Piecewise-linear reformulations for the availability-aware Service Function Chain routing problem

R. Colares[1], A. Benamiche[2], Y. Carlinet[2], N. Perrot[2]

[1] Université Clermont Auvergne, CNRS, Clermont Auvergne INP, Mines Saint-Etienne, LIMOS, 63000 Clermont-Ferrand, France
rafael.colares_borges@uca.fr

[2] Orange Labs, 46 Avenue de la République, 92320 Châtillon, France

Keywords : *MIP reformulations, network resilience, 5G networks*

1 Introduction

In the context of telecommunication networks, a Service Function Chain (SFC) can be seen as an origin-destination traffic demand with some additional requirements. These requirements manifest as a predefined sequence of Network Functions (NF) which must be visited (in the specified order) along the SFCs origin-destination route. Typical examples of NFs include firewalls, video optimizers, load balancers, and parental control.

The virtualization of NFs enables on-demand execution of a NF within a virtual server, dissociating it from any dedicated hardware. Although this virtualization implies more flexibility and significant cost reductions for the service providers, it also creates an operational challenge as Virtual Network Function (VNF) are more prone to failures when compared to legacy dedicated hardware. To make things worse, if a single VNF goes down, the whole service chain is impacted. Moreover, new 5G use-cases (*e.g.*, communication between autonomous vehicles, e-health applications, data manipulation within smart factories and smart cities) are part of the so-called Ultra-Reliable Low-Latency Communication (URLLC) services, which require strict levels of end-to-end availability. For this reason, backup VNFs must be placed on the network so that the services can still be ensured even in case of node failures. Since the placement of backup network functions incurs extra costs, a substantial challenge for infrastructure and service providers such as Orange is therefore to efficiently decide the how to place VNFs throughout the network so that a given set of SFCs can achieve the required availability levels.

2 Problem definition

We next define the Availability-aware Service Function Chain Routing (ASFCR) problem. We are given a directed, loopless, connected graph $G = (V, A)$ representing the telecommunication network, where each node $v \in V$ has a capacity $C_v \in \mathbb{R}^+$, and an availability $0 < A_v < 1$, (*i.e.*, a risk of $1 - A_v$ of being unavailable). Moreover, let $\mathcal{F}$ be the set of Virtual Network Function (VNF) types, where each VNF $f \in \mathcal{F}$ has a resource consumption $R^f \in \mathbb{R}^+$, and a placement cost $P_v^f \in \mathbb{R}^+$ for each node $v \in V$. Finally, let K be the set of SFC demands to be routed, where each demand $k \in K$ is defined by (i) an origin $o_k \in V$ and a destination $d_k \in V$, (ii) a bandwidth $B_k \in \mathbb{R}^+$, (iii) a required availability $\mathcal{A}^k \in [0, 1]$, and (iv) an ordered set of distinct VNFs $F^k \subseteq \mathcal{F}$ that must be visited.

The availability of a path π routing a given SFC is defined as the probability that all its VNFs are properly running. Let $S(\pi) \subseteq V$ denote the set of nodes hosting a VNF for the given path π. Then, the availability of path π is given by

$$a(\pi) = \prod_{v \in S(\pi)} A_v.$$

Usually, a single path is not enough for ensuring the SFC's required availability. In this case, a set of paths $\mathcal{P}^k$ is assigned to the SFC k. In order to make the routing paths independent, we say that two distinct paths π_1 and π_2 in $\mathcal{P}^k$ cannot use the same node for hosting a VNF. In other words, $S(\pi_1)$ and $S(\pi_2)$ must be disjoint. With this in mind, the SFC availability can now be defined as the probability that at least one of its assigned paths is available, that is,

$$a(\mathcal{P}^k) = 1 - \prod_{\pi \in \mathcal{P}^k} \Big(1 - a(\pi)\Big).$$

A set of paths $\mathcal{P}^k$ is then said to secure SFC $k \in K$ if and only if $a(\mathcal{P}^k) \geq \mathcal{A}^k$. A feasible solution to the ASFCR problem consists of (i) a VNF placement on nodes, and (ii) for each SFC, the associated set of paths passing through the requested VNFs in the right order that secures the SFC. Additionally, node capacities, path latency and arc bandwidth volumes must be verified. The goal is to find a feasible solution that minimizes the VNF placement cost.

3 Problem formulation and contributions

A natural compact nonlinear formulation for the ASFCR problem can be easily obtained. Due to the page limit, we provide here only the availability constraints that are responsible for rendering the formulation nonlinear. They are stated as follows:

$$a_p^k \leq z_p^k \qquad \forall p \in P, k \in K \tag{1}$$

$$a_p^k \leq \prod_{v \in V} A_v^{y_{vp}^k} \qquad \forall p \in P, k \in K, \tag{2}$$

$$\prod_{p \in P} (1 - a_p^k) \leq 1 - \mathcal{A}^k, \qquad \forall k \in K, \tag{3}$$

$$a_p^k \geq 0 \qquad \forall p \in P, k \in K, \tag{4}$$

$$y_{vp}^k, z_p^k \in \{0, 1\} \qquad \forall v \in V, p \in P, k \in K, \tag{5}$$

where y_{vp}^k is a binary variable stating whether or not a VNF is hosted on node $v \in V$ within the p-th path assigned to SFC $k \in K$, z_p^k is a binary variable stating whether or not the p-th path is used for routing SFC $k \in K$, and a_p^k is a real variable denoting the end-to-end availability of the p-th path is used for routing SFC $k \in K$.

The presence of the nonlinear inequalities (2) and (3) makes the formulation difficult to be solved by standard commercial MIP solvers such as CPLEX or Gurobi. In this presentation, we show that without introducing any new variable, one can get rid of the nonlinear constraints by considering an exponential number of linear constraints.

Unfortunately, the resulting MIP remains extremely hard to solve exactly. Inspired by the works from [2], we also explore ways of approximating the problem's feasible region with a polynomial number of constraints. This is done by considering a logarithmic reformulation of the original nonlinear program, which results in a new compact nonlinear program where the only nonlinear components are logarithmic functions. We can then apply piecewise linear approximations to these logarithmic functions to obtain a linear reformulation. Different strategies are investigated to build such piecewise linear functions, including a preprocessing step that uses the shortest path algorithm to minimize approximation errors [1]. An exact framework is finally proposed, exploring the exact and approximated reformulations together.

References

[1] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. Network flows. 1988.

[2] Alain Billionnet. Mathematical optimization ideas for biodiversity conservation. *European Journal of Operational Research*, 231(3):514–534, 2013.