Tree Decomposition Based Local Search for Segment Routing

C. Dang[1,2], C. Bazgan[2], T. Cazenave[2], M. Chopin[1], P-H Wuillemin[3]

[1] Orange Labs, Chatillon, France
{chen.dang,morgan.chopin}@orange.com
[2] Universite Paris-Dauphine, PSL Research University, CNRS, UMR 7243, LAMSADE, F-75016, Paris, France
{cristina.bazgan,tristan.cazenave}@dauphine.psl.eu
[3] Sorbonne Universite, CNRS, UMR 7606, LIP6, F-75005, Paris, France
pierre-henri.wuillemin@lip6.fr

Mots-clés : *traffic engineering, segment routing, tree decomposition.*

1 Introduction

Traffic engineering is an important task in the optimization of the telecommunication networks to make more efficient use of the existing infrastructure and avoid congestion, thus improving the quality of service (QoS) of the network. Network managers usually employ complicated mechanisms to optimize the traffic flow as a part of the traffic engineering process. In the last past years, new routing protocols have been proposed to address the limitations of the traditional shortest path routing protocols (OSPF, IS-IS). Among them, Segment Routing (SR) is a recent technology proposed by IETF which attracted much attention. This new protocol allows one or more waypoints, or so called segments, to be applied on the packet, which needs to be visited in order during their transport toward the destination. Waypoints makes large-scale steering of individual packets possible, and allows more sophisticated control over the traffic.

The flexibility of SR raised many traffic engineering problems to optimize network performance with low costs. How to efficiently compute the segment list to minimize the maximum link utilization (MLU) is one of the most popular and the most important problems. Lower MLU usually leads to lower congestion, avoiding bottlenecks of network performance.

Many studies have addressed different aspects of this problem. Linear programming is widely used as an exact approach to solve this problem under different constraints, but it has scalability issue when solving large scale networks. Some local search based algorithms are also exploited for extremely fast optimization of minimizing MLU, who often provide a sub-optimal solution in less than a second.

2 Segment routing traffic engineering

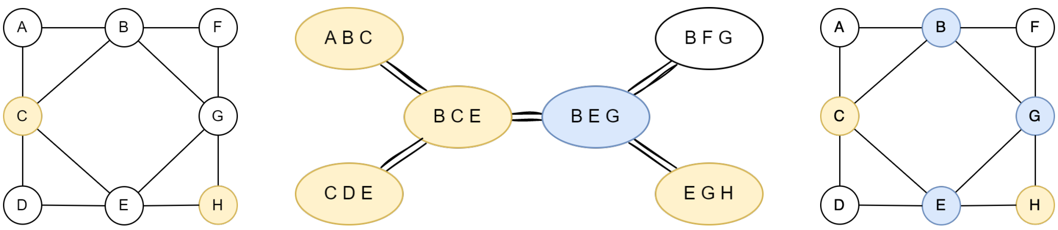
In this work, we are interested in the Segment Routing Traffic Engineering Problem (SRTEP) defined as follows. The problem aims to optimize the choice of the waypoints for each demand so as to minimize the maximum link utilization (MLU) in the network. More formally, we note the input network as $G(V, A)$, where every node $v \in V$ corresponds to a router while an arc $uv \in A$ corresponds to a link between routers u and v. Every arc uv is associated with a capacity c_{uv}. Let K be a set of demands to be routed in G. Every demand $k \in K$ is routed from its source s^k to its target t^k with a traffic volume d^k. The path of the demand k can be represented as a sequence of nodes $s^k = m_0^k, m_1^k, m_2^k, ..., m_n^k, m_{n+1}^k = t^k$, where $m_1^k, m_2^k, ..., m_n^k$ are the n waypoints of the demand k. Between each pair of successive nodes (m_i^k, m_{i+1}^k), the demand is routed according to the shortest paths. The traffic can be split according to the

Equal-Cost Multi-Path (ECMP) rule when there are more than one shortest paths. The load of an arc uv is denoted as $load_{uv}$, which is the sum of all the traffic flows passing through the arc. The MLU of the network is then defined as maximum link utilization of the network : $MLU = max(\frac{load_{uv}}{c_{uv}}), \forall uv \in A$.

The general segment routing problem was proved NP-hard [1].It was found in [2] from experimental results that 2 segments, i.e., one waypointper logical path, are sufficient for good performance of MLU minimization. The computational cost of solving this problem is high due to the very large search space. We focus here on how to select a subset of the nodes rather than making every node a waypoint candidate. It was proposed in [2] to use centrality-based metrics to select the most important nodes in the network, but this may not be appropriate because every demand, despite having different source and target, has the same small number of waypoint candidates, which may limit the effectiveness of the optimization.

3 Tree decomposition based local search

Toward the goal of designing scalable algorithms, we propose to use a tree decomposition of the network to extract a small set of nodes as waypoint candidates for each demand. More specifically, a tree decomposition $(X, T = (I, F))$ is generated from the network $G(V, A)$, where X is a family of subsets (bags) of V ; T is a tree whose set of nodes is X. We note the set of bags that contain node u as $X_{(u)} \subset X$. For each demand k, the path denoted P_k in T is the shortest path among the paths joining one bag in $X_{(s^k)}$ to one bag in $X_{(t^k)}$. We can then use all the nodes from the bags of P_k as the waypoint candidates for the demand k, which provides all the neighbor nodes of s^k and t^k and the shortest paths between them. Another way of choosing waypoint candidate is choosing in $X_{(s^k)} \cap X_{(t^k)}$ (if non empty).



(a) network with a demand from node C to H

(b) Tree decomposition of the network, with bags containing source node $X_{(C)}$, target node $X_{(H)}$, and its shortest path

(c) waypoint candidates for the demand

FIG. 1 – Example of midpoints candidates from the tree decomposition

We also provide a local search algorithm for SRTEP to validate this approach. Preliminary experimental results show that the tree decomposition provides a smaller search space for the problem, speeding up the algorithm while maintaining the high quality of the solution.

Références

[1] Renaud Hartert, Pierre Schaus, Stefano Vissicchio, and Olivier Bonaventure. Solving Segment Routing Problems with Hybrid Constraint Programming Techniques. In Principles and Practice of Constraint Programming. 2015.

[2] George Trimponias, Yan Xiao, Hong Xu, Xiaorui Wu, and Yanhui Geng. Centrality-Based Middlepoint Selection for Traffic Engineering with Segment Routing. ArXiv :1703.05907 [Cs], May. http ://arxiv.org/abs/1703.05907. 2019