear programming. In nonlinear objective functions, such as mean delay, they make linear approximations.

On topology optimization, some researchers used mixed-integer linear programming (MILP). However, there are some drawbacks of using linear programming. The problem becomes intractable for networks that have more than 10 nodes[6]. Even for topologies of 23 nodes, running time can be as long as 9 hours [7]. In addition to performance issues with MILP methods, a few metrics we propose here cannot be formulated as a linear programming problem since they are nonlinear, such as mean delay. Balon and his colleagues addressed this problem by using linear approximation [3]. They also highlighted the drawback of this linear approximation.

V. Comparison of Optimization Metrics

In this section, first we justify the use of machine learning. Then we look at the commonly used optimization metrics, and finally we present our evaluation methodology.

A. Why Machine Learning?

In this work, we use machine learning to evaluate the different optimization metrics for two reasons. Most of the VTR optimization methods use heuristics. For each optimization, it is necessary to use a different heuristic. Using different heuristics, prevents comparison of different objectives on fair grounds.

Before we are able to use machine learning, we need to show that this optimization problem is feasible to solve with machine learning. Machine learning works well when there is a trend or pattern in the data or variables that can be captured statistically. There is no consensus on self similarity of Internet traffic in the the research community. However, long range dependency (LRD) was accepted and observed in real traffic settings [8]–[10].

1) Traffic Analysis: First, we analyze real traffic trace from GEANT topology, which consists of 23 nodes, provided by TOTEM project [11]. To understand the correlation in a finer detail, a measure called Hurst exponent is used. Figure 2 shows the Hurst exponents of GEANT traffic taken from GEANT topology between January 1st to January 11th 2006. . A Hurst exponent close to 0.5 indicates an uncorrelated series, and a Hurst exponent between 0.5 and 1 means long-term positive autocorrelation. Higher Hurst exponent values means stronger correlation. Note that, GEANT traffic shows slightly stronger correlations than our synthetic traffic.

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 27,2024 at 00:16:05 UTC from IEEE Xplore. Restrictions apply.

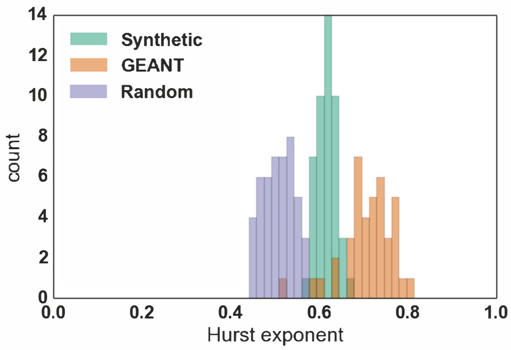

Fig. 2. The histogram shows the Hurst parameters for the real traffic trace from GEANT topology, and our synthetic traffic. An uncorrelated series has a Hurst exponent of 0.5.

B. Optimization Objectives

In VTR, three commonly objectives are minimizing maximum link utilization, minimizing average weighted number of hops and minimizing average end-to-end delay [6]. Though many optimization objectives has been proposed, we limit our discussion to the most commonly used, and non-parametric ones.

1) Maximum Utilization (maxUtil): Link utilization of a link can be described by $u_i = \frac{l_i}{c_i}$, where l_i is the load of the link i, and c_i is the capacity of the link. Then link utilization of the most heavily loaded link is denoted as u_{max}.

2) Blanchy: The sensitivity of u_{max} can be solved by considering all link utilizations. Blanchy et. al. proposed a metric that tries to reduce variance of link utilizations by using

$$\text{Blanchy} = \sum_{i \in E} (u_i - u_{mean})^2 \tag{2}$$

Here, u_{mean} is average utilization of all links. The aim here is to balance the load across all links (i.e. E).

3) Average Weighted Number of Hops (weightedHop): Along with $maxUtil$, average weighted number of hops is most common metric in topology optimization. We simply refer it as *weightedHop.* weightedHop is the average number of paths traversed by one unit traffic [6]. It is a traffic weighted hop count, rather than the pure hop count. Balon and his colleagues used minimum hop count in their work [3].

4) Delay: It has been discussed that a natural choice link cost is delay [12], and it can be calculated by:

$$\text{Delay} = \sum_{i \in E} \frac{1}{c_i - l_i} \tag{3}$$

5) Normalized Available Bandwidth ($NABW$): In traffic engineering context, a method called minimum interference routing algorithm (MIRA) has been introduced previously [2]. Authors propose an objective function to maximize available bandwidth on all possible pairs. In MIRA, basic motivation is to maximize future traffic demands. It is not possible to apply directly MIRA in topology optimization context due to inherent differences of topology optimization with TE. However, we propose a new algorithm called normalized available bandwidth (NABW), which tries to achieve similar maximum future demands.

Let's assume that $l_{max}(i, j)$ is the maximum utilization on the path $i - j$, we define an objective function for each path $i - j$ as

$$NABW(path_{i,j}) = \frac{1 - l_{max}(i, j)}{\text{\# of paths passing through } l_{max}(i, j)} \tag{4}$$

then, we sum for all paths as

$$total(NABW) = \sum_{\forall i,j \in N} NABW(path_{i,j}) \tag{5}$$

The intuition is, if there are more paths passing through a link, then that link has to have more importance than another link having same amount of residual bandwidth. Our goal is to maximize $total(NABW)$.

C. Modifying ASB

Due to the inherent nature of ASB, we must make a few modifications to provide fairness for different objectives. In ASB, α is calculated by

$$\alpha = \frac{1}{1 + e^{50(\mu_{max} - 0.5)}} \tag{6}$$

For other metrics, we need to have similar mapping to [0,1] range. However, for metrics like weightedHop, this mapping is not straightforward. Unlike u_{max}, it is not possible to know what can be a good weightedHop for a given traffic demand and topology. Theoretically the lower bound for weightedHop can be 1, but it is hard to find an upper bound and come up with a mapping function from weightedHop to α. This is also true for $Delay$, $NABW$ and $Blanchy$.

The approach we take in this work is as following. In the training phase, the first 200 rounds, we record the minimum and maximum seen values such as $NABW_{min}$ and $NABW_{max}$. Then, after the training period, we can calculate α as

$$\alpha = \frac{NABW - NABW_{min}}{NABW_{max} - NABW_{min}} \tag{7}$$

VI. Simulation Results

In this section, we first present the simulation settings, then present our results.

The simulations has been randomized using different seeds, number of ports and number of attractors. Dijsktra's shortest path algorithm was used for both traffic and lightpath routing.

Table I presents the results of simulations for three well established performance metrics. The best performances are shown in bold, while worst performances in italic. $Delay$ performs best on all three metrics. $weightedHop$ performs very close to $Delay$ in total available bandwidth (ABW).

Figure 3 shows the algorithms performance under various traffic loads. As the figure shows, $Delay$ performs best for minimizing maximum link utilization, and $MaxUtil$ comes closer under very heavy traffic loads. This result also shows

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 27,2024 at 00:16:05 UTC from IEEE Xplore. Restrictions apply.

TABLE I
COMPARISON OF OBJECTIVES.

objective	μ_{max}	total ABW	weighted hop
maxUtil	0.53	*0.50*	2.07
NABW	0.53	0.54	2.05
Delay	**0.33**	**0.71**	**1.94**
Blanchy	*0.56*	0.54	*2.09*
weightedHop	0.49	0.68	2.01

how our work differs from previous work [3]. Since we use dynamic traffic, and try to optimize based on traffic matrix of previous round. As a result, we do not necesarily expect the objective function to achieve the best result in its related metric, such as $MaxUtil$ achieving the best u_{max}, or $weightedHop$ to best weighted hop.

Next, we look at the total ABW as the traffic load increases. Figure 4 shows $Delay$ outperforms all other objectives, however at high loads again $maxUtil$ comes closer to $Delay$. Overall, we observed that $maxUtil$ performs poorly. Conversely, $Delay$ performs best for all three metrics we considered.

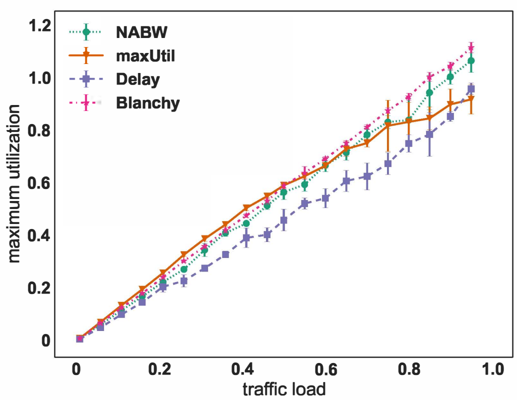

Fig. 3. Maximum utilization for different optimization objectives as the traffic load is varied. The bars correspond to 95% confidence intervals. Note that NABW and Blanchy results in overloaded links (i.e. utilization higher than 1) for high traffic loads.

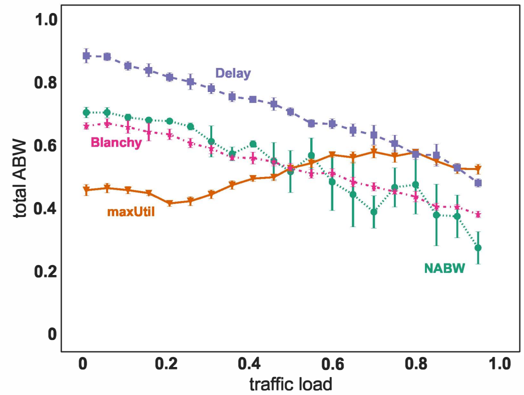

Fig. 4. Total ABW as the traffic load increases. $Delay$ achieves much higher ABW than all the other metrics. $weightedHop$ is not shown, as it performs very similar to $Delay$.

VII. CONCLUSION

Previously, only in one work, using linear programming some researchers evaluated the efficiency of such objectives. However, use of linear approximation for nonlinear objective functions can be problematic. In addition, even though it was suggested that $Delay$ is a better optimization objective than $maxUtil$, the research community has been sticking with $maxUtil$.

We compared different topology optimization metrics using machine learning. Comparison of optimization objectives and use of machine learning are two novel aspects of this study. Use of machine learning is especially crucial, as it strives to provide a fair framework for all objective functions.We found out that $Delay$ is the best metric, which is in agreement with conclusions of Balon and his colleagues. However, our conclusion is more comprehensive. Instead of static traffic, we used a dynamic traffic and analyzed the 33 days of traffic. In the end, we showed the predictive capability of different objective functions. Even though we took a different approach, our conclusion agrees with the previous work.

REFERENCES

[1] B. Fortz and M. Thorup, "Optimizing ospf/is-is weights in a changing world," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 4, pp. 756–767, May 2002.
[2] K. Kar, M. Kodialam, and T. Lakshman, "Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 12, pp. 2566–2579, 2000.
[3] S. Balon, F. Skivée, and G. Leduc, "How well do traffic engineering objective functions meet TE requirements?" in *Proceedings of the 5th IFIP Networking*, May 2006.
[4] Y. Koizumi, T. Miyamura, S. Arakawa, E. Oki, K. Shiomoto, and M. Murata, "Adaptive virtual network topology control based on attractor selection," *J. Lightwave Technol.*, vol. 28, no. 11, pp. 1720–1731, Jun 2010.
[5] Y. S. Hanay, S. Arakawa, and M. Murata, "Network topology selection with multistate neural memories," *Expert Systems with Applications*, vol. 42, no. 6, pp. 3219–3226, 2015.
[6] J. Zheng and H. T. Mouftah, *Optical WDM Networks*. Wiley-IEEE Press, 2004.
[7] R. Aparicio-Pardo, N. Skorin-Kapov, P. Pavon-Marino, and B. Garcia-Manrubia, "(non-)reconfigurable virtual topology design under multi-hour traffic in optical networks," *Networking, IEEE/ACM Transactions on*, vol. 20, no. 5, pp. 1567–1580, Oct 2012.
[8] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido, "A nonstationary poisson view of Internet traffic," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 3. IEEE, 2004, pp. 1558–1569.
[9] P. Borgnat, G. Dewaele, K. Fukuda, P. Abry, and K. Cho, "Seven years and one day: Sketching the evolution of internet traffic," in *INFOCOM 2009, IEEE*, Apr. 2009, pp. 711–719.
[10] W.-B. Gong, Y. Liu, V. Misra, and D. Towsley, "Self-similarity and long range dependence on the internet: a second look at the evidence, origins and implications," *Computer Networks*, vol. 48, no. 3, pp. 377–399, 2005.
[11] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 83–86, 2006.
[12] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "Mate: MPLS adaptive traffic engineering," in *INFOCOM '01*, vol. 3. IEEE, pp. 1300–1309.

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 27,2024 at 00:16:05 UTC from IEEE Xplore. Restrictions apply.