Network Resilience Assessment via QoS Degradation Metrics: An Algorithmic Approach

Lan N. Nguyen and My T. Thai*

Department of Computer & Information Science & Engineering

University of Florida, US

lan.nguyen@ufl.edu, mythai@cise.ufl.edu

ABSTRACT

This paper focuses on network resilience to perturbation of edge weight. Other than connectivity, many network applications nowadays rely upon some measure of network distance between a pair of connected nodes. In these systems, a metric related to network functionality is associated to each edge. A pair of nodes only being functional if the weighted, shortest-path distance between the pair is below a given threshold T . Consequently, a natural question is on which degree the change of edge weights can damage the network functionality? With this motivation, we study a new problem, *Quality of Service Degradation*: given a set of pairs, find a minimum budget to increase the edge weights which ensures the distance between each pair exceeds T . We introduce four algorithms with theoretical performance guarantees for this problem. Each of them has its own strength in trade-off between effectiveness and running time, which are illustrated both in theory and comprehensive experimental evaluation.

ACM Reference Format:

Lan N. Nguyen and My T. Thai. 2019. Network Resilience Assessment via QoS Degradation Metrics: An Algorithmic Approach. In *ACM SIGMETRICS Int'l Conference on Measurement & Modeling of Computer Systems (SIGMETRICS'19 Abstracts)*, June 24–28, 2019, Phoenix, AZ, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3309697.3331470>

1 INTRODUCTION

Graph connectivity has been considered as an important metric on measuring the functionality of a network. However, many network applications now consider other factors when determining a network functionality in addition to connectivity. For example, consider a time-sensitive delivery on a road network, where edge weights represents the travel time between destinations. Connectivity between a source and a destination is insufficient when a guarantee on the delivery time is required [3].

Therefore, a natural question is whether a tech-savvy attacker can damage the network functionality without impacting the connectivity? Under various forms, this kind of attacks actually is common, yet stealthy (e.g. Congestion attack on road networks [1]). With this motivation, we consider the *Quality of Service Degradation* (QoSD) problem. Given a directed graph G representing a network, a threshold T and a set of pairs S in G , the objective is to identify a minimum budget to increase the edge weights (or node weights for node version) to ensure the weighted, shortest-path

distance between each pair in S is no smaller than T . Intuitively, the goal of this problem is to assess how robust the network is; the greater budget to increase edge weights found, the more resilient the network is to the perturbation in terms of edge weights. In addition, the budget to increase weight of an edge in the solution provides an indication of the importance of this edge to the desired functionality.

In the context of network reliability, Kuhnle et al. [3] have recently studied a special case of our problem under the name LB-MULTICUT. The objective of this problem is to identify a minimum set of edges whose **removal** ensures the distance between each pair of nodes is no smaller than T . Directly adopting the LB-MULTICUT solutions to our QoSD problem is not feasible since most of those solutions exploited a trait that their problems can be formulated by Integer Programming and exhibit submodular behaviors. QoS problem, on the other hand, is shown to be neither submodular nor supermodular, making QoS more challenging to devise an efficient algorithm. Also, modern networked systems are increasingly massive in scale, often with size of millions of vertices and edges. The need for a scalable algorithm on large-scale networks poses another challenge for our problem. Motivated by these observations, the main contribution of this work are:

- We propose three highly scalable algorithms for QoSD, including two iterative algorithms, IG and AT, with approximation ratio $O(\gamma^{-1}(\ln T + h \ln n))$ and $O(\ln T + h \ln n)$ respectively, where γ is a metric measuring the concave property of edge weight functions and h is the maximum no. edges of a path connecting a pair; and SA, a probabilistic approximation algorithm returning $O(\frac{\ln T + h \ln d}{\gamma(1-e^{-\gamma})(1-\epsilon)})$ approximation result with high probability, where d is the maximum degree of G .
- When the edge weights are linear w.r.t budgets, we propose LR, a randomized rounding algorithm based on linear relaxation of the problem with $O(h \ln n)$ approximation guarantee.
- We extensively evaluate our algorithms on both synthetic networks and large-scale, real-world networks. All of our four algorithms are demonstrated to scale to networks with millions of nodes and edges in under a few hours and return nearly optimal solutions. Also, the experiments show the trade-off between our proposed algorithms in terms of runtime and quality of solution.

2 PROBLEM DEFINITION AND COMPLEXITY

We abstract the network using a directed graph $G = (V, E)$ with $|V| = n$ nodes and $|E| = m$ directed edges. Each edge e is associated with a function $f_e : \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$ which indicates the weight of e w.r.t a budget on e . f_e is monotonically increasing. We call $\mathbf{x} = \{x_1, \dots, x_m\}$ a budget vector where x_e is a budget on edge e . Under a budget vector $\mathbf{x}$, the length of a path p is defined as $\sum_{e \in p} f_e(x_e)$. QoSD is formally defined as follows:

*My T. Thai is the corresponding author.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGMETRICS'19 Abstracts, June 24–28, 2019, Phoenix, AZ, USA

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6678-6/19/06.

<https://doi.org/10.1145/3309697.3331470>

DEFINITION 1. Quality of Service Degradation (QoSD). Given a directed graph $G = (V, E)$, a set $F = \{f_e : \mathbb{Z}^{\geq} \rightarrow \mathbb{Z}^+\}$ of edge weight functions and a target set $S = \{(s_1, t_1), \dots, (s_k, t_k)\}$, determine a minimum budget x such that under x , the weighted, shortest-path between each pair in S exceeds a threshold T .

Hardness and Inapproximability. Since LB-MULTICUT is a special case of QoSD, QoSD is NP-hard. Furthermore, any inapproximability result of LB-MULTICUT or the Multicut problem is also the inapproximability of QoSD. We summarize those results as follows: (1) Kuhnle et al. [3] Let $T \geq 16$. Unless $NP \subseteq BPP$, there is no polynomial-time algorithm to approximate QoSD within a factor of $\frac{T}{6} - 1 - \epsilon$ for any $\epsilon > 0$; (2) Lee et al. [4]: When T is fixed and initial edge weights are uniform, QoSD is inapproximable within a factor of $\Omega(\sqrt{T})$ assuming the Unique Games Conjecture; (3) Chawla et al. [2]: There exist no $O(\log \log n)$ -approximation algorithm for QoSD unless $P = NP$.

3 ALGORITHMS

3.1 Iterative Solution

In general, this solution tackles two challenging tasks of QoSD. The first one is the number of feasible paths connecting pairs could be extremely large. Thus, instead of listing all paths, the solution builds a set of candidate paths incrementally and iteratively until blocking all such candidate paths is sufficient to separate all pairs in S . The second challenge is that the objective function to make the paths lengths exceed T can be non-submodular, depending on edge weight functions. Thus we devise two methods to block paths, which are *Iterative Greedy* (IG) and *Adaptive Trading* (AT).

Iterative Greedy is basically a greedy algorithm, in which we iteratively add a unit budget vector into x , which maximizes the marginal gain of the objective function, until x is sufficient to make all paths' length exceed T . However, as the objective function is neither submodular nor supermodular, the performance of IG strongly depends on the *concave ratio* γ of edge weight functions, which is defined as follows:

DEFINITION 2. (Concave ratio) The concave ratio of a set F of non-negative monotone increasing functions is the largest scalar $\gamma \in [0, 1]$ such that $f(x+1) - f(x) \geq \gamma \cdot (f(y+1) - f(y))$ for all $f \in F$, $f : \mathbb{Z}^{\geq} \rightarrow \mathbb{Z}^+$ and $0 \leq x \leq y$.

THEOREM 3.1. IG returns a solution within $O(\gamma^{-1}(h \ln n + \ln T))$ factor of the optimal solution for QoSD.

Adaptive Trading improves from IG in which for each iteration, AT adds a budget to an edge that maximizes the gain of objective function per unit of the additional budget. AT could discard the dependency on γ on performance guarantee but runtime complexity increase as a trade-off.

THEOREM 3.2. AT returns a solution within $O(h \ln n + \ln T)$ factor of the optimal solution for QoSD.

3.2 Sampling Approach

In essence, SA attempts to iteratively add a budget vector v , $\|v\| \leq q$ (q is an adjustable parameter), into x which maximizes the overall increasing lengths of feasible paths connecting pairs in S . Rather than an expensive listing all feasible paths, an estimator is employed by path sampling procedure to find the vector v . This process is repeated until the budget vector x is sufficient to guarantee all feasible paths have length at least T .

THEOREM 3.3. By generating a sufficient large number of sample paths in each sampling iteration, SA returns a solution within

$O(\frac{h \ln d + \ln T}{\gamma(1-e^{-\gamma})(1-\epsilon)})$ factor of optimum to the QoSD instance with probability at least $1 - \delta$ for $0 < \epsilon, \delta < 1$.

Interestingly, the approximation ratio of SA does not depend on q . So whatever the value of q is, the result of SA always has the same upper bound, which means a large value of q could reduce the number of sampling iterations but the number of sample paths in each iteration would increase as the trade-off.

3.3 Linear Rounding

LR considers a special case of QoSD where each edge weight function is linear w.r.t the budget on each edge. In overview, LR formulates the QoSD problem under an Integer Programming and obtains the optimal solution of its linear relaxation. Then a randomized rounding technique is applied to discretize the linear solution while guaranteeing the discrete budget is bounded within a certain factor to linear optimum.

THEOREM 3.4. Given fixed $0 < \delta < 1$, LR returns a solution within $O(h \ln n + \ln \delta^{-1})$ factor of optimum to the QoSD instance with probability at least $1 - \delta$.

4 EXPERIMENTAL EVALUATION

We evaluate our proposed approximation algorithms by 1) comparing their performance to an intuitive heuristic as there is no other solution to QoSD in a general case; and 2) comparing our algorithms to [3] as a special case of QoSD. The full experimental results can be found in [5].

We summarize several key results as follows: (1) In case of linear weight functions, LR always returned the best solution. The solution quality of IG and AT were worse than LR but usually by only a small factor. In addition, IG usually ran the fastest while the runtime of AT was more impacted by the varying of T than the other two; (2) In general cases where LR is no more applicable, AT was always the algorithm that returned the best quality of solution. IG was competitive to AT only if the weight functions tend to be more concave. In trade-off, IG performed much more faster than AT in most experiments; (3) SA was usually the worst among our algorithms in term of both the quality of solution and runtime. However, in several cases when the set of feasible paths was small or the input network was sparse, SA outperformed our other algorithms in runtime and the intuitive heuristics in solution quality.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and helpful suggestions. We would also like to show our gratitude to Dr. Figueiredo (UFRJ) for shepherding our paper. This work is supported in part by NSF EFRI-1441231, NSF CNS-1814614, and DTRA HDTRA1-14-1-0055.

REFERENCES

- [1] 2018. Congestion Attack on CV-based Traffic Signal Control. (2018). <https://sites.google.com/view/cav-sec/congestion-attack> Accessed: 2018-10-26.
- [2] Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D Sivakumar. 2006. On the hardness of approximating multicut and sparsest-cut. *computational complexity* 15, 2 (2006), 94–114.
- [3] Alan Kuhnle, Victoria G Crawford, and My T Thai. 2018. Network Resilience and the Length-Bounded Multicut Problem: Reaching the Dynamic Billion-Scale with Guarantees. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 1 (2018), 4.
- [4] Euiwoong Lee. 2016. Improved hardness for cut, interdiction, and firefighter problems. *arXiv preprint arXiv:1607.05133* (2016).
- [5] Lan N Nguyen and My T Thai. 2019. Network Resilience Assessment via QoS Degradation Metrics: An Algorithmic Approach. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3, 1 (2019), 1.