A constraint-programming framework for network routing

Pierre Bauguion, Youcef Magnouche, Sébastien Martin, Jérémie Leguay

Huawei Technologies France, Boulogne-Billancourt, France.

{pierre.bauguion, youcef.magnouche, sebastien.martin, jeremie.leguay}@huawei.com

Mots-clés : *Combinatorial optimization, constraint programming, multi-commodity flow, network routing, telecommunications.*

1 Introduction

Over the years, the size of telecommunication networks is continuously increasing and new applications are emerging, which is leading to ever-evolving traffic engineering requirements (multi-constraint routing, resilience to failures, etc.). Most of the time, these requirements can be efficiently addressed using dedicated combinatorial algorithms based on linear programming. Nevertheless, these approaches cannot deal with problems that do not get convenient properties on their solution spaces (such as linearity, convexity, etc...). Furthermore, as telecommunication ecosystems evolve quickly, routing algorithms need to be frequently upgraded to meet new requirements with minimal effort.

Emerged in the 70's, Constraint programming (CP) allows great flexibility in the types of constraints that can be considered for a combinatorial problem as it is not bounded by any space property or paradigm. Besides, there is virtually an infinite number of constraint types that can be independently implemented and introduced into the optimization model. These two qualities make this paradigm an interesting compromise to overcome some problem specificities while providing the expected adaptability. This motivated DEFO [1], a declarative framework based on CP to solve a wide range of traffic engineering problems. It consists of a Domain Specific Language (DSL) and the corresponding CP model to find an acceptable subset of (forwarding) routing paths for a set of demands. The modeling and the DSL have been tailored for Segment Routing [2], a source routing extension to IPv6.

In this talk, we propose a tailored CP solver dedicated to multi-elementary path(s) computation subject to a subset of constraints that can be declared -or even implemented- by the user. Unlike in DEFO, the path variable is described *explicitly*, which is particularly convenient to enforce specific constraint satisfaction (such as resource lower bound constraints) but also global constraints binding different paths (such as capacity), while controlling the memory consumption of the feasible domain(s). The specialization of such a solver allows to optimize all data structures so the core operations (filtering, propagation,...) can be conducted with minimal computational effort. This advantage allows to cover a large search space and can provide an optimal solution in a reasonable amount of time without requiring heuristics.

2 The solver

As mentioned before, unlike [1], our solution should be composed of a set of explicit elementary paths (one for each demand) satisfying the set of declared constraints. To exploit the specific structure of the problem, we construct variables encoding the choice for an elementary path. As a consequence, the initial domain of such a variable will be the whole network graph, which will be reduced as the search goes, until it forms a complete path or an unfeasible solution.

More formally, feasibility conditions can be summarized as follows:

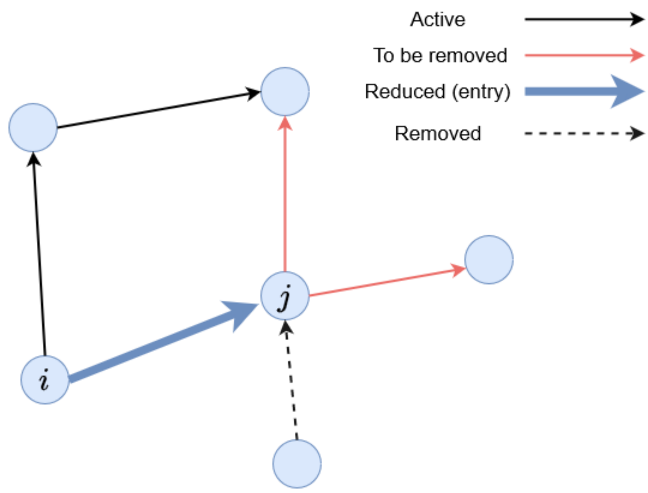

FIG. 1: *By removing the arc (i, j) from the domain one can observe that no more ongoing active arcs exist, meaning that we can remove all arcs outgoing from j.*

Définition 1 *Given a graph $G = (V, A)$, a set of demands K made of triplets $(s^k, t^k, d^k) \in V \times V \times \mathbb{R}, \forall k \in K$. By denoting $x_a^k \in \{0, 1\}$ the variable that states if the arc $a \in A$ is used by the demand $k \in K$, one can state that the solution is* feasible *if the following properties are satisfied:*

$$\begin{aligned}
&\forall k \in K, \forall v \in V \setminus \{s^k, t^k\} && \sum_{a \in \delta^+(v)} x_a^k = \sum_{a \in \delta^-(v)} x_a^k \\
&\forall k \in K, && \sum_{a \in \delta^+(s^k)} x_a^k = 1 \\
&\forall k \in K, \forall M \subset V, && \sum_{i,j \in M} x_{i,j}^k \leq |M| - 1 \\
&\forall c \in \mathcal{C}, && X \in \mathcal{X}^c
\end{aligned} \tag{1}$$

where $\delta^+(v)$ (resp. $\delta^-(v)$) represents the outgoing (resp. ongoing) arcs of $v \in V$, X the vector of $x_a^k, \forall k \in K, a \in A$ and $\mathcal{C}$, the set of constraints (with $\mathcal{X}^c$ the set of feasible values of X according to the constraint $c \in \mathcal{C}$).

From those conditions, one can derive elementary rules (e.g. Fig. 1), to check the inner consistency of each variable domain at any time. On top of practical efficiency, calling all these rules through a propagation and filtering scheme appears to be polynomial.

We integrate these elementary rules into a more general constraint propagation architecture to solve a wide range of routing problems with the presented conditions and show the efficiency and flexibility of such an approach. We will present numerical results for the resolution of a constrained resource shortest path problem with multiple resources and compare the performance of the solver against best-in-class algorithms. On top of its user-friendly capabilities, we show that the performances can outperform the latter algorithms.

References

[1] Hartert R., Vissicchio S., Schaus P., Bonaventure O., Filsfils C., Telkamp T., and Francois P. (2015). A Declarative and Expressive Approach to Control Forwarding Paths in Carrier-Grade Networks. ACM SIGCOMM Computer Communication Review. 45. 15-28.

[2] Ventre, Pier Luigi, et al. "Segment routing: a comprehensive survey of research activities, standardization efforts, and implementation results." IEEE Communications Surveys & Tutorials 23.1 (2020): 182-221.