New exact approaches for the Unsplittable Shortest Path Routing Problem

Amal Benhamiche, Morgan Chopin

Orange Innovation, Orange Labs, Châtillon, France
{amal.benhamiche,morgan.chopin}@orange.fr

Mots-clés : *Traffic engineering, Mixed Integer Programming, Dynamic Programming, Tree-width.*

Introduction. In this work, we consider the Delay Constrained Unsplittable Shortest Path Routing (D-USPR) problem which arises in the field of *traffic engineering* for IP networks. This problem consists, given a directed graph and a set of commodities, to compute a set of routing paths and the associated administrative weights such that each commodity is routed along the unique shortest path between its origin and its destination, according to these weights. More formally, we consider given a bidirected graph $G = (V, A)$ that represents an IP network topology. Every node $v \in V$ corresponds to a router while an arc $a = uv \in A$ represents a logical link between router nodes u and v. Every arc uv is associated a capacity (bandwidth) denoted by $c_{uv} \geqslant 0$ and a latency value denoted $\delta_{uv} \geqslant 0$. We let K denote a set of commodities (traffic demands) to be routed over the graph G. Every commodity k is defined by a pair (s^k, t^k) with s^k, t^k being the origin and destination of k, respectively, along with the traffic volume $D^k \geqslant 0$ to be routed from s^t to t^k and a a maximum delay value $\Delta^k \geqslant 0$. The D-USPR problem is to find a set of weights to assign to the arcs of G and a set of routing paths induced by those weights such that (i) there is a unique shortest path satisfying the delay constraints for each commodity according to the identified weights and (ii) the network congestion is minimum.

We propose two exact algorithms to solve the problem. First, we present a compact MILP formulation for the problem, extending the work in [3, 2] along with some valid inequalities to strengthen its linear relaxation. Then, we further propose a dynamic programming algorithm based on a *tree decomposition* of the graph. To the best of our knowledge, this is the first exact combinatorial algorithm for the problem. Finally, we outline the main steps of an hybrid exact algorithm combining both approaches.

Basic MILP formulation and valid inequalities. Let x_a^k be a binary variable that takes the value 1 if commodity k is routed along a path using arc a and 0 otherwise. We define the binary variables u_a^t that takes the value 1 if a belongs to a shortest path towards destination t and 0 otherwise. We further let w_{uv} denote the weight assigned to the arc uv and r_u^v be the potential of node u, that is the distance between node u and node v. The D-USPR problem is then equivalent to the following MILP formulation :

$$\min L \tag{1}$$

$$s.t. \sum_{a \in \delta^+(v)} x_a^k - \sum_{a \in \delta^-(v)} x_a^k = \begin{cases} 1 & \text{if } v = s^k, \\ -1 & \text{if } v = t^k, \\ 0 & \text{otherwise.} \end{cases} \quad \begin{array}{l} \forall v \in V, \\ \forall k \in K, \end{array} \tag{2}$$

$$\sum_{k \in K} D^k x_a^k \leqslant c_{uv} L, \forall a \in A, \tag{3}$$

$$\sum_{a \in A} \delta_a x_a^k \leqslant \Delta^k, \forall k \in K, \tag{4}$$

$$\sum_{a \in \delta^+(v)} u_a^t \leqslant 1, \forall v \in V, \forall t \in T, \tag{5}$$

$$x_a^k \leqslant u_a^{t^k}, \forall a \in A, \forall k \in K, \tag{6}$$

$$u_a^t \leqslant \sum_{k \in K, t^k = t} x_a^k, \forall a \in A, \forall t \in T, \tag{7}$$

$$w_{uv} - r_u^t + r_v^t \geqslant 1 - u_{uv}^t, \forall uv \in A, \forall t \in T, \tag{8}$$

$$w_{uv} - r_u^t + r_v^t \leqslant M(1 - u_{uv}^t), \forall uv \in A, \forall t \in T \tag{9}$$

Trivial and integrity constraints are omitted due to space limit. The objective (1) is to minimize the load of the most loaded link, denoted L. Inequalities (2) ensure that a unique path is associated to each commodity k and (3) express the load over an arc a. Inequalities (4) are the delay constraints over the routing paths while (5) and (6)-(7) are anti-arborescence and linking constraints, respectively. In particular, inequalities (5) ensure that there is at most one path traversing any node v towards a given destination $t \in T$, which is necessarily implied by Bellman property. Constraints (8) and (9) guarantee that the weight of any arc used by a shortest path towards a destination t corresponds to the difference of potentials between the end nodes of this arc and larger otherwise. We further strengthen the formulation by adding two families of valid inequalities namely *subpath consistency* and *node precedence* constraints.

The basic formulation (2)-(9) was implemented in Python using Cplex 12.8 with the default settings and NetworkX graph library. We have tested our formulation on several instances derived from SNDlib (sndlib.zib.de) topologies of variying size and density along with the following features. First by (i) solving the basic formulation, second by introducing (ii) the subpath consistency inequalities, (iii) the node-precedence inequalities and (iv) both families of valid inequalities, to the basic formulation.

A dynamic programming algorithm. We design a dynamic programming (DP) algorithm based on *tree decomposition* for solving the D-USPR problem. Observe that the problem is trivial in the case where the input graph is a tree since there can only be one path to route any demand. Unfortunately, it is not possible to generalize this positive result to graphs of bounded treewidth since the problem is NP-complete even on bidirected rings [1]. However, we show that there exists a polynomial-time algorithm for graphs of bounded treewidth with bounded number of demands and delays. This algorithm has not only a theoretical interest but is also expected to boost the effectiveness of a branch-and-cut (B&B) algorithm for solving the above formulation (especially on “tree-like” networks) as described in the following two steps :

1. Run the DP algorithm until an optimal solution is obtained or a certain stop criteria is reached (*e.g* time limit).
2. If no optimal solution was found by the DP then populate a pool of partial routing paths that can be used to generate dynamically cuts throughout a B&B algorithm applied over the MILP formulation.

Conclusion and future directions. Although the obtained results are promising there is still room for improvements. First, we expect that solving the formulation using a branch-and-cut algorithm will substantially improve the performance of the MILP. Second, we also plan to implement and validate the performance of the proposed hybrid DP/B&B approach.

References.

[1] A. Bley. Approximability of unsplittable shortest path routing problems. *Networks*, 54(1) :23–46, 2009.

[2] A. Bley. An integer programming algorithm for routing optimization in IP networks. *Algorithmica*, 60(1) :21–45, 2011.

[3] A. Bley, B. Fortz, E. Gourdin, K. Holmberg, O. Klopfenstein, M. Pióro, A. Tomaszewski, and H. Ümit. *Optimization of OSPF Routing in IP Networks*, pages 199–240. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.