OSPF Weight Setting Problem : extended models and exact algorithms using neural networks

Amal Benhamiche[1], Morgan Chopin[1], Sébastien Martin[2]

[1] Orange Labs, DATA&IA, Chatillon, France
{amal.benhamiche,morgan.chopin}@orange.com

[2] Huawei Technologies & co. 20 Quai du Point du Jour, Boulogne Billancourt, France
sebastien.martin@huawei.com

Mots-clés : *Traffic engineering, column generation, dynamic programming, neural networks*

Introduction. The arising of Network Virtualization will be a key enabler for the deployment of virtualized components capable of performing efficient path computation on behalf of the routers, thus allowing the optimization of operational IP networks. This perspective change draws again the traffic engineering community's attention to classical problems related to IP network optimization and raises the question of finding effective algorithms allowing to solve those problems for large scale networks. In particular, traffic routing in IP networks still draws heavily on shortest paths based routing protocols, such as open shortest path first (OSPF), and finding a set of link weights that induce shortest paths while also minimizing the network congestion is one of the key issues for the design of efficient IP networks.

In this context, we address the *Minimum Unsplittable Shortest Path Routing* (MinUSPR) problem. This problem is formally defined as follows. We consider given a bidirected graph $G = (V, A, c)$ representing an IP network where V is the set of nodes and A is the set of arcs. Every node $v \in V$ corresponds to a router and every arc $a \in A$ corresponds to a link connecting two routers with a given capacity $c_a \geqslant 0$. We are also given a set of commodities K representing a traffic matrix where each commodity $k \in K$ is characterized by an origin (router) node s^k, a destination (router) node t^k and a volume of traffic $D^k \geqslant 0$ (measured in Mbit/s) to be routed between s^k and t^k. The MinUSPR problem consists then in finding a set of weights to assign to the arcs of G such that (i) there is a unique shortest path in G for each commodity between its origin and destination according to the identified weights and (ii) the network congestion is minimum.

The MinUSPR problem has been widely studied in the literature for both its *splittable* and *unsplittable* versions. Several works propose algorithms to solve this problem using meta-heuristic, approximate and exact approaches (see [1] and the references therein). Our work extends the results proposed in [1] and aims at investigating hybrid approaches based on OR/ML methodologies for the problem.

Contributions. Our contributions can be summarized as follows

Path-based formulation. We first propose a non-compact MILP formulation for the problem obtained by applying a Dantzig-Wolfe decomposition to the compact (arc-flow) formulation presented in [2]. The resulting model contains a exponential number of (path) variables and we manage to solve its linear relaxation using a column generation procedure. More precisely, we solve iteratively the problem with a subset of columns. The remaining formulation is called *Restricted Master Problem* (RMP), and new columns (path variables) with negative reduced cost are generated dynamically though a shortest path-based pricing subproblem then added to the RMP.

Routing-path formulation. This model has also an exponential number of variables based on routing. A *routing* is an assignment of one path for each commodity. Let R be the set of all possible routings. The decision variables are the variables x_r that takes the value 1 if the routing r is used and 0 otherwise, for each $r \in R$. There is a variable $r_u^v \in \mathbb{R}^+$ for each $u, v \in V$ equal to the potential of node $u \in V$, that is the distance between router $u \in V$ and router $v \in V$. A variable $u_a^t \in \{0, 1\}$ for every $a \in A, t \in T$ equal to 1 if the link $a \in A$ belongs to a shortest path towards destination $t \in T$ and 0 otherwise. Finally, a variable $w_a \in \mathbb{Z}^+$ corresponding to the weight assigned to the arc a for each $a \in A$.

$$\min \quad \sum_{r \in R} x_r L_r \tag{1}$$

$$\lambda : \sum_{r \in R} x_r = 1 \tag{2}$$

$$\lambda_a^t : \sum_{r \in R} \alpha_{ra}^t x_r = u_a^t \qquad \forall a \in A, \forall t \in T, \tag{3}$$

$$w_{uv} - r_u^t + r_v^t \geq 1 - u_{uv}^t \qquad \forall a \in A \;\; \forall t \in T \tag{4}$$

$$w_{uv} - r_u^t + r_v^t \leq M(1 - u_{uv}^t) \qquad \forall a \in A \;\; \forall t \in T \tag{5}$$

where M is a big value. Inequality (2) ensures that only one routing is selected. Inequalities (3) make the link between variables x and u. Inequalities (4) and (5) ensure a feasible weight on each link. The pricing problem consists in finding a routing r (multi-commodity flow) such that $\lambda + \sum_{a \in A} \sum_{t \in T} \lambda_a^t y_a^t > L_r$ where y_a^t is equal to 1 if at least one commodity with the destination t use the arc a and 0 otherwise. It can be solved by a variant of the classical multi-commodity flow path based model.

Neural networks. In the column generation loop we add a neural network to find a heuristic solution at each iteration. Let r be the routing added by one loop of the column generation. The neural network is composed by an input vector that corresponds to the utilization of each link given by the routing r and an output vector that corresponds to the weights (w_a). Thus the input vector and output vector have the same size. We also apply our neural network to the linear relaxation of path variables of the path-based formulation. To train the neural network, we generate on the same input graph a dataset with few commodities that can be solved optimally using exact methods.

Generating dataset through dynamic programming. A dynamic programming algorithm based on a *tree decomposition* for solving the MinUSPR problem was proposed in [2]. More precisely, the algorithm computes an optimal set of weights in time $2^{O(\omega^8|K|)} \cdot n$ where ω is the treewidth of the input graph. Recall, that the "treewidth" is a graph invariant that measures how close a graph is of being a tree. For instance, trees have the smallest possible treewidth value 1 while clique graphs of n vertices have the largest treewidth value n. Thus, the algorithm is particularly suited to solve the problem on "tree-like" (sparse) networks. A promising research direction would be to use this algorithm to quickly generate a dataset for our neural network approach with examples of sparse networks and "small" number of commodities. Computing such dataset may seem restrictive at first, but most of the real life networks are sparse. Furthermore, we think that learning on a small, but carefully chosen, set of demands is enough to help the neural network to generalize well.

Future works include the integration of these approaches into a branch-and-price algorithm and an experimental study to assess its effectiveness on state of the art instances.

Références

[1] Andreas Bley, Bernard Fortz, Éric Gourdin, Kaj Holmberg, Olivier Klopfenstein, Michal Pióro, Artur Tomaszewski and Hakan Ümit. *Optimization of OSPF Routing in IP Networks.* Graphs and Algorithms in Communication Networks 2010 : 199-240

[2] Amal Benhamiche, and Morgan Chopin. Toward Scalable Algorithms for the Unsplittable Shortest Path Routing Problem. *CoRR abs/2006.04324*, *https ://arxiv.org/abs/2006.04324*, 2020.