Received: 12 October 2020 | Accepted: 14 October 2020 | Published on: 10 November 2020

DOI: 10.1002/net.21999

RESEARCH ARTICLE

WILEY

An efficient approach to optimization of semi-stable routing in multicommodity flow networks

Artur Tomaszewski[1] | **Michał Pióro**[1] | **Davide Sanvito**[2] | **Ilario Filippini**[2] | **Antonio Capone**[2]

[1]Instytut Telekomunikacji, Politechnika Warszawska, Warsaw, Poland
[2]Dipartimento di Elettronica, Informazione e Bioingegneria, Milan, Italy

Correspondence
Michał Pióro, Institute of Telecommunications, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland.
Email: m.pioro@tele.pw.edu.pl

Funding information
National Science Centre, Poland, Grant/Award Number: 2017/25/B/ST7/02313

Abstract

Ideally, the network should be dynamically reconfigured as traffic evolves. Yet, even within the software defined network paradigm, network reconfigurations cannot be too frequent due to a number of reasons related to route consistency, forwarding rules instantiation, individual flows dynamics, traffic monitoring overhead, and so on. In this paper, we focus on the fundamental issue of deciding whether, when, and how to reconfigure the network while traffic evolves. We consider a problem of optimizing semi-stable routing in the capacitated multicommodity flow network when one may use at most a given maximum number of routing configurations (called routing clusters) and when each routing configuration must be used for at least a given minimum amount of time. We propose an efficient solution approach based on routing cluster generation that provides a tight lower bound on the minimum of a selected objective function (like maximum link delay or a sum of link delays) and suboptimal solutions very close to the calculated bound. The approach scales well with the size of the network.

KEYWORDS

integer programming, multicommodity flows, semi-stable routing, software defined networks, time-dependent traffic

1 | INTRODUCTION

The dynamic nature of network traffic caused by daily fluctuations is the origin of a crucial trade-off between routing optimality and frequency of network reconfiguration. However, network operators have traditionally privileged static (or stable; the word stable is used in a popular sense) routing approaches, like oblivious routing [2] and robust routing [10, 18, 20], that apply a single routing configuration based on"worst case" traffic conditions. This unavoidably creates over-provisioning and suboptimal utilisation of network capacity.

Recently, software-defined networking (SDN) has provided tools for making online network reconfiguration a potentially viable solution: dynamic reconfigurations of routing can be applied at the network devices to optimize performance as the traffic evolves [5, 8, 9, 15]. However, reconfiguring the network too frequently, in general, can affect network state consistency, since reprogramming flow rules can take longer than the reconfiguration period.

A group of hybrid approaches, often referred to as semi-stable routing, have been recently proposed to combine static and dynamic routing [3, 6, 16, 17, 21]. Using a limited set of routing configurations, each designed for specific time intervals, allows for reducing the penalty of assuming the "worst case" traffic conditions, and, simultaneously, for controlling the reconfiguration frequency. As a result, the optimization problem of selecting a sequence of routing configurations, and timepoints when the consecutive routing configurations must be activated, arises.

In this paper we consider the problem of optimizing routing in the capacitated multicommodity flow network, in which demand volumes change periodically over an ordered set of timepoints. Following the semi-stable routing approach, we analyze

a specific version of the problem where one may use at most a given maximum number of routing configurations and where each routing configuration must be used for at least a given minimum number of consecutive timepoints, in order to meet the maximum network reconfiguration frequency constraint. Referring to the set of consecutive timepoints as the routing (timepoint) cluster, we name this problem the semi-stable routing cluster design problem (SSRCDP). In the considered version of SSRCDP, the optimization objective is to minimize the network delay, that is, the sum of timepoint delays (over all timepoints) where for a single timepoint its delay is defined as the sum of the link delays. Although we have chosen the link delay as the network congestion measure, the solution method we propose is general enough to cope with other types of the congestion measure.

The works on semi-stable routing available in the literature usually exhibit one of the following limitations: (a) they ignore the time domain by not providing any limit on the reconfiguration rate [3, 17, 21], (b) the number of created clusters is limited and reconfiguration timepoints are arbitrary [3, 6]. Other semi-stable approaches have more recently been proposed to overcome these limitations [4, 14]. In particular, the techniques presented there compute a set of routing configurations that can be combined together to generate a routing configuration for a new traffic realization. However, combining multiple configurations may, in particular, generate a large number of paths and flow split ratios that might not be feasible to handle by network devices.

For SSRCDP we propose a solution method based on cluster generation that delivers provably near-optimal solutions, that is, the method also provides a good lower bound on the network delay. In addition, the proposed method scales well with the size of the network and can be effectively applied to networks of large sizes. The problem formulation, the solution method, and an illustrative realistic numerical example are presented in this work.

The rest of the paper is organized as follows. After introducing basic notation and formulating the SSRCDP problem (Section 2), in Section 3 we discuss its exact solution methods. Then, in Section 4, we describe in detail the proposed solution approach. Numerical results illustrating the efficiency of our approach are presented in Section 5. Finally, in Section 6, we conclude the paper and discuss directions of future work. Additional theoretical considerations that can lead to improving the proposed approach are described in Appendix A.

Finally, we note that the presented paper is a substantially extended version of our conference paper [19].

2 | NOTATION AND PROBLEM FORMULATION

2.1 | Notation

The notation used in the paper, summarized in Table 1, is as follows. Let the capacitated multicommodity flow network be modeled with a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{D})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of directed links (where $c(e) \geq 0$, $e \in \mathcal{E}$, is the capacity of link e), and $\mathcal{D}$ is the set of directed demands (where $o(d), t(d), d \in \mathcal{D}$, are, respectively, the originating node and the terminating node of demand d). Next, let $\mathcal{P}(d)$ be a given set of (routing) paths in graph $\mathcal{G}$ that are admissible for demand $d, d \in \mathcal{D}$, where each path $p \in \mathcal{P}(d)$ connects the demand's origin $o(d)$ with its termination $t(d)$; $\mathcal{P}$ will denote the set of all admissible paths, that is, $\mathcal{P} := \bigcup_{d \in \mathcal{D}} \mathcal{P}(d)$. Additionally, let $\mathcal{Q}(e, d) \subseteq \mathcal{P}(d), e \in \mathcal{E}, d \in \mathcal{D}$, denote the set of admissible paths of demand d that use link e. Finally, let $\mathcal{T} := \{0, 1, \ldots, T-1\}$ be the set of consecutive *timepoints*, and let $h(d, t) \geq 0, d \in \mathcal{D}, t \in \mathcal{T}$, be the volume of demand d to be realized at timepoint t.

We assume that the *routing configuration* is defined by vector $x := (x_{dp})_{d \in \mathcal{D}, p \in \mathcal{P}(d)}$, where x_{dp} is the fraction (i.e., $x_{dp} \in [0, 1]$) of the volume of demand d that is assigned to path p. The following condition must thus hold:

$$\sum_{p \in \mathcal{P}(d)} x_{dp} = 1 \quad d \in \mathcal{D}. \tag{1}$$

Then, if routing configuration x is used at timepoint $t \in \mathcal{T}$, the *utilization* $w_e^t(x)$ of link e at t is defined as:

$$w_e^t(x) := \frac{1}{c(e)} \sum_{p \in \mathcal{Q}(e,d)} h(d, t) x_{dp} \quad e \in \mathcal{E}. \tag{2}$$

Note that the quantity $\sum_{p \in \mathcal{Q}(e,d)} h(d, t) x_{dp}$ in the right-hand side of definition (2) expresses the load of link e at timepoint t; note that, in general, this load can be greater than the capacity of the link. Furthermore, let $F : [0, +\infty) \to [0, +\infty)$ be an increasing convex piece-wise linear function with $F(0) = 0$. Note that, as explained in Section 3.1, this function is of the form $F(w) = \max \{a(k)w + b(k) : k = 1, 2, \ldots, K\}$. We will call $F(w)$ the *delay function* (see [7, 13]) as it is supposed to measure the packet delay on a link for a given link utilization w. Finally, the quantity

$$z^t(x) := \sum_{e \in E} F(w_e^t(x)) \tag{3}$$

will be called the *timepoint delay* at timepoint t.

We may now introduce the notion of the *(timepoint) cluster* $C(t, l)$ with parameters t (the timepoint at which the cluster starts) and l (the length, or size, of the cluster). Namely, $C(t, l)$ is the set of l consecutive timepoints that starts at timepoint t.

TABLE 1 Notation

Notation	Description
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{D})$	Network graph, $\mathcal{V}$—set of nodes, $\mathcal{E}$—set of (directed) links, $\mathcal{D}$ - set of (directed) demands
$\mathcal{T} = \{0, 1, \dots, T-1\}$	Set of timepoints
$c(e)$	Capacity of link e ($e \in \mathcal{E}$)
$h(d, t)$	Volume of demand d to be realized at timepoint t ($d \in \mathcal{D}, t \in \mathcal{T}$)
$o(d)$, $t(d)$	Originating node and terminating node, respectively, of demand $d \in \mathcal{D}$
$\mathcal{P}(d)$	Set of admissible (routing) paths for demand $d \in \mathcal{D}$
$\mathcal{Q}(e, d)$	Set of paths in $\mathcal{P}(d)$ that contain link e ($e \in \mathcal{E}, d \in \mathcal{D}$)
$\mathcal{P} = \bigcup_{d \in \mathcal{D}} \mathcal{P}(d)$	Set of all admissible paths
$x = (x_{dp})_{d \in \mathcal{D}, p \in \mathcal{P}(d)}$	Routing configuration (vector of path flows)
$w_e^t(x)$, $F(w_e^t(x))$	Utilization of link e at timepoint t and the corresponding delay
$z^t(x)$	Timepoint delay (sum of link delays at timepoint $t \in \mathcal{T}$) implied by routing configuration x
$\mathcal{C}$	Clusters composed of timepoints
$t(\mathcal{C})$, $l(\mathcal{C})$	Starting timepoint and length (respectively) of cluster $\mathcal{C}$ ($t(\mathcal{C}) \in \mathcal{T}, l(\mathcal{C}) \in \{1, 2, \dots, T\}$)
$\mathcal{C}(t, l)$	Cluster with $t(\mathcal{C}) = t, l(\mathcal{C}) = l$, $\mathcal{C}(t, l) = \{t, t \oplus 1, \dots, t \oplus (l-1)\}$ ($\oplus$ denotes addition modulo T)
$\mathscr{C}$	Control family - family of control clusters $\mathcal{C}$
$x(\mathcal{C})$	Routing configuration used in cluster $\mathcal{C}$
$z(\mathcal{C}, x) = \sum_{t \in \mathcal{C}} z^t(x)$	Cluster delay for cluster $\mathcal{C}$ with routing configuration x
$Z(\mathcal{C})$	Cluster delay of $\mathcal{C}$ minimized over all routing configurations x ($Z(\mathcal{C})$ is a solution of RP($\mathcal{C}$))
$Z(\mathcal{C}\|\infty) = \sum_{t \in \mathcal{C}} Z(\{t\})$	A lower bound for $Z(\mathcal{C})$
$\mathscr{R}$	Partition of the set of timepoints $\mathcal{T}$ into at most N ($1 \le N \le \frac{T}{L}$) routing clusters $\mathcal{R}$, each of length at least L ($\|\mathcal{R}\| \ge L, \mathcal{R} \in \mathscr{R}$)
$\mathscr{R}$	Routing clusters - clusters belonging to routing partition $\mathscr{R}$
$z(\mathscr{R}) = \sum_{\mathcal{R} \in \mathscr{R}} z(\mathcal{R}, x(\mathcal{R}))$	Network delay for partition $\mathscr{R}$ (with routing configurations $x(\mathcal{R}), \mathcal{R} \in \mathscr{R}$)
$Z(\mathscr{R}) = \sum_{\mathcal{R} \in \mathscr{R}} Z(\mathcal{R})$	Minimum network delay for partition $\mathscr{R}$
SSRCDP	Semi-stable routing cluster design problem
Z^*	Minimum of $Z(\mathscr{R})$ over all partitions $\mathscr{R}$ (Z^* is the optimal solution value of SSRCDP)
RP($\mathcal{U}$)	Routing problem for $\mathcal{U} \subseteq \mathcal{T}$ (finding routing configuration realizing $Z(\mathcal{U})$)
APP($\mathscr{C}$)	Approximative partitioning problem using control cluster family $\mathscr{C}$
$\mathscr{R}(\mathscr{C})$	Routing partition solving APP($\mathscr{C}$)
$Y(\mathscr{C})$	Minimum objective value of APP($\mathscr{C}$) (lower bound for SSRCDP)
CGA	Cluster generation algorithm
$\mathbb{B}, \mathbb{Z}^+, \mathbb{R}^+$	$\mathbb{B} = \{0, 1\}$, $\mathbb{Z}^+ = \{0, 1 \dots\}$, $\mathbb{R}^+$: nonnegative real numbers

Hence, $\mathcal{C}(t, l) := \{t, t \oplus 1, \dots, t \oplus (l-1)\}$, where $\oplus$ denotes addition modulo T (i.e., the timepoints are counted modulo T). For a given cluster $\mathcal{C} = \mathcal{C}(t, l)$, let $t(\mathcal{C}) = t$ and $l(\mathcal{C}) = l$ denote, respectively, the start and the length of $\mathcal{C}$.

Suppose that the same routing configuration (denoted by $x(\mathcal{C}) = (x(\mathcal{C})_{dp})_{d \in \mathcal{D}, p \in \mathcal{P}(d)}$) is used for all timepoints of cluster $\mathcal{C}$. Then, we will call $\mathcal{C}$ a *(stable) routing cluster*. For a routing cluster $\mathcal{C}$ and a given routing configuration x, the quantity

$$z(\mathcal{C}, x) := \sum_{t \in \mathcal{C}} z^t(x) \tag{4}$$

will be referred to as *(routing) cluster delay* (of cluster $\mathcal{C}$ under routing configuration x). The minimum cluster delay (i.e., the value of $z(\mathcal{C}, x)$ minimized over all routing configurations x will be denoted by $Z(\mathcal{C})$.

2.2 | Problem formulation

The *semi-stable routing cluster design problem* (SSRCDP) we consider is this: given $\mathcal{G}$, $\mathcal{P}$, $\mathcal{T}$, and a pair of positive integer numbers $N \le T$ and $L \le T$, find

- a *partition* $\mathscr{R}$ of the set of timepoints $\mathcal{T}$ into at most N (nonempty) routing clusters $\mathcal{R}$ (i.e., $|\mathscr{R}| \le N$), each of length at least L (i.e., $|\mathcal{R}| \ge L, \mathcal{R} \in \mathscr{R}$)
- a routing configuration $x(\mathcal{R})$ for each routing cluster $\mathcal{R} \in \mathscr{R}$

so as to minimize the *network delay* $Z(\mathscr{R}) := \sum_{\mathcal{R} \in \mathscr{R}} Z(\mathcal{R})$.

In the following, the minimum value of the network delay resulting from SSRCDP will be denoted by Z^*. Note that the assumptions on N, L, and T imply that for a given L, $1 \le L \le T$, it is sufficient to consider only the values of N not greater

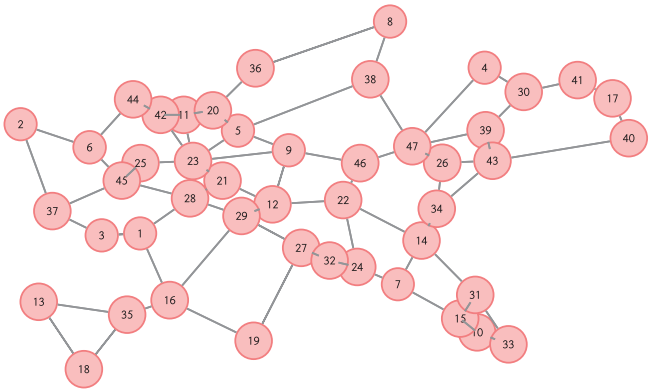

FIGURE 1 Network topology [Color figure can be viewed at wileyonlinelibrary.com]

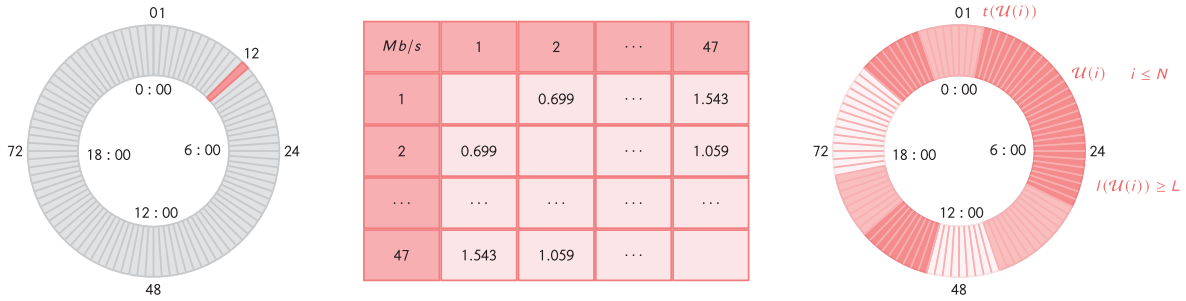

FIGURE 2 Timepoints, traffic matrices and clusters [Color figure can be viewed at wileyonlinelibrary.com]

than $\left\lfloor \frac{T}{L} \right\rfloor$, because $N = \left\lfloor \frac{T}{L} \right\rfloor$ is the maximum number of disjoint clusters with length not less than L forming a partition of $\mathcal{T}$. Because of that, the condition $N \leq \left\lfloor \frac{T}{L} \right\rfloor$ will be assumed throughout this paper.

2.3 | Example

Figure 1 depicts a real network topology linking 47 cities in a European Union country that will be used in the numerical study presented in Section 5. The network consists of $|\mathcal{V}| = 47$ nodes (routers) linked with $|\mathcal{E}| = 140$ directed links, each of capacity 4 Gbps, and $|\mathcal{D}| = 47 \times 46 = 2162$ traffic demands, corresponding to all ordered pairs of nodes in $\mathcal{V}$. For this network we consider a 24-hour time horizon and traffic measurements taken every 15 minutes. Hence, we deal with $T = 24 \times 4 = 96$ timepoints (each corresponding to a 15-minute time interval) and the set of timepoints $\mathcal{T}$ is equal to $\{0, 1, \ldots, 95\}$. Set $\mathcal{T}$ is depicted on the left-hand side of Figure 2, where the distinguished slot $t = 12$ corresponds to the time interval between 4 : 00 and 4 : 15 am.

The traffic matrices for timepoints $t \in \mathcal{T}$ were derived from real traffic measurements taken every 15 minutes on a specific weekday (a Wednesday in 2018), obtained from the network operator. The entries of each such matrix are given in Mb/s and express the bitrate required by the corresponding ordered node-pairs, averaged over the 15 minute time interval corresponding to the timepoint t, for which the matrix is specified. The entries of all 96 obtained traffic matrices range from 0.028 to 1712.044 Mb/s. The traffic matrix for timepoint $t = 12$ is partly depicted in the middle of Figure 2.

Note that for each timepoint $t \in \mathcal{T}$, the demand volumes $h(d, t), d \in \mathcal{D}$, used in the optimization formulations throughout this paper represent the entries of the above described traffic matrix specified for a given t when these entries are numbered linearly according to the lexicographical order, that is, when $d = 1$ corresponds to node pair (1, 2), $d = 2$ corresponds to (1, 3), and so on.

In the numerical study of Section 5 we will assume that the maximal number of clusters is equal to $N = 8$, and the minimum cluster length to $L = 8$. This means that we accept at most 8 changes of the routing configuration during 24 hours and require that a routing configuration change can occur after the hold-off time of at least 2 hours. A feasible partition $\mathscr{R}$ of $\mathcal{T}$ into 8 clusters $\mathcal{U}(i) : i = 1, 2, \ldots, 8$, is illustrated on the right-hand side of Figure 2. Note that the length of any cluster is not smaller than 8 and that timepoint $t = 0$ is not a starting point of any cluster of partition $\mathscr{R}$.

3 | EXACT SOLUTION METHODS FOR SSRCDP

3.1 | Fixed partition subcase

If the sets forming a partition $\mathcal{R}$ of set $\mathcal{T}$ are given and fixed, SSRCDP reduces to finding a routing configuration $x(\mathcal{R})$ minimizing $Z(\mathcal{R})$ for each routing cluster $\mathcal{R} \in \mathcal{R}$, and this can be done independently for each cluster. Thus, we first analyze the problem of finding an optimal routing configuration for a given cluster. We aim, in particular, at deriving some properties that can be useful in formulating and solving the original semi-stable routing cluster design problem.

Finding an optimal routing configuration for a given set of (not necessarily consecutive) timepoints $\mathcal{U} \subseteq \mathcal{T}$ is identical to a well-known problem of finding an optimal routing configuration for a given set of traffic matrices. Such a *routing problem* (denoted by RP($\mathcal{U}$)) consists in finding a single routing configuration $x(\mathcal{U})$ that minimizes the sum of timepoint delays over $\mathcal{U}$:

Problem RP($\mathcal{U}$)

$$Z(\mathcal{U}) = \min \quad \sum_{t\in\mathcal{U}} \left(\sum_{e\in\mathcal{E}} z_e^t \right) \tag{5a}$$

$$\sum_{p\in\mathcal{P}(d)} x_{dp} = 1 \qquad d \in \mathcal{D} \tag{5b}$$

$$w_e^t \geq \frac{1}{c(e)} \sum_{p\in Q(e,d)} h(d,t)\ x_{dp} \qquad t \in \mathcal{U}, e \in \mathcal{E} \tag{5c}$$

$$z_e^t \geq a(k) w_e^t + b(k) \qquad t \in \mathcal{U}, e \in \mathcal{E}, k \in \mathcal{K} \tag{5d}$$

$$x_{dp} \in [0,1] \qquad d \in \mathcal{D}, p \in \mathcal{P}(d) \tag{5e}$$

$$z_e^t, w_e^t \in \mathbb{R}^+ \qquad t \in \mathcal{U}, e \in \mathcal{E}. \tag{5f}$$

Above, variables $x_{dp}, d \in \mathcal{D}, p \in \mathcal{P}(d)$, define a routing configuration $x(\mathcal{U})$ common for all timepoints in $\mathcal{U}$, variables $w_e^t, t \in \mathcal{U}, e \in \mathcal{E}$, express link utilizations at the timepoints in $\mathcal{U}$, and variables $z_e^t, t \in \mathcal{U}, e \in \mathcal{E}$, specify the corresponding link delays (hence, the term $\sum_{e\in\mathcal{E}} z_e^t$ in the objective function (5a) expresses the delay at timepoint t). In (5d), parameters $a(k), b(k), k \in \mathcal{K} := \{1, 2, \ldots, K\}$, determine the delay function $F(z) := \max\{a(k)z + b(k) : k \in \mathcal{K}\}$, where $b(1) = 0 > b(2) > \cdots > b(K)$, $0 < a(1) < a(2) < \cdots < a(K)$.

Note that RP($\mathcal{U}$) is a linear programming (LP) problem in a noncompact formulation that can be solved to optimality using the column (path) generation approach (see [12, 13]) based on a shortest path algorithm for the pricing problem: to generate a new path $p \in \mathcal{P}(d)$ for demand $d \in \mathcal{D}$ and price out a new variable x_{dp} one has to find a shortest path in graph $\mathcal{G}$ between the end nodes of d, with the costs of links equal to $\frac{1}{c(e)} \sum_{t\in\mathcal{U}} h(d,t)\pi_e^t$, $e \in \mathcal{E}$, where π_e^t are the optimal values of dual variables associated with constraint (5c). A path is added to the problem if its cost is less than λ_d - the optimal value of dual variable associated with constraint (5b).

It is well known that RP($\mathcal{U}$) can be reformulated as a compact LP problem using the node-link notation. Such a formulation uses link flows (instead of path flows) and does not require column generation (see [13]). However, according to our experience, for networks of realistic size, such as the one considered in Section 2.3 or larger, the most efficient way to solve the routing problem is to use the noncompact formulation (5) embedded in an iterative algorithm. Problem (5) is the master problem, which is solved alternately with the pricing problem that finds new paths to be added to the master. It is important that after the pricing problem is solved the consecutive master problem can use the optimal simplex basis from the previous iteration for the warm start. With such an implementation, it happens that for a given cluster $\mathcal{U}$ considered for the network example from Section 2.3, it takes approximately $l(\mathcal{U})$ seconds to solve RP($\mathcal{U}$) using the hardware/software configuration described in Section 5.

Solving SSRCDP for a given partition $\mathcal{R}$ consists in finding the values $Z(\mathcal{R})$ for each cluster $\mathcal{R} \in \mathcal{R}$, through solving RP($\mathcal{R}$), and then computing the network delay $Z(\mathcal{R}) = \sum_{\mathcal{R}\in\mathcal{R}} Z(\mathcal{R})$. When $\mathcal{R} = \{\mathcal{T}\}$ then the resulting SSRCDP solution is referred to as *static routing*, and when $\mathcal{R} = \{\{t\} : t \in \mathcal{T}\}$ then the resulting solution is referred to as *dynamic routing*. Note that static routing and dynamic routing impose, respectively, the upper and the lower bound on the optimal value of the SSRCDP objective function when all possible partitions $\mathcal{R}$ of set $\mathcal{T}$ are considered.

We end this section with the following simple observation.

Remark. For any two sets $\mathcal{U}', \mathcal{U}$ such that $\mathcal{U}' \subseteq \mathcal{U} \subseteq \mathcal{T}$, the inequality

$$Z(\mathcal{U}') \leq \sum_{t\in\mathcal{U}'} z^t(x^*(\mathcal{U})) \tag{6}$$

holds, where $x^*(\mathcal{U})$ is the optimal routing configuration resulting from RP($\mathcal{U}$), and $z^t(x^*(\mathcal{U})), t \in \mathcal{U}$, are defined by (2). The reason is that if $Z(\mathcal{U}')$ were larger than $\sum_{t\in\mathcal{U}} z^t(x^*(\mathcal{U}))$, then configuration $x^*(\mathcal{U})$, when applied to $\mathcal{U}'$, would decrease the value of $Z(\mathcal{U}')$. Clearly, when $\mathcal{U} = \mathcal{U}'$ the right-hand side of (6) is equal to $Z(\mathcal{U}')$.

3.2 | Solving SSRCDP through dynamic programming

3.2.1 | Linear partition case

Consider a subfamily $\mathcal{T}(0)$ of all partitions of the set of timepoints $\mathcal{T}$ such that each partition $\mathcal{R}$ in $\mathcal{T}(0)$ contains a cluster of the form $C(0, l)$ for some $L \le l \le T$. Let $\langle t', t''\rangle$, where $0 \le t' \le t'' \le T-1$, denote the cluster $\{t', t'+1, \dots, t''\}$, that is, cluster $C(t', l)$ where $l := t'' - t' + 1$. It follows that each partition $\mathcal{R}$ in $\mathcal{T}(0)$ is of the form $\mathcal{R} = \{\langle t(0), t(1)\rangle, \langle t(1)+1, t(2)\rangle, \dots, \langle t(n-1), t(n)\rangle\}$, where, $n \le N$ (the number of clusters in $\mathcal{R}$ cannot be larger than N), $t(0) = 0$, $t(n) = T-1$, and $t(k) - t(k-1) + 1 \ge L$, $k = 1, 2, \dots, n$ (cluster length cannot be smaller than L). It follows that the number J of admissible clusters, that is, clusters that can be used by the partitions in subfamily $\mathcal{T}(0)$, is equal to

$$J = \begin{cases} 1, & \text{if } L \le T \le 2L-1 \\ 3, & \text{if } T = 2L \\ 2(T-2L+1)+1, & \text{if } 2L+1 \le T \le 3L-1 \\ \frac{(T-3L+1)(T-3L+2)}{2} + 2(T-2L+1)+1, & \text{if } T \ge 3L. \end{cases} \tag{7}$$

In the first case of feasible T ($L \le T \le 2L-1$) there is only one admissible cluster equal to $\langle 0, T-1\rangle$, simply because there is no room for two clusters of size at least L. In the second case there are only three admissible clusters: $\langle 0, T-1\rangle$, $\langle 0, L-1\rangle$, and $\langle L, 2L-1\rangle$. In the third case, any partition in $\mathcal{T}(0)$ contains either just one cluster $\langle 0, T-1\rangle$ (note that $\{\langle 0, T-1\rangle\}$ is a one-element partition which is always feasible), or two clusters (since there is no room for three admissible clusters of size at least to L). Note that any two-element partition consists of one cluster of the form $\langle 0, t\rangle$ and one cluster of the form $\langle t+1, T\rangle$, where $t = L-1, L, \dots, T-L-1$. The number of such pairs of clusters is $T-2L+1$, and each of them determines two unique clusters; hence $J = 2(T-2L+1)+1$ in (7) is correct. Finally, in the fourth case, the second term on the right-hand side, that is, $2(T-2L+1)$, counts the number of clusters that can occur in the two-cluster partitions in $\mathcal{T}(0)$, and the last term, that is, 1, represents cluster $\langle 0, T-1\rangle$. The first term, in turn, gives the number of clusters that are admissible in the rest of partitions, that is, the number of clusters that can occur between cluster $\langle 0, t'\rangle$ and $\langle t'', T-1\rangle$, where $t' \ge L-1$, $t'' \le T-1-L$, and $L \le t'' - t' \le T-2L$. This number is equal to the number of clusters of size at least L that are contained in set $\{L, L+1, \dots, T-L-1\}$, and this is exactly what formula $\frac{(T-3L+1)(T-3L+2)}{2}$ expresses.

Now let $W(t, c)$ denote the optimal objective function value of SSRCDP when the set of timepoints is reduced to $\{0, 1, \dots, t-1\}$ (where $L \le t \le T$) and only c clusters (where $c \le N$, and, as previously assumed, $N \le \left\lfloor \frac{T}{L} \right\rfloor$) from the family of admissible clusters $\{\langle t(0), t(1)\rangle : 0 \le t(0), t(1) \le t-1, t(1) - t(0) + 1 \ge L\}$ can be used (note that this family is empty for $t < L$). The following list of equalities allows one to find $W(T, N)$, that is, the optimal objective function value of the original problem SSRCDP, through dynamic programming [11]. Note that below, instead of $Z(\langle t(0), t(1)\rangle)$ (i.e., the quantity that expresses the optimal value of the objective function of the routing problem RP($\langle t(0), t(1)\rangle$) formulated above in (5)), we will simply write $Z(t(0), t(1))$.

$$W(t, 1) = Z(0, t-1) \qquad L \le t \le T \tag{8a}$$

$$W(t, c) = W(t, 1) \qquad L \le t \le 2L-1, 2 \le c \le N \tag{8b}$$

$$W(2L, c) = Z(0, L-1) + Z(L, 2L-1) \qquad 2 \le c \le N \tag{8c}$$

$$W(t, c) = \min_{L-1 \le \tau \le t-L-1} \{W(t-\tau-1, c-1) + Z(t-\tau-1,\ t-1)\} \qquad 2L \le t \le T, \quad 2 \le c \le \left\lfloor \frac{t}{L} \right\rfloor. \tag{8d}$$

$$W(t, c) = W\left(t, \left\lfloor \frac{t}{L} \right\rfloor\right) \qquad \left\lfloor \frac{t}{L} \right\rfloor < c \le N. \tag{8e}$$

The first equality simply says that when only one cluster can be used then it must be the cluster $\langle 0, t-1\rangle$. The second equality states that when $L \le t \le 2L-1$ then only one cluster can be used (even when $c > 1$), since otherwise at least one cluster would be of length smaller than L (note that the cluster to be used must be $\langle 0, t-1\rangle$). Equality (8c), in turn, gives the proper value of $W(t, c)$ for $t = 2L$. Note that in this case also the cluster $\langle 0, 2L-1\rangle$ is feasible but since $Z(0, L-1) + Z(L, 2L-1) \le Z(0, 2L-1)$, (8c) is correct. Next, equality (8d) defines the basic recursive relation between the values of $W(t, c)$. Note that for $t = 2L$,

equality (8c) is a special case of equality (8d). Finally, equality (8e) takes into account the fact that the number of clusters in a partitioning of $\{0, 1, \ldots, t-1\}$ cannot exceed $\left\lfloor \frac{t}{L} \right\rfloor$.

Observe that optimal solutions of SSRCDP for $T \le 2L$ are directly implied by equalities (8b) and (8c). For $T > 2L$, a dynamic programming (DP) algorithm that recursively uses equalities (16) for $t \ge 2L+1$ and $c \ge 2$ will lead to the optimal SSRCDP objective function value $W(T, N)$ (where $N \le \left\lfloor \frac{t}{L} \right\rfloor$) and, at the same time, to an optimal (and feasible) partition of the set $\mathcal{T}$ into routing clusters. It should be mentioned here that equalities (8) are an extended version of the recursive equality presented in [1], where the DP approach was used for the above considered special case of SSRCDP, that is, assuming $\mathcal{T}(0)$. Finally, note that for applying the DP algorithm we first need to solve, in the preprocessing phase, the routing problem RP(C) for every cluster $C \in \mathcal{T}(0)$.

3.2.2 | Cyclic partition case

Now, we proceed to the general (cyclic) case when the partitions of $\mathcal{T}$ do not have to necessarily contain a cluster of the form $C(0, l)$. In this case the number of (feasible) clusters we need to consider for DP is given by the formula:

$$J = \begin{cases} 1, & \text{if } L \le T \le 2L-1 \\ T(T-2L+1)+1, & \text{if } T \ge 2L. \end{cases} \tag{9}$$

The formula follows from the observation that, besides the cluster identical with $\mathcal{T}$, a cluster belongs to one of allowable partitions of $\mathcal{T}$ if, and only if, its length is between L and $T-L$, so that for a given starting timepoint $t \in \mathcal{T}$ there are $T-2L+1$ of such clusters.

SSRCDP can be solved by applying the DP algorithm described above for the partition family $\mathcal{T}(0)$. This is done by applying the algorithm T times, for each partition subfamily $\mathcal{T}(\theta), \theta = 0, 1, \ldots, T-1$, where for a given θ each partition in subfamily $\mathcal{T}(\theta)$ must contain a cluster of the form $C(\theta, l)$ for some $L \le l \le T-L$. Observe that now, for each $\theta = 1, 2, \ldots, T-1$, the equalities in (8) take the following form:

$$W(t,1) = Z(\theta, \theta \oplus (t-1)) \qquad L \le t \le T \tag{10a}$$

$$W(t,c) = W(t,1) \qquad L \le t \le 2L-1, 2 \le c \le N \tag{10b}$$

$$W(2L,c) = Z(\theta, \theta \oplus (L-1)) + Z(\theta \oplus L, \theta \oplus (2L-1)) \qquad 2 \le c \le N \tag{10c}$$

$$W(t,c) = \min_{L-1 \le \tau \le t-L-1} \{ W(t-\tau-1, c-1) + Z(\theta \oplus (t-\tau-1), \theta \oplus (t-1)) \} \qquad 2L \le t \le T, \qquad 2 \le c \le \left\lfloor \frac{t}{L} \right\rfloor \tag{10d}$$

$$W(t,c) = W\left(t, \left\lfloor \frac{t}{L} \right\rfloor\right) \qquad \left\lfloor \frac{t}{L} \right\rfloor < c \le N. \tag{10e}$$

Above, $W(t, c)$ denotes the optimal objective function value of SSRCDP when the set of timepoints is reduced to $\{\theta, \theta \oplus 1, \ldots, \theta \oplus (t-1)\}$ (and each partition must contain a cluster of the form $C(\theta, l)$ for some $L \le l \le t-L$).

An alternative way for applying DP the cyclic case of SSRCDP is to consider all feasible clusters C that contain timepoint $t = 0$. Let us denote the family of such clusters by $\mathcal{F}(0)$. Clearly, any $C \in \mathcal{F}(0)$ is either equal to $\mathcal{T}$ or, assuming that $T \ge 2L$, can be divided into two subclusters C' and C'' (so that $C = C' \cup C'', C' \cap C'' = \emptyset$) where:

$$C' = C(0, k(1)), \quad C'' = C(T-k(2), k(2)) \tag{11a}$$

$$1 \le k(1) \le T-L \tag{11b}$$

$$\max\{0, L-k(1)\} \le k(2) \le T-L-k(1). \tag{11c}$$

Note that the above conditions imply that $l(C) = k(1) + k(2)$ and hence $L \le l(C) \le T-L$. This means that C is a feasible cluster that starts at $t = 0$ and ends at $t = k(1)-1$ when C'' is empty (which is possible since $k(2)$ can be equal to 0); otherwise, when $k(2) > 0$ it starts at $t = T-k(2)$ and ends at $t = k(1)-1$. Observe that the number of clusters in family $\mathcal{F}(0)$ is equal to

$$P := L + (L+1) + \cdots + (L+T-L) = \frac{1}{2}((T-L+1)(T-L) - L(L-1)). \tag{12}$$

It follows that SSRCDP can be solved by successive application of DP to the sets of timepoints $\mathcal{T}(C) := \{k(1), k(1)+1, \ldots, T-k(2)-1\}$ separately for each $C \in \mathcal{F}(0)$. Let $t' := k(1)$, $T' := T - l(C)$, and $N' = \min\left\{N-1, \left\lfloor \frac{t}{L} \right\rfloor\right\}$. The recursive formulae appropriate for the set of timepoints $\mathcal{T}(C)$ are analogous to (8) and involve the values $W(t,c)$ for $L \le t \le T'$, $1 \le c \le N'$:

$$W(t,1) = Z(t', t'+t-1) \qquad L \le t \le T' \quad (13a)$$

$$W(t,c) = W(t,1) \qquad L \le t \le 2L-1, 2 \le c \le N' \quad (13b)$$

$$W(2L,c) = Z(t', t'+L-1) + Z(t'+L, t'+2L-1) \qquad 2 \le c \le N \quad (13c)$$

$$W(t,c) = \min_{L-1 \le \tau \le t-L-1} \{ W(t-\tau-1, c-1) + Z(t'+t-\tau-1,\ t'+t-1) \} \qquad 2L \le t \le T',\quad 2 \le c \le \left\lfloor \frac{t}{L} \right\rfloor \quad (13d)$$

$$W(t,c) = W\left(t, \left\lfloor \frac{t}{L} \right\rfloor\right) \qquad \left\lfloor \frac{t}{L} \right\rfloor < c \le N'. \quad (13e)$$

Note that the optimal objective value of SSRCDP when a particular cluster $C \in \mathcal{F}(0)$ is forced to be used in partitions of $\mathcal{T}$ is equal to $Z(C) + W(T', M')$, where $W(T', M')$ results from the DP algorithm applied to $\mathcal{T}(C)$ using formulae (13).

Finally, let us note that the first of the two above described alternative ways of using DP for solving SSRCDP in the cyclic case involves T runs of the DP algorithm, while the second one involves P (where P is calculated using formula (12)) DP runs. Although T is in general considerably smaller than P, the second option can be more efficient as it involves smaller DP problems. More precisely, in the first option the DP table to be calculated has always $T \times N$ elements, while in the second option the DP table for a given $C \in \mathcal{F}(0)$ has $(T - l(C)) \times \min\left\{N-1, \left\lfloor \frac{t}{L} \right\rfloor\right)$ elements.

3.3 | Comments

Even though the above described DP algorithm is polynomial, its application can be excessively time consuming. In the example discussed in Section 2.3 we have $T = 96$ and $L = 8$, which, according to formula (9), means that the family of feasible clusters contains $J = 7777$ clusters to be preprocessed for the DP algorithm. Since, as mentioned in Section 3.1, solving the routing problem RP(C) takes approximately $l(C)$ seconds, and the average cluster length is 48.5, the preprocessing time itself is (approximately) equal to $7777 \times 48.5 = 377{,}184.5$ seconds, that is, more than 105 hours. This makes the DP approach impractical.

Note that already for the special case when only the partitions from subfamily $\mathcal{T}(0)$ are considered, formula (7) implies that there are $J = 2791$ feasible clusters, and the preprocessing time for these clusters will be of the order of tens of hours.

Observe that the DP execution time will grow enormously when longer time horizons $\mathcal{T}$ are considered. For example, for a one-week time horizon with 5-minute measurement intervals and at least 2-hour interval between activating two consecutive routing reconfigurations, we have $T = 2016$, $L = 24$ and, according to formula (9), the value of J becomes equal to 3,969,505. Since in this case the average cluster length is equal to 1008.5, the total preprocessing time becomes (approximately) equal to 1,112,012 hours.

Let us also note that SSRCDP could be formulated as a compact mixed-integer programming (MIP) problem and, in theory, directly solved to optimality using a MIP solver. This, however, would not be efficient in practice due to an excessive number of binary variables in the formulation and a poor linear relaxation. We have confirmed this observation while trying a number of such formulations.

Thus, to be able to effectively cope with SSRCDP we need some other approach. What we propose in this paper is to decompose the semi-stable routing cluster design problem into a cluster design problem and a routing design problem, where the routing design problem is solved only for a small number of potential clusters. Although this is a rather straightforward idea, the real issue is how to couple these two subproblems. This issue will be dealt with in the remaining part of the paper.

4 | EFFICIENT SUBOPTIMAL APPROACH

4.1 | Approximation problem

The suboptimal approach to SSRCDP presented below consists in formulating an optimization problem that determines a suboptimal partition $\mathscr{R}$ of the set of timepoints $\mathcal{T}$ into timepoint clusters, where for each timepoint cluster $\mathcal{R} \in \mathscr{R}$, an optimal routing configuration $x^*(\mathcal{R})$ will then be found by solving problem RP($\mathcal{R}$) in a postprocessing phase; $\mathcal{R}$ will be called a routing partition and its elements will be called routing clusters.

Let u^t ($t \in \mathcal{T}$) be a binary variable that equals 1 if, and only if, t is a start of a routing cluster, and 0 otherwise, and let y^t ($t \in \mathcal{T}$) be a continuous variable that approximates (from below) the minimum timepoint delay at t. Let $\mathcal{C}$ be a fixed subfamily of the family of all timepoint clusters (the family $\mathcal{C}$ will be called a *family of control clusters* or simply a *control family*), and let $Z(C|\infty) := \sum_{t\in C} Z(\{t\})$ for each $C \in \mathcal{C}$. Note that $\mathcal{C}$ does not have to be a partition of the set of timepoints $\mathcal{T}$.

The *approximate partitioning problem* APP($\mathcal{C}$) of finding a routing partition $\mathcal{R}$ that minimizes the *approximated* network delay is as follows:

Problem APP(C)

$$Y(C) = \min \sum_{t\in T} y^t \tag{14a}$$

$$\sum_{t\in\mathcal{T}} u^t \le N \tag{14b}$$

$$\sum_{0\le k\le L-1} u^{t\oplus k} \le 1 \qquad t \in \mathcal{T} \tag{14c}$$

$$U^C = \sum_{1\le k<l(C)} u^{t(C)\oplus k} \qquad C \in \mathcal{C} \tag{14d}$$

$$Y^C = \sum_{t\in C} y^t \qquad C \in \mathcal{C} \tag{14e}$$

$$y^t \ge Z(\{t\}) \qquad t \in \mathcal{T} \tag{14f}$$

$$Y^C \ge Z(C) + (Z(C|\infty) - Z(C)) \cdot U^C \qquad C \in \mathcal{C} \tag{14g}$$

$$u^t \in \mathbb{B}, \;\; y^t \in \mathbb{R}^+ \qquad t \in \mathcal{T} \tag{14h}$$

$$U^C \in \mathbb{Z}^+, Y^C \in \mathbb{R}^+ \qquad C \in \mathcal{C}. \tag{14i}$$

Constraints (14b) and (14c) guarantee that each feasible binary vector $u := (u^t)_{t\in\mathcal{T}}$ specifies a partition of the set of timepoints $\mathcal{T}$ which contains at most N clusters, each of length at least L. Let us denote such a partition by $\mathcal{R}$. Then, constraint (14d) defines integer variables U^C that specify with how many clusters in partition $\mathcal{R}$ a given cluster C from the control family $\mathcal{C}$ intersects. Note that when $U^C = 0$ then cluster $\mathcal{C}$ intersects with only one cluster in partition $\mathcal{R}$ (which is the cluster that contains cluster C), when $U^C = 1$ then cluster C intersects with exactly two clusters in partition $\mathcal{R}$, and so on. Additionally, constraint (14e) defines the quantity Y^C: an approximated cluster delay for each control cluster C.

Constraints (14f) and (14g) specify two kinds of *valid inequalities*, that is, inequalities that are satisfied by the maximal link utilizations $z^t(x(\mathcal{R}))$, $\mathcal{R} \in \mathcal{R}$, $t \in \mathcal{T}$, determined (through definition (3)) for any partition $\mathcal{R}$ and any set of routing configurations $x(\mathcal{R})$, $\mathcal{R} \in \mathcal{R}$ (satisfying condition (1)).

The inequality in constraint (14f) holds since, for any given $t \in \mathcal{T}$, $Z(\{t\})$, as the optimal solution of RP($\{t\}$), provides the absolute lower bound on the timepoint delay at t. Thus, (14f) is a valid inequality. Note also that (14f) implies that $Y^C = \sum_{t\in C} y^t \ge Z(C|\infty)$.

Now, observe that the right-hand side of inequality in (14g) defines an affine function of variable U^C (defined by (14d)). Let us denote this function by A. Since $Z(C) \ge Z(C|\infty)$ (by definition of $Z(C|\infty)$), function A is nonincreasing, and strictly decreasing when $Z(C) > Z(C|\infty)$. Since $A(0) = Z(C)$, for $U^C = 0$ the inequality in (14g) reduces to $\sum_{t\in C} y^t \ge Z(C)$. Moreover, condition $U^C = 0$ means that $C \subseteq \mathcal{R}$ for some $\mathcal{R} \in \mathcal{R}$, and hence, by the Remark in Section 3.1, implies the inequality $\sum_{t\in C} z^t(x(\mathcal{R})) \ge Z(C)$. This means that for $U^C = 0$ the inequality in (14g) is valid.

Next, since $A(1) = Z(C|\infty)$, for $U^C = 1$, the inequality in (14g) reduces to $\sum_{t\in C} y^t \ge Z(C|\infty)$, which, as mentioned above, is already implied by (14f). This means that in this case (14g) is valid as well. Moreover, since A is nonincreasing, $A(U) \le A(1)$ for $U > 1$ and this means that (14g) is valid for all $U^C > 1$. Thus, (14g) is valid for all possible values of U^C, and this finally implies that APP($\mathcal{C}$) is a relaxation of SSRCDP so that its optimal solution value $Y(\mathcal{C})$ is a lower bound for the minimum network delay Z^*.

Observe that the reason for using the particular form of the inequality in (14g) is that it is stronger than inequality

$$\sum_{t\in C} y^t \ge Z(C) \;\; (1 - U^C) \qquad C \in \mathcal{C} \tag{15}$$

as far as the linear relaxation of APP($\mathcal{C}$) is concerned.

In order to find a (suboptimal) solution of SSRCDP we can first solve APP($\mathcal{C}$) for a given control family $\mathcal{C}$, for example consisting of all clusters with length not greater than L, obtaining a routing partition $\mathcal{R}(\mathcal{C})$ of $\mathcal{T}$. Then, we can solve the routing

problem RP($\mathcal{R}$) for each $\mathcal{R} \in \mathscr{R}(\mathscr{C})$, and determine $Z(\mathscr{R}(\mathscr{C}))$, that is, the minimum of the network delay for partition $\mathscr{R}(\mathscr{C})$. An issue is, however, how to find a way for extending the current control family $\mathscr{C}$ in order to decrease the so obtained $Z(\mathscr{R}(\mathscr{C}))$. The following three basic properties of formulation APP($\mathcal{C}$) will help resolving this issue.

Proposition 1. *Let $\mathscr{C}$ be an arbitrary control family. For any routing partition $\mathcal{R}$ of $\mathcal{T}$ into at most N routing clusters with length at least L each, there exists a feasible solution $u = (u^t)_{t\in\mathcal{T}}, y = (y^t)_{t\in\mathcal{T}}$ of problem APP($\mathscr{C}$) that defines the partition $\mathcal{R}$ and such that for each $\mathcal{R} \in \mathscr{R}$, $y^t = z^t(x(\mathcal{R}))$, $t \in \mathcal{R}$, that is, y^t is equal to the timepoint delay at t implied by the routing scheme $x(\mathcal{R})$ of the routing cluster $\mathcal{R}$.*

Proof. For each $t \in \mathcal{T}$ we put $u^t = 1$ if $t = t(\mathcal{R})$ for some $\mathcal{R} \in \mathscr{R}$; otherwise, we put $u^t = 0$. Clearly, the so obtained vector u satisfies constraints (14b), (14c) and uniquely defines the routing partition $\mathcal{R}$. Also, the vector y specified in the thesis of the proposition is feasible for APP($\mathscr{C}$) since, as explained above, inequalities (14f) and (14g) are valid for any routing family $\mathscr{R}$ in question. ∎

Proposition 2. *Let $\mathscr{R}(\mathscr{C})$ be the routing partition determined by an optimal solution of APP($\mathscr{C}$), that is, by u^*. Then,*

$$Y(\mathscr{C}) \le Z^* \le Z(\mathscr{R}(\mathscr{C})), \tag{16}$$

where $Y(\mathscr{C}) = \sum_{t\in\mathcal{T}} y^{t*}$ is the optimal objective function value of APP($\mathscr{C}$), Z^* is the optimal objective function value of SSRCDP (i.e., the minimum network delay), and $Z(\mathscr{R}(\mathscr{C})) = \sum_{\mathcal{R}\in\mathscr{R}(\mathscr{C})} Z(\mathcal{R})$.

Proof. Inequality $Y(\mathscr{C}) \le Z^*$ holds because APP($\mathscr{C}$) is a relaxation of SSRCDP. The second inequality ($Z^* \le Z(\mathscr{R}(\mathscr{C}))$ holds because partition $\mathscr{R}(\mathscr{C})$ with optimized clusters' routing configurations is a feasible solution of SSRCDP. ∎

Proposition 3. *Let $\mathscr{R}(\mathscr{C})$ denote an optimal routing partition resulting from APP($\mathscr{C}$) and suppose that $\mathscr{R}(\mathscr{C})$ is a subset of $\mathcal{C}$. Then $Z(\mathscr{R}(\mathscr{C}))$ is an optimal solution of SSRCDP.*

Proof. Consider the vectors u, y defined for partition $\mathscr{R}(\mathscr{C})$ as in Proposition 1, where $x(\mathcal{R})$ is a routing configuration optimized for each routing cluster $\mathcal{R} \in \mathscr{R}(\mathscr{C})$ by means of RP($\mathcal{R}$). By Proposition 1, the solution u, y is feasible for APP($\mathcal{C}$). We will show that it is also optimal. Consider an arbitrary routing cluster $\mathcal{R} \in \mathscr{R}(\mathscr{C})$ and note that among the inequalities in (14g) that involve variables y^t, $t \in \mathcal{R}$, the one corresponding to $\mathcal{C} = \mathcal{R}$ is satisfied tightly since, by assumption, $\sum_{t\in\mathcal{R}} y^t = Z(\mathcal{R})$. Since for each $\mathcal{C}' \subset \mathcal{R}$ (whether or not $\mathcal{C}'$ is in $\mathscr{C}$), the inequality $\sum_{t\in\mathcal{C}'} y^t \ge Z(\mathcal{C}')$ holds (by Remark in Section 3.1), we conclude that vector y is optimal for APP($\mathscr{C}$), and hence $Y(\mathscr{C}) = \sum_{t\in\mathcal{T}} y^t = \sum_{\mathcal{R}\in\mathscr{R}} \sum_{t\in\mathcal{R}} y^t = \sum_{\mathcal{R}\in\mathscr{R}} Z(\mathcal{R})$. Thus, by (16), $Z(\mathscr{R}(\mathscr{C})) = Z^*$. ∎

4.2 | Cluster generation algorithm

The above properties justify the following algorithm for solving SSRCDP.

CGA: cluster generation algorithm

Step 0: Specify an initial control family $\mathscr{C}$.
Step 1: Solve APP($\mathscr{C}$) to obtain $\mathscr{R}(\mathscr{C})$ and $Y(\mathscr{C})$. Compute $Z(\mathscr{R}(\mathscr{C}))$ by solving RP($\mathcal{R}$) for each $\mathcal{R} \in \mathscr{R}(\mathscr{C})$.
Step 2: If $\mathscr{R}(\mathscr{C}) \subseteq \mathscr{C}$ or $\frac{Z(\mathscr{R}(\mathscr{C}))-Y(\mathscr{C})}{Y(\mathscr{C})} \le \varepsilon$ then stop: $\mathscr{R}(\mathcal{C})$ is a suboptimal (or even optimal) routing partition solving SSRCDP (where for each $\mathcal{R} \in \mathscr{R}$ its routing is optimized by RP($\mathcal{R}$)).
Step 3: $\mathscr{C} \leftarrow \mathscr{C} \cup \mathscr{R}(\mathscr{C})$ and go to Step 1.

If in Step 2 the condition $\mathscr{R}(\mathscr{C}) \subseteq \mathscr{C}$ is fulfilled then the routing partition $\mathscr{R}(\mathscr{C})$ delivered by CGA is optimal and $Z(\mathscr{R}(\mathscr{C}))$ is the optimal objective value. The same is true when $\frac{Z(\mathscr{R}(\mathscr{C}))-Y(\mathscr{C})}{Y(\mathscr{C})}$ equals 0. Clearly, the delivered family can be optimal even when $\mathscr{R}(\mathscr{C})\backslash\mathscr{C} \neq \emptyset$ and $\frac{Z(\mathscr{R}(\mathscr{C}))-Y(\mathscr{C})}{Y(\mathscr{C})} > 0$, as in this case the optimality will be proven in the next CGA iteration.

Finally, observe that because the number of all clusters is finite, CGA will stop (and then return an optimal partition $\mathscr{R}(\mathscr{C})$ for SSRCDP) in a finite number of steps, even if $\varepsilon = 0$ is assumed. This, however, may take excessive computation time.

4.3 | An efficient heuristic

In this section we describe a heuristic consisting in solving only one iteration of the CGA algorithm but using a modified version of APP($\mathscr{C}$). Consider a routing partition $\mathscr{R}$ defined by a binary vector $u = (u^t)_{t\in\mathcal{T}}$ feasible for APP($\mathscr{C}$), that is, fulfilling conditions (14b) and (14c).

Proposition 4. *Let $C = C(t_0, l)$ be a control cluster with $l \geq 2$ that has a nonempty intersection with exactly two (neighboring) clusters from $\mathcal{R}$ (i.e., $U^C = 1$). Let us also define the following quantity:*

$$Z(C|1) \coloneqq \min_{1 \leq k \leq l-1} \{Z(C(t_0, k)) + Z(C(t_0 \oplus k,\ l-k))\}. \tag{17}$$

Then the inequality

$$\sum_{t \in C} y^t \geq Z(C|1) \tag{18}$$

is valid.

Proof. Suppose that $C \subseteq \mathcal{R}' \cup \mathcal{R}'$, where $\mathcal{R}'$ and $\mathcal{R}''$ are two neighboring (and disjoint) clusters from family $\mathcal{R}$ specified by u. Then $C = C(t_0, k) \cup C(t_0 \oplus k, l-k)$ for some $1 \leq k \leq l-1$. Let $C' = C(t_0, k) \cap \mathcal{R}'$ and $C'' = C(t_0 \oplus k, l-k) \cap \mathcal{R}''$. By Proposition 4.1, $Z(C') \leq \sum_{t \in C'} z^t(x^*(\mathcal{R}'))$ and $Z(C'') \leq \sum_{t \in C''} z^t(x^*(\mathcal{R}''))$. Thus, $\sum_{t \in C'} z^t(x^*(\mathcal{R}')) + \sum_{t \in C''} z^t(x^*(\mathcal{R}'')) \geq Z(C') + Z(C'') \geq Z(C|1)$, which shows that (18) is a valid inequality. Note, that whenever $C' = \mathcal{R}'$ and $C'' = \mathcal{R}''$ in an optimal solution of APP($\mathscr{C}$), inequality (18) becomes tight. ∎

Clearly, for $U^C = 1$, inequality (18) is tighter than the inequality $Y^C \geq Z(C|\infty)$ implied by constraint (14g) (recall that $Y^C = \sum_{t \in C} y^t$), since in general $Z(C|1) > Z(C|\infty)$. Thus, substituting constraint (14g) in (14) with

$$Y^C \geq Z(C) + (Z(C|1) - Z(C)) \cdot U^C \qquad C \in \mathscr{C} \tag{19}$$

will result in a modified version of APP(C) (referred to as MAPP(C)) with a stronger linear relaxation than the original one.

Observe, however, that for $U^C \geq 2$, inequality (19) is in general not valid. For example, for $U^C = 2$, the value of $Z(C) + (Z(C|1) - Z(C)) \cdot 2$ can be greater than the proper value given by the following formula (analogous to (17)):

$$Z(C|2) \coloneqq \min_{1 \leq k_1 < k_2 \leq l-1,\ k_2 - k_1 \geq L} \{Z(C(t_0, k_1)) + Z(C(t_0 \oplus k_1,\ k_2 - k_1)) + Z(C(t_0 \oplus k_2,\ l - k_1 - k_2))\}. \tag{20}$$

It follows that MAPP(C) is correct only when the control family $\mathscr{C}$ is a subfamily of $\mathcal{C}(L+1)$—the family of all clusters of length at most $L+1$—since only then it is guaranteed that $U^C \leq 1$ for all $C \in \mathscr{C}$, making inequality (19) valid. Thus, the modified problem cannot be used in the CGA algorithm, as in general the routing partition $\mathcal{R}(\mathscr{C})$ contains clusters of length larger than $L+1$ and such sets cannot be added to the control family $\mathscr{C}$ when MAPP($\mathscr{C}$) is applied; therefore its use in CGA is limited to just one iteration. As we will see in Section 5, even this (noniterative) solution gives very good results when applied to SSRCDP.

4.4 | Improvements

The efficiency of the cluster generation algorithm described in Section 4.2, that is, CGA, can be improved in two complementary ways briefly described below (and fully in Appendix A).

First, the linear relaxation of formulation (14) can be strengthened (by improving, that is, increasing, the lower bound on the optimal objective function value delivered by its linear relaxation) in order to speed up the branch-and-bound algorithm (used to solve APP($\mathscr{C}$) in Step 1 of CGA) and also to decrease the gap $\frac{Z(\mathcal{R}(\mathscr{C})) - Y(\mathscr{C})}{Y(\mathscr{C})}$ between the integer solution and the relaxed solution. The lower bound computed through the linear relaxation of formulation (14) can be increased by improving valid inequalities specified in constraint (14g). In fact, these inequalities are tight only for the case of $U^C = 0$, that is, when the control cluster C is contained in a cluster of the constructed routing partition $\mathcal{R}$. (Recall that in this case the inequality in question takes the form $\sum_{t \in C} y^t \geq Z(C)$.) For $U^C \geq 1$ the inequalities implied by (14g) are weaker than the inequality in (14f), which, as already mentioned, implies that $\sum_{t \in C} y^t \geq Z(C|\infty)$, and this inequality is in general not tight.

A tight valid inequality generalizing (14g) can be obtained by constructing, for each $C \in \mathscr{C}$, a piece-wise linear function $G^C(U), 0 \leq U \leq M(C)$, where $M(C) \coloneqq \left\lceil \frac{l(C)-1}{L} \right\rceil$ is an upper bound on U^C, and for integer values of the argument U we put $G^C(U) = Z(C|U)$, where $Z(C|0) \coloneqq Z(C)$, $Z(C|1)$ is defined by (17), $Z(C|2)$ by (20), and $Z(C|U)$, $U \geq 3$, are defined analogously. Then, the valid inequality in (14g) should be replaced with the tight valid inequality $Y^C \geq G^C(U)$. (Such an inequality is not linear but can be transformed, using additional binary variables and linear constraints, to a form appropriate for a MIP formulation.) A detailed description of the above idea is given in Appendix A.1.

Second, apart from the clusters belonging to the routing partition $\mathcal{R}(\mathscr{C})$, which are added to the control family $\mathscr{C}$ in Step 3 of CGA, we may seek to add extra control clusters C', for which constraints (14g) are violated to the largest extent by the current optimal values y^*. This approach is described in detail in Appendix A.2.

TABLE 2 Performance of the solution procedure

Task	lb	ub	gap	t	n_{clusters}	n_{paths}
Static routing	—	563.65	—	5m7s	1	(+4461) 6623
Dynamic routing	545.47	—	3.33%	1m23s	96	(+89) 6712
Preprocessing	—	—	—	1h1m16s	768	(+1298) 8010
Partitioning LR	545.47	—	3.33%	1 second	(192)	—
Partitioning MIP	550.50	—	2.43%	2 seconds	(192)	—
Routing	—	551.86	0.25%	1m16s	8	(+1) 8011

5 | NUMERICAL EXPERIMENT

Below we describe a numerical experiment illustrating the efficiency of the proposed APP(C)-based approach for the realistic network example described in Section 2.3. Recall that the network consists of 47 nodes linked with 140 directed links (each of capacity 4 Gbps), and 2162 traffic demands corresponding to all ordered node pairs. The demand volumes used in the calculations are derived from real traffic measurements taken every 15 minutes on a selected weekday. Thus, the number of the considered timepoints T equals 96. We set the maximal number of clusters to $N = 8$ and the minimum cluster length to $L = 8$, that is, we assume the network can be reconfigured at most 8 times per day (on average every 3 hours), and the next network network reconfiguration can be applied only 2 hours after the previous one.

In the experiment reported below, for solving the semi-stable routing cluster design problem SSRCDP we used formulation MAPP(C) and one iteration of CGA in the way described in Section 4.3. The procedure was implemented and executed using the following hardware and software platform: Lenovo Thinkpad, Intel i7-6500U 3.10 GHz, 8 GB RAM, Windows 10 x64, ILOG CPLEX Studio 12.8, ILOG Concert library, C# language, CPLEX 12.8 solver, 2 threads.

For the control family $\mathscr{C}$ we used all the clusters of length L and $L+1$. There are $2T = 192$ such clusters, and thus, in the preprocessing phase, for each of them we needed to calculate the values $Z(C) = Z$ and $Z(C|1)$ according to formulae (5a) and (17), respectively. For that, the routing problem RP($\mathcal{U}$) formulated in (5) was solved $8T = 768$ times, that is, for all clusters having length between 2 and 9.

In RP($\mathcal{U}$), we used a delay function $F(w) := \max\{0.1w, w - 0.45, 10w - 8.5\}$ (with $K = 3$ linear pieces), which implies $b(1) = 0$, $b(2) = -0.45$, $b(3) = -8.5$ and $a(1) = 0.1$, $a(2) = 1$, $a(3) = 10$. Thus, $F(w)$ grows from 0 to 0.05 in the interval [0,0.5], from 0.05 to 0.5 in the interval [0.5,0.9], and from 0.5 to $+\infty$ in the interval $[0.9, +\infty]$.

The results of our experiment are presented in Table 2. For each task of the solution procedure, the corresponding row of the table first gives the determined lower (column *LB*) and upper (column *UB*) bound for the optimal objective function value, and the current relative gap between the two (column *GAP*). Next, column t shows the total execution time of the task. Then, column $N_{CLUSTERS}$ gives the number of clusters that we analyze within the task, that is, clusters for which we solve the routing problem, and in brackets, if applicable, the number of clusters that are contained in the control family of the partitioning problem. Finally, column N_{PATHS} first shows (in brackets, with the plus sign) the total number of paths generated while solving routing problems in the task, and (not in brackets) the final size of the set of paths $\mathcal{P}$ obtained in the routing problem (note that not necessarily all those paths are used in the final solution).

The row *STATIC ROUTING* summarizes the case when only one routing cluster, that is, $\mathcal{T}$, is applied. For the optimized single routing scheme the optimal value of the objective function equal to $Z(\mathcal{T})$ is given in column *UB*, as this value is the upper bound for the true SSRCDP optimal solution objective function value. The row *DYNAMIC ROUTING* corresponds to the case when each timepoint is considered as a cluster, that is, the routing scheme is optimized individually for each timepoint. The optimal objective function value $\sum_{t\in\mathcal{T}} Z(\{t\})$ is given in column *LB*, since it is clearly the cheapest solution value for SSRCDP (it is the case when the partition into routing clusters is not constrained with L or N, and thus each timepoint can be a separate cluster having, potentially, but not necessarily, an individual routing configuration). The value in column *GAP* is equal to $\frac{UB\text{-}LB}{LB} \times 100\%$ (*UB* taken from *STATIC ROUTING* and *LB* taken from *DYNAMIC ROUTING*). The row *PREPROCESSING* contains information concerning preparation of the control family $\mathscr{C}$ and computation of initial routing paths (recall that RP($\mathcal{U}$) is solved through path generation). Next, the row *PARTITIONING LR* shows the results of solving the linear relaxation of the problem using the modified APP(C) formulation, that is, of problem MAPP($\mathscr{C}$) described in Section 4.3. The so obtained value of *LB* happens to be the same as for *dynamic routing*, although in general it could be higher. Furthermore, the solution of the MIP problem MAPP($\mathscr{C}$) is described in the row *PARTITIONING MIP*. The lower bound column delivered by this solution, contained in column *LB*, is increased with respect to the preceding row and hence the value in column *GAP* is decreased. Finally, the row *ROUTING* shows the results of optimizing the routing scheme for each of the routing clusters $\mathcal{R}$ of the partition $\mathscr{R}(\mathscr{C})$ obtained in the previous step of solving the MAPP($\mathscr{C}$). In particular, *UB* gives the value of $Z(\mathscr{R}(\mathscr{C}))$: observe that the

gap between this feasible SSRCDP solution value and the best lower bound obtained with *PARTITIONING MIP* is very small and equals 0.25%. We note, that in the final solution the optimal routing partition $\mathscr{R}(\mathscr{C})$ is composed of five 8-element, one 13-element, one 15-element, and one 28-element clusters.

The results indicate that already the simplified version of the proposed method, without any special tuning, is capable of finding a suboptimal solution of SSRCDP in a reasonable time within the optimality gap as small as 0.25%.

6 | CONCLUSIONS

In this paper we propose a scalable solution to the problem of designing clusters for semi-stable routing in multicommodity flow networks. Although the problem can be approached directly using a compact mixed-integer formulation, it cannot be just solved with a solver, even for small-size networks, due to an excessive number of binary variables and a poor linear relaxation. Thus we considered a number of exact and hybrid approaches (as in [16]) trying to separate the design of a partition of the time horizon into clusters from the design of traffic routing for those clusters.

Although there are just $O(T^2)$ clusters with length between 1 and T (where, for a single day, T is typically between 96 and 288 as the traffic measurement period is either 15 or 5 minutes), our numerical trials show that in practice we can analyze only a small fraction of those clusters. Using a link-path formulation combined with path generation and a warm start for the master problem, on average it takes around l seconds to solve the routing problem for a cluster of length l for a 50-node network. And that time might grow considerably as we aim at networks whose number of nodes approaches 500.

Therefore, leveraging the valid inequalities of an approximate time-horizon partitioning problem, we developed an efficient heuristic algorithm based on using a family of preprocessed clusters to control the partitioning process. Our algorithm is capable of providing upper and lower bounds on the objective function value with very low optimality gaps, well below 0.5% (as shown in the presented numerical study, and in some other studies not reported here). It also offers the trade-off between the quality of the solution and the number of clusters in the control family of the partitioning problem: the larger control family potentially provides a better solution but leads to the increase of the preprocessing time, and the size and the solution time of the partitioning problem.

In addition, in the appendix we have proposed two possible ways of improving the efficiency of our approach that lead to interesting future research. First, we can use stronger formulations of APP(C), equipped with improvements described in Appendix A.1. Second, we can either implement a full version of the cluster generation algorithm presented in Appendix A.2.1, or, even better, incorporate cluster generation into the branch-and-bound procedure of solving the partitioning problem, by analyzing relaxed or incumbent solutions and generating appropriate user cuts. All these (nontrivial) extensions are currently being developed and, after implementation, will be tested on larger network examples. In parallel we will use traffic data with lower correlation among the traffic matrices, as this might result in the gap between the static and dynamic routing solutions that is more substantial than the 3.33% observed in the current example (which our current algorithm nonetheless managed to decrease tenfold).

ACKNOWLEDGMENT

The Polish authors were supported by the National Science Centre, Poland, grant number 2017/25/B/ST7/02313: "Packet routing and transmission scheduling optimization in multi-hop wireless networks with multicast traffic."

ORCID

Michał Pióro https://orcid.org/0000-0002-9347-9764

REFERENCES

[1] Y. Al Najjar, S. Paris, J. Elias, J. Leguay, and W. Ben-Ameur, Optimal routing configuration clustering through dynamic programming, Proc. AlgoTel (2019).
[2] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Räcke, *Optimal oblivious routing in polynomial time*, J. Comp. Syst. Sci. **69** (2004), 383–394.
[3] W. Ben-Ameur and M. Żotkiewicz, *Robust routing and optimal partitioning of a traffic demand polytope*, Intl. Trans. Oper. Res. **18** (2011), 307–333.
[4] W. Ben-Ameur and M. Żotkiewicz, *Multipolar routing: Where dynamic and static routing meet*, Electron. Notes Discr. Math. **41** (2013), 61–68.
[5] T. Benson, A. Anand, A. Akella, and M. Zhang, *MicroTE: Fine grained traffic engineering for data centers*, Proc. ACM CoNext (2011), 1–12. https://doi.org/10.1145/2079296.2079304.
[6] P. Casas, L. Fillatre, and S. Vaton, *Multi hour robust routing and fast load change detection for traffic engineering*, Proc. IEEE ICC. Beijing: IEEE International Conference on Communications; (2008), 5777–5782.
[7] B. Fortz and M. Thorup, *Optimizing OSPF/IS-IS weights in a changing world*, IEEE JSAC **20** (2002), 756–767.
[8] C.Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, *Achieving high utilization with software-driven WAN*, ACM SIGCOMM CCR **43** (2013), 15–26.

[9] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, *B4: Experience with a globally-deployed software defined WAN*, ACM SIGCOMM CCR **43** (2013), 3–14.
[10] M. Kodialam, T. Lakshman, and S. Sengupta, *Efficient and robust routing of highly variable traffic*, Proc. ACM HotNets (2004).
[11] M. Minoux, *Mathematical Programming: Theory and Algorithms*, John Wiley & Sons, New York, 1986.
[12] S. Orlowski and M. Pióro, *Complexity of column generation in network design with path-based survivability mechanisms*, Networks **59** (2012), 132–147.
[13] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*, Morgan-Kaufmann, Burlington, MA, 2004.
[14] M. Poss and C. Raack, *Affine recourse for the robust network design problem: Between static and dynamic routing*, Networks **61** (2013), 180–198.
[15] M. Roughan, M. Thorup, and Y. Zhang, Traffic engineering with estimated traffic matrices, Proc. ACM IMC. (2003), 248–258.
[16] D. Sanvito, I. Filippini, A. Capone, S. Paris, and J. Leguay, *Adaptive robust traffic engineering in software defined networks*, Proc. IFIP Netw. Zurich, Switzerland: IFIP Networking Conference (IFIP Networking) and Workshops; (2018), 145–153. https://doi.org/10.23919/IFIPNetworking.2018.8696406.
[17] M. Silva, M. Poss, and N. Maculan, *Solving the bifurcated and nonbifurcated robust network loading problem with k-adaptive routing*, Networks **72** (2018), 151–170.
[18] V. Tabatabaee, A. Kashyap, B. Bhattacharjee, R.J. La, and M.A. Shayman, *Robust routing with unknown traffic matrices*, Proc. IEEE INFOCOM (2007), 2436–2440. https://doi.org/10.1109/INFCOM.2007.296.
[19] A. Tomaszewski, M. Pióro, D. Sanvito, I. Filippini, and A. Capone, *On optimization of semi-stable routing in multicommodity flow networks*, Proc. INOC (2019), 54–59.
[20] H. Wang, H. Xie, L. Qiu, Y.R. Yang, Y. Zhang, and A. Greenberg, *COPE: Traffic engineering in dynamic networks*, ACM SIGCOMM CCR **36** (2006), 99–110.
[21] Y. Zhang and Z. Ge, *Finding critical traffic matrices*, 2005 International Conference on Dependable Systems and Networks (DSN'05), Yokohama, Japan, (2005), 188–197, doi: 10.1109/DSN.2005.51.

How to cite this article: Tomaszewski A, Pióro M, Sanvito D, Filippini I, Capone A. An efficient approach to optimization of semi-stable routing in multicommodity flow networks. *Networks.* 2021;77:538–558. https://doi.org/10.1002/net.21999

APPENDIX A: IMPROVING THE APPROXIMATION APPROACH

The efficiency of the cluster generation algorithm (CGA) described in Section 4.2 can be improved in two complementary ways described below.

A.1 | Stronger formulations of the approximation problem

In this section we will discuss the issue of strengthening inequalities (14g) in formulation APP($\mathscr{C}$) for a given control cluster C. Recall that these inequalities are crucial to achieve a tight lower bound on the optimal objective function of SSRCDP (and of APP($\mathscr{C}$)) through the linear relaxation of APP($\mathscr{C}$).

A.2 | Piece-wise linear nonconvex lower bound

Let us first assume that the considered cluster C is partitioned into a family $\mathcal{S}$ of m (where $m \geq 1$) subclusters, each of length greater than or equal to L. Now let

$$\hat{Z}(C|m) \coloneqq \min_{\mathcal{S}} \sum_{S \in \mathcal{S}} Z(S), \tag{A1}$$

where the minimum is taken over all such partitions $\mathcal{S}$. Assuming that $C = C(t,l)$, where $l \geq m \cdot L$, the value of $\hat{Z}(C|m)$ can be calculated recursively as follows:

$$\hat{Z}(C(t,l)|1) = Z(C(t,l)) \tag{A2a}$$

$$\hat{Z}(C(t,l)|m) = \min_{L \leq k \leq l-(m-1)L} \{Z(C(t,k)) + \hat{Z}(C(t \oplus k, l-k|m-1)\} \qquad \left(2 \leq m \leq \frac{l}{L}\right). \tag{A2b}$$

Next consider partitions $\mathcal{S}$ of cluster $C = C(t,l)$ into m nonempty subclusters, where all subclusters, apart from the first and the last one, are of length at least L. Note that in this case m is between 1 (in this case C is not partitioned) and $M(C) \coloneqq 1 + \left\lceil \frac{l-1}{L} \right\rceil$.) Let $Z(C|m)$ denote the quantity analogous to (A1) but with the minimum taken over all such modified partitions $\mathcal{S}$. In this case the following recursive formula applies:

$$Z(C(t,l)|1) = Z(C) \tag{A3a}$$

$$Z(C(t,l)|m) = \min_{k_1 \geq 1, k_2 \geq 1, k_1+k_2 \leq l-(m-2)L} \{Z(C(t,k_1)) + Z(C(t \oplus (l-k_2), k_2)) +$$

$$+ \widehat{Z}(C(t \oplus k_1, l - (k_1 + k_2))|m-2)\} \qquad (2 \le m \le M(C)), \tag{A3b}$$

where $\widehat{Z}(C(t,l)|0) \coloneqq 0$. Note that $Z(C|1) \ge Z(C|m), m = 1, 2, \dots, M(C)$.

Now recall that the number $U^C + 1$ denotes with how many clusters in the family $\mathcal{R}$ specified by variables u a given cluster $C \in \mathcal{C}$ intersects, and observe that the constraint

$$\sum_{t \in C} y^t \ge Z(C|U^C + 1) \qquad C \in \mathcal{C} \tag{A4}$$

represents valid inequalities that can be used (for each $C \in \mathcal{C}$) in APP($\mathcal{C}$) instead of (14g). This substitution will in general strengthen formulation (14), because $Z(C|m) \ge Z(C|\infty)$ for all $m = 1, 2, \dots, M(C)$, and the inequalities are in general sharp. Yet, the new inequalities cannot be simply added to (14) in the above form because this would lead to a formulation that is not a MIP formulation (contrary to (14)) anymore. However, a MIP formulation in question can be achieved by introducing, for each $C \in \mathcal{C}$, the following piece-wise linear function $F^C(U)$ with the domain $[0, M(C) - 1]$:

$$F^C(U) = \begin{cases} (Z(C|2) - Z(C|1)) \cdot U + Z(C|1), & U \in [0,1) \\ (Z(C|3) - Z(C|2)) \cdot U + Z(C|2), & U \in [1,2) \\ \dots \\ (Z(C|M(C)) - Z(C|M(C)-1)) \cdot U + Z(C|M(C)-1), & U \in [M(C)-2, M(C)-1]. \end{cases}$$

Note that, as required, $F^C(U) = Z(C|U+1)$ for $U = 0, 1, \dots, M(C) - 1$.

Having defined the piece-wise linear functions $F^C(U), C \in \mathcal{C}$, we can now reformulate APP($\mathcal{C}$) as follows:

$$\text{APP}(\mathcal{C}) : Y(\mathcal{C}) = \min \sum_{t \in \mathcal{T}} y^t \tag{A6a}$$

$$\sum_{t \in \mathcal{T}} u^t \le N \tag{A6b}$$

$$\sum_{0 \le k \le L-1} u^{t \oplus k} \le 1 \qquad t \in \mathcal{T} \tag{A6c}$$

$$U^C = \sum_{1 \le k < l(C)} u^{t(C) \oplus k} \qquad C \in \mathcal{C} \tag{A6d}$$

$$Y^C = \sum_{t \in C} y^t \qquad C \in \mathcal{C} \tag{A6e}$$

$$y^t \ge Z(\{t\}) \qquad t \in \mathcal{T} \tag{A6f}$$

$$Y^C \ge F^C(U^C) \qquad C \in \mathcal{C} \tag{A6g}$$

$$u^t \in \mathbb{B} \qquad t \in \mathcal{T} \tag{A6h}$$

$$U^C \in \mathbb{Z}^+ \qquad C \in \mathcal{C} \tag{A6i}$$

$$y^t \in \mathbb{R}^+ \qquad t \in \mathcal{T} \tag{A6j}$$

$$Y^C \in \mathbb{R}^+ \qquad C \in \mathcal{C}. \tag{A6k}$$

The above formulation still is not a MIP because constraint (A6g) is not linear. A proper MIP formulation for APP($\mathcal{C}$) can be achieved in several ways, each of them requiring around $M(\mathcal{C})$ additional binary variables (and around $M(\mathcal{C})$ continuous variables). We will apply the so called incremental way of dealing with piece-wise linear functions. It works as follows.

Consider a cluster $C \in \mathcal{C}$ and denote the slopes of the consecutive linear pieces of $F^C(U)$ defined by (A5) as follows:

$$A(C|m) \coloneqq Z(C|m+1) - Z(C|m) \qquad m = 1, 2, \dots, M(C) - 1. \tag{A7}$$

Next, let us introduce a vector of continuous variables $x^{\mathcal{C}} = (x^{\mathcal{C}}_m)_{1\le m\le M(C)-1}$ and a vector of binary variables $s^{\mathcal{C}} = (s^{\mathcal{C}}_m)_{2\le m\le M(C)-1}$. Then, $F^{\mathcal{C}}(U)$ (for a given U $(0 \le U \le M(C)-1)$ is determined by the following system of equations and inequalities:

$$F^{\mathcal{C}}(U) = Z(C|1) + A(C|1)x^{\mathcal{C}}_1 + A(C|2)x^{\mathcal{C}}_2 + \cdots + A(C|M(C)-1)x^{\mathcal{C}}_{M(C)-1} \tag{A8a}$$

$$U = x^{\mathcal{C}}_1 + x^{\mathcal{C}}_2 + \cdots + x^{\mathcal{C}}_{M(C)-1} \tag{A8b}$$

$$s^{\mathcal{C}}_2 \le x^{\mathcal{C}}_1 \le 1 \tag{A8c}$$

$$s^{\mathcal{C}}_3 \le x^{\mathcal{C}}_2 \le s^{\mathcal{C}}_2$$
$$\cdots \tag{A8d}$$

$$s^{\mathcal{C}}_{M(C)-1} \le x^{\mathcal{C}}_{M(C)-2} \le s^{\mathcal{C}}_{M(C)-2} \tag{A8e}$$

$$0 \le x^{\mathcal{C}}_{M(C)-1} \le s^{\mathcal{C}}_{M(C)-1}. \tag{A8f}$$

The so calculated value $F^{\mathcal{C}}(U)$ is correct because constraints (A8c)-(A8f) imply that if $x^{\mathcal{C}}_m > 0$ then $x^{\mathcal{C}}_k = 1$ for all $k = 1, 2, \ldots, m-1$. Therefore, if $U = m + z$ (for some $z \in (0, 1]$) then

$$x^{\mathcal{C}}_1 = x^{\mathcal{C}}_2 = \cdots = x^{\mathcal{C}}_m = 1,\ \ x^{\mathcal{C}}_{m+1} = z,\ \ x^{\mathcal{C}}_{m+2} = \cdots = x^{\mathcal{C}}_{M(C)-1} = 0.$$

Hence, by (A8a) and (A7),

$$F^{\mathcal{C}}(U) = Z(C|1) + (Z(C|2) - Z(C|1)) + \cdots + (Z(C|m+1) - Z(C|m)) + (Z(C|m+1) - Z(C|m))z,$$

that is,

$$F^{\mathcal{C}}(U) = Z(C|m) + (Z(C|m+1) - Z(C|m))z$$

as required. The so described method for calculating $F^{\mathcal{C}}(U)$ leads to the following strengthened MIP formulation of APP(C).

$$\text{APP}(C): \quad Y(C) = \min \sum_{t\in\mathcal{T}} y^t \tag{A9a}$$

$$\sum_{t\in\mathcal{T}} u^t \le N \tag{A9b}$$

$$\sum_{0\le k\le L-1} u^{t\oplus k} \le 1 \qquad t \in \mathcal{T} \tag{A9c}$$

$$U^{\mathcal{C}} = \sum_{1\le k<l(C)} u^{t(C)\oplus k} \qquad \mathcal{C} \in \mathcal{C} \tag{A9d}$$

$$Y^{\mathcal{C}} = \sum_{t\in C} y^t \qquad \mathcal{C} \in \mathcal{C} \tag{A9e}$$

$$y^t \ge Z(\{t\}) \qquad t \in \mathcal{T} \tag{A9f}$$

$$Y^{\mathcal{C}} \ge Z(\mathcal{C}|1) + A(C|1)x^{\mathcal{C}}_1 + A(C|2)x^{\mathcal{C}}_2 + \cdots + A(C|M(C)-1)x^{\mathcal{C}}_{M(C)-1} \quad \mathcal{C} \in \mathcal{C} \tag{A9g}$$

$$U^{\mathcal{C}} = x^{\mathcal{C}}_1 + x^{\mathcal{C}}_2 + \cdots + x^{\mathcal{C}}_{M(C)-1} \qquad \mathcal{C} \in \mathcal{C} \tag{A9h}$$

$$s^{\mathcal{C}}_2 \le x^{\mathcal{C}}_1 \le 1 \qquad \mathcal{C} \in \mathcal{C} \tag{A9i}$$

$$s^{\mathcal{C}}_3 \le x^{\mathcal{C}}_2 \le s^{\mathcal{C}}_2 \qquad \mathcal{C} \in \mathcal{C}$$
$$\cdots \tag{A9j}$$

$$s^{\mathcal{C}}_{M(C)-1} \le x^{\mathcal{C}}_{M(C)-2} \le s^{\mathcal{C}}_{M(C)-2} \qquad \mathcal{C} \in \mathcal{C} \tag{A9k}$$

$$0 \le x^{\mathcal{C}}_{M(C)-1} \le s^{\mathcal{C}}_{M(C)-1} \qquad \mathcal{C} \in \mathcal{C} \tag{A9l}$$

$$u^t \in \mathbb{B} \qquad t \in \mathcal{T} \tag{A9m}$$

$$U^{\mathcal{C}} \in \mathbb{Z}^+ \qquad \mathcal{C} \in \mathcal{C} \tag{A9n}$$

$$y^t \in \mathbb{R}^+ \qquad t \in \mathcal{T} \tag{A9o}$$

$$Y^{\mathcal{C}} \in \mathbb{R}^+ \qquad \mathcal{C} \in \mathcal{C} \tag{A9p}$$

$$x^{\mathcal{C}} \in (\mathbb{R}^+)^{M(C)-1} \qquad \mathcal{C} \in \mathcal{C} \tag{A9q}$$

$$s^{\mathcal{C}} \in \mathbb{B}^{M(C)-2} \qquad \mathcal{C} \in \mathcal{C}. \tag{A9r}$$

Observe that if for some control cluster $\mathcal{C} \in \mathcal{C}$ it happens that $F^{\mathcal{C}}$ is convex (which can be easily checked when writing down formulation (A9)), then constraints (A9g)-(A9l) can be substituted with

$$Y^{\mathcal{C}} \ge (Z(\mathcal{C}|m+1) - Z(\mathcal{C}|m)) \cdot U^{\mathcal{C}} + Z(\mathcal{C}|m) \qquad m = 1, 2, \ldots, M(C) \tag{A10}$$

and this does not involve extra variables $x^{\mathcal{C}}$ and $s^{\mathcal{C}}$. In particular, when this is the case for all control clusters, the computational complexity of the strengthened formulation is similar to that of formulation (14).

A.3 | Lower convex envelope bound

Since in general the use of the piece-wise linear functions of $F^{\mathcal{C}}$ in (A9) can be unacceptably time consuming because of the excessive number of the binary variables they introduce, it seems reasonable to use the *lower convex envelopes* of functions $F^{\mathcal{C}}$. Recall that the lower convex envelope $\breve{f}$ of a function f defined on an interval $[a, b]$ is defined at each point of the interval as the supremum of all convex functions that lie under that function, that is,

$$\breve{f}(x) = \sup\{g(x) : g \quad \text{is convex and} \quad g \le f \quad \text{over} \quad [a, b]\}.$$

The value of the lower convex envelope of $\breve{F}^{\mathcal{C}}$ of $F^{\mathcal{C}}$ at a given point $U \in [0, M(C) - 1]$ can be (efficiently) computed as the minimum of the following LP formulation.

$$\sum_{m \in M(C)} \alpha^{\mathcal{C}}_m = 1 \tag{A11a}$$

$$\alpha^{\mathcal{C}}_m \in \mathbb{R}^+ \qquad m \in \mathcal{M}(\mathcal{C}), \tag{A11b}$$

where $\mathcal{M}(\mathcal{C}) \coloneqq \{0, 1, \ldots, M(C) - 1\}$.

Hence, applying the lower convex envelope approximation of $F^{\mathcal{C}}$ in APP($\mathcal{C}$) leads to the following MIP formulation.

$$\text{APP}(\mathcal{C}): \quad Y(C) = \min \sum_{t \in \mathcal{T}} y^t \tag{A12a}$$

$$\sum_{t \in \mathcal{T}} u^t \le N \tag{A12b}$$

$$\sum_{0 \le k \le L-1} u^{t \oplus k} \le 1 \qquad t \in \mathcal{T} \tag{A12c}$$

$$U^{\mathcal{C}} = \sum_{1 \le k < l(C)} u^{t(C) \oplus k} \qquad \mathcal{C} \in \mathcal{C} \tag{A12d}$$

$$Y^{\mathcal{C}} = \sum_{t \in C} y^t \qquad \mathcal{C} \in \mathcal{C} \tag{A12e}$$

$$y^t \geq Z(\{t\}) \qquad t \in \mathcal{T} \tag{A12f}$$

$$Y^{\mathcal{C}} \geq \sum_{m \in \mathcal{M}(C)} \alpha_m^C \; Z(C|m+1) \qquad \mathcal{C} \in \mathcal{C} \tag{A12g}$$

$$U^{\mathcal{C}} = \sum_{m \in \mathcal{M}(C)} \alpha_m^C \; m \qquad \mathcal{C} \in \mathcal{C} \tag{A12h}$$

$$\sum_{m \in \mathcal{M}(C)} \alpha_m^C = 1 \qquad \mathcal{C} \in \mathcal{C} \tag{A12i}$$

$$u^t \in \mathbb{B} \qquad t \in \mathcal{T} \tag{A12j}$$

$$U^{\mathcal{C}} \in \mathbb{Z}^+ \qquad \mathcal{C} \in \mathcal{C} \tag{A12k}$$

$$y^t \in \mathbb{R}^+ \qquad t \in \mathcal{T} \tag{A12l}$$

$$Y^{\mathcal{C}} \in \mathbb{R}^+ \qquad \mathcal{C} \in \mathcal{C} \tag{A12m}$$

$$\alpha_m^{\mathcal{C}} \in \mathbb{R}^+ \qquad m \in \mathcal{M}(C). \tag{A12n}$$

A.4 | Comments

Clearly, both strengthened versions of APP($\mathcal{C}$) formulated in (A9) and (A12) above can be used instead of (14) in the CGA algorithm presented in Section 4.1. In general, we may expect that formulation (A9) (and, perhaps to a less extent, formulation (A12)) will decrease the gap $\frac{Z(\mathcal{R}(C))-Y(C)}{Y(C)}$ faster than formulation (14). On the other hand, the computational time of formulation (A9) may soon become excessive when the size of the control family $\mathcal{C}$ grows, due to a large number of additional binary variables and the necessity of computing the $Z(C|m)$ values. The latter issue is, however, less significant for (A12).

A.5 | Enhanced cluster generation algorithm

Below we present an extension of the cluster generation algorithm CGA described in Section 4.2; the extension applies a more efficient cluster generation method.

A.6 | ECGA formulation

Let us consider the *linear relaxation* of APP($\mathcal{C}$) for a given control family $\mathcal{C}$ and denote such a relaxation by APP/LR($\mathcal{C}$). Let and u^* and y^* be the optimal solution of APP/LR($\mathcal{C}$) and consider the following *pricing problem*: find cluster C' that maximizes the right-hand side of constraint (14g) (or (A9g) or (A12g), depending on the version of APP($\mathcal{C}$) used), where $U^{C'} = \sum_{1 \leq k < l(C')} u^{t(C') \oplus k *}$; the so described pricing problem will be denoted by PP(u^*, y^*).

Note that if the so maximized value is strictly greater than $Y^{C'} := \sum_{t \in C'} y^{t*}$, then the solution u^*, y^* will become infeasible for APP($\mathcal{C}$) when C' becomes a control cluster, that is, when it is added to the control family $\mathcal{C}$. Hence, we can expect that adding C' to the control family will result in the minimum of the objective function of APP/LR($\mathcal{C} \cup \{C'\}$) being greater than that of APP/LR($\mathcal{C}$). The pricing problem PP(u^*, y^*) will be formulated in the next section.

ECGA: enhanced cluster generation algorithm

Step 0: Specify an initial control family $\mathcal{C}$.
Step 1: Solve APP/LR($\mathcal{C}$). Let and u^* and y^* be the resulting optimal solution.
Step 2: Solve PP(u^*, y^*). Let C' be the resulting optimal solution.
Step 3: If for C' the objective function of PP(u^*, y^*) is strictly greater than $Y^{C'}$, then $\mathcal{C} \leftarrow \mathcal{C} \cup \{C'\}$, that is, add the generated cluster to the control family, and go to Step 1.
Step 4: Solve APP($\mathcal{C}$). If $\mathcal{R}(\mathcal{C}) \subseteq \mathcal{C}$ or $\frac{Z(\mathcal{R}(C))-Y(C)}{Y(C)} \leq \varepsilon$ then stop: $\mathcal{R}(\mathcal{C})$ is a suboptimal (or even optimal) routing partition solving SSRCDP (where for each $\mathcal{R} \in \mathcal{R}$ its routing is optimized by RP($\mathcal{R}$)).
Step 5: $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{R}(\mathcal{C})$ and go to Step 1.

Note that the remarks made just after formulation of CGA in Section 4.2 are valid for ECGA as well. The rationale behind ECGA is that it potentially improves the lower bound for SSRCDP more quickly than CGA and with fewer control clusters added during the optimization process.

A.7 | Pricing problem

The pricing problem formulated below is applicable to formulation (14).

$$\mathrm{PP}(u^*, y^*): \quad B(u^*, y^*) = \max\{Z + (Z' - Z) \cdot U - Y\} \tag{A13a}$$

$$\sum_{t \in \mathcal{T}} a^t = 1 \tag{A13b}$$

$$\sum_{t \in \mathcal{T}} b^t = 1 \tag{A13c}$$

$$a^t + b^t \leq 1 \qquad t \in \mathcal{T} \tag{A13d}$$

$$c^t = c^{t \ominus 1} + a^t - b^t \qquad t \in \mathcal{T} \tag{A13e}$$

$$U = \sum_{t \in \mathcal{T}} u^{t*} \ c^t \tag{A13f}$$

$$Y = \sum_{t \in \mathcal{T}} y^{t*} \ c^t \tag{A13g}$$

$$Z' = \sum_{t \in \mathcal{T}} Z(\{t\}) \ c^t \tag{A13h}$$

$$Z = \sum_{d \in \mathcal{D}} \lambda_d \tag{A13i}$$

$$\sum_{e \in \mathcal{E}} \pi_e^t = c^t \qquad t \in \mathcal{T} \tag{A13j}$$

$$\lambda_d \leq \sum e \in \mathcal{E}(d,p) \ \frac{1}{c(e)} \ \sum_{t \in \mathcal{T}} h(d,t) \ \pi_e^t \qquad d \in \mathcal{D}, p \in \mathcal{P}(d) \tag{A13k}$$

$$a^t, b^t, c^t \in \mathbb{B} \qquad t \in \mathcal{T} \tag{A13l}$$

$$U \in \mathbb{Z}^+ \tag{A13m}$$

$$Y, Z, Z' \in \mathbb{R}^+ \tag{A13n}$$

$$\lambda_d \in \mathbb{R} \qquad d \in \mathcal{D} \tag{A13o}$$

$$\pi_e^t \in \mathbb{R}^+ \qquad t \in \mathcal{T}, e \in \mathcal{E}. \tag{A13p}$$

In the above MIP formulation, binary variables $a = (a^t)_{t \in \mathcal{T}}, b = (b^t)_{t \in \mathcal{T}}, c = (c^t)_{t \in \mathcal{T}}$ determine the control cluster C' we are looking for. The particular variable $a^t = 1$ (unique due to equality (A13b)) determines the epoch t where C' starts, while $b^t = 1$ (unique due to (A13c)) means that C' ends at epoch $t - 1$. Additionally, constraint (A13d) assures that the cluster does not end in the epoch just before the starting epoch (this eliminates the trivial cluster $\mathcal{T}$). In turn, variables c determine the epochs comprising the constructed control set, that is, $C' := \{t \in \mathcal{T} : \ c^t = 1\}$. The proper values of c are forced by equalities (A13e).

The next set of equalities, (A13g)-(A13h), define the quantities $U^{C'}, Y^{C'}, Z(C'|\infty)$, respectively (cf. (14d), (14e), and the definition of $Z(\mathcal{C}|\infty)$ just before formulation (14) in Section 4.1). Determination of the value $Z = Z(C')$ requires more effort.

This value is, by definition, the optimal objective of the following minimization problem, derived from the routing problem RP($\mathcal{U}$) formulated in (5).

$$\text{RP}(c): \quad Z = \min \quad \sum_{t\in\mathcal{T}} c^t \; z^t \tag{A14a}$$

$$\sum_{p\in\mathcal{P}(d)} x_{dp} = 1 \qquad [\lambda_d] \quad d \in \mathcal{D} \tag{A14b}$$

$$z^t \geq \frac{1}{c(e)} \sum_{p\in\mathcal{Q}(e,d)} h(d,t)\; x_{dp} \qquad [\pi_e^t \geq 0] \quad t \in \mathcal{T}, e \in \mathcal{E} \tag{A14c}$$

$$x_{dp} \in \mathbb{R}^+ \qquad d \in \mathcal{D}, p \in \mathcal{P}(d) \tag{A14d}$$

$$z^t \in \mathbb{R} \qquad t \in \mathcal{T}. \tag{A14e}$$

Since (A14) is a minimization problem it cannot be embedded into formulation of PP(u^*, y^*) which involves maximization of Z (see (A13a)). Thus, we form the dual of (A14) (using dual variables specified in the square brackets on the right-hand sides of (A14b) and (A14c)):

$$\text{DRP}(c): \quad Z = \max \quad \sum_{d\in\mathcal{D}} \lambda_d \tag{A15a}$$

$$\sum_{e\in\mathcal{E}} \pi_e^t = c^t \qquad t \in \mathcal{T} \tag{A15b}$$

$$\lambda_d \leq \sum_{e\in\mathcal{E}(d,p)} \frac{1}{c(e)} \sum_{t\in\mathcal{T}} h(d,t)\; \pi_e^t \qquad d \in \mathcal{D}, p \in \mathcal{P}(d) \tag{A15c}$$

$$\lambda_d \in \mathbb{R} \qquad d \in \mathcal{D} \tag{A15d}$$

$$\pi_e^t \in \mathbb{R}^+ \qquad t \in \mathcal{T}, e \in \mathcal{E} \tag{A15e}$$

(where $\mathcal{E}(d,p)$ denotes the set of links traversed by path $p \in \mathcal{P}(d)$) and embed it in the pricing problem by means of constraints (A13i)-(A13k) and (A13o)-(A13p).

Observe that the term $(Z' - Z) \cdot U$ in the objective function (A13a) of PP(u^*, y^*) contains bi-linearities since $(Z' - Z) \cdot U$ is actually equal to:

$$\left(\sum_{t\in\mathcal{T}} Z(\{t\})\; c^t - \sum_{d\in\mathcal{D}} \lambda_d \right) \cdot \sum_{t\in\mathcal{T}} u^{t*}\; c^t = \sum_{t,t'\in\mathcal{T}} (Z(\{t\})\; u^{t'*})\; c^t\; c^{t'} - \sum_{t\in T} \sum_{d\in\mathcal{D}} u^{t*}\; \lambda_d\; c^t. \tag{A16}$$

The bi-linearities can be eliminated in the standard way by introducing auxiliary variables $C^{tt'}$, Λ_d^t expressing, respectively, the products $c^t c^{t'}$ and $\lambda_d c^t$ and the corresponding (forcing) constraints. In effect we get the following MIP formulation of PP(u^*, y^*):

$$B(u^*, y^*) = \max \left\{ \sum_{d\in\mathcal{D}} \lambda_d + \left(\sum_{t,t'\in\mathcal{T}} (Z(\{t\})\; u^{t'*})\; C^{tt'} - \right.\right.$$
$$\left.\left. - \sum_{t\in\mathcal{T}} \sum_{d\in\mathcal{D}} u^{t*}\; \Lambda_d^t \right) - Y \right\} \tag{A17a}$$

$$\sum_{t\in\mathcal{T}} a^t = 1 \tag{A17b}$$

$$\sum_{t\in\mathcal{T}} b^t = 1 \tag{A17c}$$

$$a^t + b^t \leq 1 \qquad t \in \mathcal{T} \tag{A17d}$$

$$c^t = c^{t\ominus 1} + a^t - b^t \qquad t \in \mathcal{T} \tag{A17e}$$

$$Y = \sum_{t\in\mathcal{T}} y^{t*}\ c^t \tag{A17f}$$

$$\sum_{e\in\mathcal{E}} \pi_e^t = c^t \qquad t\in\mathcal{T} \tag{A17g}$$

$$\lambda_d \le \sum_{e\in\mathcal{E}(d,p)} \frac{1}{c(e)} \sum_{t\in\mathcal{T}} h(d,t)\ \pi_e^t \qquad d\in\mathcal{D}, p\in\mathcal{P}(d) \tag{A17h}$$

$$C^{tt'} \le c^t,\ \ C^{tt'} \le c^{t'},\ \ C^{tt'} \ge c^t + c^{t'} - 1 \qquad t,t'\in\mathcal{T} \tag{A17i}$$

$$\Lambda_d^t \le M(d)c^t,\ \ \Lambda_d^t \le \lambda_d,\ \ \Lambda_d^t \ge \lambda_d - M(d)(1-c^t) \qquad t\in\mathcal{T}, d\in\mathcal{D} \tag{A17j}$$

$$Y \in \mathbb{R}^+ \tag{A17k}$$

$$a^t, b^t, c^t \in \mathbb{B} \qquad t\in\mathcal{T} \tag{A17l}$$

$$\lambda_d \in \mathbb{R} \qquad d\in\mathcal{D} \tag{A17m}$$

$$\pi_e^t \in \mathbb{R}^+ \qquad t\in\mathcal{T}, e\in\mathcal{E} \tag{A17n}$$

$$C^{tt'} \in \mathbb{R}^+ \qquad t,t'\in\mathcal{T} \tag{A17o}$$

$$\Lambda_d^t \in \mathbb{R}^+ \qquad t\in\mathcal{T}, d\in\mathcal{D}. \tag{A17p}$$

Above, the quantity $M(d)$ represents an upper bound on variable λ_d. For example the following formula can be used:

$$M(d) = \frac{\sum_{t\in\mathcal{T}} h(d,t)}{c(e)}. \tag{A18}$$

Finally, observe that analogous pricing problems can be formulated for APP(C) in versions (A9) and (A12).