On Optimization of Semi-stable Routing in Multicommodity Flow Networks

Artur Tomaszewski
Institute of Telecommunications,
Warsaw University of Technology
Warsaw, Poland
a.tomaszewski@tele.pw.edu.pl

Michał Pióro
Institute of Telecommunications,
Warsaw University of Technology
Warsaw, Poland
m.pioro@tele.pw.edu.pl

Davide Sanvito
Dipartimento di Elettronica,
Informazione e Bioingegneria,
Politecnico di Milano
Milan, Italy
davide.sanvito@polimi.it

Ilario Filippini
Dipartimento di Elettronica,
Informazione e Bioingegneria,
Politecnico di Milano
Milan, Italy
ilario.filippini@polimi.it

Antonio Capone
Dipartimento di Elettronica,
Informazione e Bioingegneria,
Politecnico di Milano
Milan, Italy
antonio.capone@polimi.it

ABSTRACT

Ideally, the network should be dynamically reconfigured as traffic evolves. Unfortunately, even in SDN paradigm, network reconfigurations cannot be too frequent due to a number of reasons related to route stability, forwarding rules instantiation, individual flows dynamics, traffic monitoring overhead, etc.

In this paper, we focus on the fundamental problem of deciding whether, when, and how to reconfigure the network during traffic evolution. 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 clusters) and when each routing configuration must be used for at least a given minimum amount of time.

We propose a solution method based on cluster generation that provides a good lower bound on the minimum network delay (i.e., the total of link delays) and scales well with the size of the network.

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. Nevertheless, network operators have traditionally privileged routing stability by resorting to approaches, like oblivious routing [1] and robust routing [9, 15, 16], that apply static routing designs based on"worst case" traffic conditions. This unavoidably creates overprovisioning and suboptimal utilisation of network capacity.

Recently, Software-Defined Networking (SDN) has provided tools for making online network reconfiguration a potentially viable solution: dynamic routing reconfigurations can be applied at the network devices to optimize performance as the traffic evolves [4, 7, 8, 12]. However, reconfiguring the network too frequently can in general affect its stability since reprogramming flow rules can take longer than the reconfiguration period.

A group of hybrid approaches [2, 5, 13, 14, 17], often referred to as semi-stable routing, have been recently proposed to combine static and dynamic routing. Considering a limited set of routing configurations, each designed and activated during specific time intervals, allows for reducing the penalty of using 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 analyse 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 a set of consecutive timepoints as a (timepoint) cluster, we name this problem the semi-stable routing cluster design problem (SSRCDP). In SSRCDP the optimization objective is to minimize the network delay, i.e., 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 delay metric, the solution method we propose is general enough to cope with other types of the congestion metric.

The works on semi-stable routing available in the literature usually exhibit one of the following limitations: (i) they ignore the time domain by not providing any limit on the reconfiguration rate [2, 14, 17], (ii) the number of created clusters is limited and reconfiguration timepoints are arbitrary [2, 5]. Other semi-stable approaches have more recently been proposed to overcome these limitations [3, 11]. 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 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, i.e., it also provides a good lower bound of the network delay. In addition, this 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 below.

2 PROBLEM FORMULATION

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 and $\mathcal{E}$ is the set of (directed) links (where $c(e) \geq 0$, $e \in \mathcal{E}$, is the capacity of link e). $\mathcal{D}$ is the set of (directed) demands, where $o(d), t(d)$, $d \in \mathcal{D}$, are the originating and terminating node, respectively, 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, $d \in \mathcal{D}$, (each path $p \in \mathcal{P}(d)$ connects the demand's origin $o(d)$ with its termination $t(d)$). (Below, $\mathcal{P}$ will denote the set of all admissible paths, i.e., $\mathcal{P} := \bigcup_{d \in \mathcal{D}} \mathcal{P}(d)$.) 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 a *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:

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

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:

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

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. Further, let $F : [0, +\infty) \to [0, +\infty)$ be an increasing convex piece-wise linear function with $F(0) = 0$. We will call $F(w)$ the *delay function* (see [6, 10]) as it is supposed to measure the packet delay on a link for a given link utilization w. Finally, the quantity

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

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

We may now introduce the notion of a *cluster* $C(t,l)$ with parameters t (timepoint in which the cluster starts) and l (length of the cluster). Namely, $C(t,l)$ is the set of l consecutive timepoints that starts at timepoint t. Hence, $C(t,l) := \{t, t \oplus 1, \ldots, t \oplus (l-1)\}$, where $\oplus$ denotes addition modulo T (i.e., the timepoints are counted modulo T). For a given cluster $C = C(t,l)$, let $t(C) = t$ and $l(C) = l$ denote, respectively, the start and the length of C.

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

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

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

The *semi-stable routing cluster design problem* (SSRCDP) we consider is this: given $\mathcal{G}, \mathcal{P}, \mathcal{D}, \mathcal{T}$, and a pair of positive integer numbers $N \leq T$ and $L \leq T$, find a *partition* $\mathscr{R}$ of the set of timepoints $\mathcal{T}$ into at most N (non-empty) routing clusters, each of length at least L (i.e., $|\mathcal{R}| \geq L$, $\mathcal{R} \in \mathscr{R}$), and find 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 total maximal network utilization resulting from SSRCDP will be denoted by Z^*. Note that the assumptions on N, L, T imply that $N \leq \frac{T}{L}$, and hence $N \leq \lfloor \frac{T}{L} \rfloor$.

3 SOLUTION METHOD

3.1 The fixed partition subcase

We start with the following observation. If the sets forming a partition $\mathscr{R}$ of set $\mathcal{T}$ were given and fixed, SSRCDP would reduce to finding a routing configuration $x(\mathcal{R})$ minimizing $Z(\mathcal{R})$ for each cluster $\mathcal{R} \in \mathscr{R}$, and this could be done independently for each cluster. Thus, we first analyse 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($\mathcal{U}$)

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

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

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

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

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

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

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. In (5d), parameters $a(k), b(k)$, $k \in \mathcal{K} := \{1, 2, \ldots, K\}$, define the delay function $F(z) := \max\{a(k)z + b(k) : k \in \mathcal{K}\}$, where $b(1) = 0 > b(2) > \ldots > b(K)$, $0 < a(1) < a(2) < \ldots < a(K)$.

Note that RP($\mathcal{U}$) is a linear programming (LP) problem in a non-compact formulation that can be easily solved to optimality (even for large networks) using the column (path) generation approach based on a shortest path algorithm: 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 optimal dual variables associated with constraint (5c). A path is added to the problem if its cost is less than λ_d – optimal dual variable associated with constraint (5b). Observe that RP($\mathcal{U}$) can alternatively be formulated as an LP problem in a compact way, using the node-link notation with link flows (instead of path flows) that does not require column generation.

We end this section with the following observation.

Remark 1. *For any two sets $\mathcal{U}', \mathcal{U}$ such that $\mathcal{U}' \subseteq \mathcal{U} \subseteq \mathcal{T}$, the inequality*

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

holds, where $x^*(\mathcal{U})$ and 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 $Z(\mathcal{U}')$ would be larger than $\sum_{t \in \mathcal{U}} z^t(x^*(\mathcal{U}))$, then the routing configuration $x^*(\mathcal{U})$, when applied to $\mathcal{U}'$, would decrease the value of $Z(\mathcal{U}')$. Clearly, when $\mathcal{U} = \mathcal{U}'$ then the right hand side of* (6) *is equal to $Z(\mathcal{U}')$.*

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
$\mathcal{T} = \{0, 1, \ldots, T-1\}$	set of timepoints
$c(e)$	capacity of link e ($e \in \mathcal{E}$)
$h(d, t)$	volume of demand d to be realized in 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
C	(routing) clusters composed of timepoints
$t(C), l(C)$	starting timepoint and length (respectively) of cluster C $\quad (t(C) \in \mathcal{T},\ l(C) \in \{1, 2, \ldots, T\})$
$C(t, l)$	cluster with $t(C) = t,\ l(C) = l,\ C(t, l) = \{t, t \oplus 1, \ldots, t \oplus (l-1)\}$ ($\oplus$ denotes addition modulo T)
$\mathscr{C}$	family of (routing) clusters $\quad (C \in \mathscr{C})$
$x(C)$	routing configuration used in routing cluster C
$z(C, x) = \sum_{t \in C} z^t(x)$	cluster delay for cluster C with routing configuration x
$Z(C)$	cluster delay of C minimized over all routing configurations x ($Z(C)$ is a solution of RP(C))
$Z(C\|\infty) = \sum_{t \in C} Z(\{t\})$	a lower bound for $Z(C)$
$\mathscr{R}$	family of routing clusters forming a partition of the set of timepoints $\mathcal{T}$ into at most N $(1 \le N \le \frac{T}{L})$ routing clusters, each of length at least L $\ (\|\mathcal{R}\| \ge L,\ \mathcal{R} \in \mathscr{R})$
$z(\mathscr{R}) = \sum_{\mathcal{R} \in \mathscr{R}} z(\mathcal{R}, x(\mathcal{R}))$	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 design problem
Z^*	minimum of $Z(\mathscr{R})$ over all partitions $\mathscr{R}$ $\quad$ (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 $\mathscr{C}$
$\mathscr{R}(\mathscr{C})$	family of routing clusters solving APP($\mathscr{C}$)
$Y(\mathscr{C})$	minimum objective value of APP($\mathscr{C}$) (lower bound for SSRCDP)
CGA	cluster generation algorithm
$\mathbb{B}, \mathbb{Z}_+, \mathbb{R}_+$	$\mathbb{B} = \{0, 1\}, \mathbb{Z}_+ = \{0, 1 \ldots\}, \mathbb{R}_+$ non-negative real numbers

3.2 Approximation problem

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

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 $\mathscr{C}$ be a fixed subfamily of the family of all timepoint clusters (below the family $\mathscr{C}$ will be called a *control family* of *control clusters*), and let $Z(C|\infty) := \sum_{t \in C} Z(\{t\})$ for each $C \in \mathscr{C}$.

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

Problem APP($\mathscr{C}$)

$$Y(\mathscr{C}) = \min \textstyle\sum_{t \in \mathcal{T}} y^t \tag{7a}$$

$$\textstyle\sum_{t \in \mathcal{T}} u^t \le N \tag{7b}$$

$$\textstyle\sum_{0 \le k \le L-1} u^{t \oplus k} \le 1 \qquad t \in \mathcal{T} \tag{7c}$$

$$U^C = \textstyle\sum_{1 \le k < l(C)} u^{t(C) \oplus k} \qquad C \in \mathscr{C} \tag{7d}$$

$$Y^C = \textstyle\sum_{t \in C} y^t \qquad C \in \mathscr{C} \tag{7e}$$

$$y^t \ge Z(\{t\}) \qquad t \in \mathcal{T} \tag{7f}$$

$$Y^C \ge Z(C) + \big(Z(C|\infty) - Z(C)\big) \cdot U^C \qquad C \in \mathscr{C} \tag{7g}$$

$$u^t \in \mathbb{B},\ y^t \in \mathbb{R}_+ \qquad t \in \mathcal{T} \tag{7h}$$

$$U^C \in \mathbb{Z}_+,\ Y^C \in \mathbb{R}_+ \qquad C \in \mathscr{C}. \tag{7i}$$

Constraints (7b) and (7c) 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 $\mathscr{R}$. Then, constraint (7d) defines integer variables U^C that specify with how many clusters in family $\mathscr{R}$ a given cluster C from family $\mathscr{C}$ intersects. Note that when $U^C = 0$ then C intersects with only one cluster in $\mathscr{R}$, when $U^C = 1$ then C intersects with exactly two clusters in $\mathscr{R}$, and so on. Additionally, constraint (7e) defines an approximated cluster delay for each control cluster C.

Constraints (7f) and (7g) specify two kinds of *valid inequalities*, i.e., inequalities that are satisfied by the maximal link utilizations $z^t(x(\mathcal{R}))$, $\mathcal{R} \in \mathscr{R}, t \in \mathcal{R}$, determined (through definition (3)) by any partition $\mathscr{R}$ and any set of routing configurations $x(\mathcal{R})$, $\mathcal{R} \in \mathscr{R}$ (satisfying condition (1)).

The inequality in constraint (7f) holds since $Z(\{t\})$, as the optimal solution of RP($\{t\}$), provides the absolute lower bound on the timepoint delay for any given $t \in \mathcal{T}$. Thus, (7f) is a valid inequality. Note also, that (7f) implies that $\sum_{t \in C} y^t \ge Z(C|\infty)$.

Now observe that the right hand side of inequality in (7g) defines an affine function of variable U^C (defined by (7d)). Let us denote this function by A. Since $Z(C) \ge Z(C|\infty)$ (by definition of $Z(C|\infty)$), function A is non-increasing, and in fact strictly decreasing when $Z(C) > Z(C|\infty)$. Since $A(0) = Z(C)$, for $U^C = 0$ the inequality in (7g) reduces to $\sum_{t \in C} y^t \ge Z(C)$. Moreover, condition $U^C = 0$ means that $C \subseteq \mathcal{R}$ for some $\mathcal{R} \in \mathscr{R}$, and hence, by Remark 1, implies inequality $\sum_{t\in C} z^t(x(\mathcal{R})) \geq Z(C)$. This means that for $U^C = 0$ the inequality in (7g) is valid.

Next, since $A(1) = Z(C|\infty)$, for $U^C = 1$, inequality in (7g) reduces to $\sum_{t\in C} y^t \geq Z(C|\infty)$, which, as mentioned above, is already implied by (7f). This means that in this case (7g) is valid as well. Moreover, since A is non-increasing, $A(U) \leq A(1)$ for $U > 1$ and this means that (7g) is valid for all $U^C > 1$. Thus, (7g) is valid for all possible values of U^C, and this finally implies that APP($\mathscr{C}$) is a relaxation of SSRCDP so that its optimal solution value $Y(\mathscr{C})$ is a lower bound for the minimum network delay Z^*.

Observe that the reason for using the particular form of the inequality in (7g) is that it is stronger than inequality

$$\textstyle\sum_{t\in C} y^t \geq Z(C)\big(1 - U^C\big) \qquad C \in \mathscr{C} \tag{8}$$

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

In order to find a (suboptimal) solution of SSRCDP we can first solve APP($\mathscr{C}$) for a given control family $\mathscr{C}$, for example for the family of all clusters with length not greater than L. Then, we can solve the routing problem RP($\mathcal{R}$) for each $\mathcal{R} \in \mathscr{R}(\mathscr{C})$, where $\mathscr{R}(\mathscr{C})$ denotes the partition of $\mathcal{T}$ resulting from solving APP($\mathscr{C}$), and determine $Z(\mathscr{R}(\mathscr{C}))$, i.e., 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 family $\mathscr{C}$ in order to decrease the so obtained $Z(\mathscr{R}(\mathscr{C}))$. The following three basic properties of formulation APP($\mathscr{C}$) will help to resolve this issue.

Property 1. *Let $\mathscr{C}$ be an arbitrary family o clusters for the set of timepoints $\mathcal{T}$. For any partition $\mathscr{R}$ 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 $\mathscr{R}$ and such that for each $\mathcal{R} \in \mathscr{R}$, $y^t = z^t(x(\mathcal{R}))$, $t \in \mathcal{R}$, i.e., 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 (7b), (7c) and uniquely defines the partition $\mathscr{R}$. Also, the vector y specified in the thesis of the proposition is feasible for APP($\mathscr{C}$) since, as explained above, inequalities (7f) and (7g) are valid for any routing family $\mathscr{R}$ in question. □

Property 2. *Let $\mathscr{R}(\mathscr{C})$ be the family of clusters determined by an optimal solution of APP($\mathscr{C}$), i.e., by u^*. Then,*

$$Y(\mathscr{C}) \leq Z^* \leq Z(\mathscr{R}(\mathscr{C})), \tag{9}$$

where $Y(\mathscr{C}) = \sum_{t\in\mathcal{T}} y^{t*}$ is the optimal objective of APP($\mathscr{C}$), Z^* is the optimal objective 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 $Y(\mathscr{C}) \leq Z^*$ holds because APP($\mathscr{C}$) is a relaxation of SSRCDP. The second inequality ($Z^* \leq Z(\mathscr{R}(\mathscr{C}))$ holds because partition $\mathscr{R}(\mathscr{C})$ with optimized clusters' routing configurations is a feasible solution of SSDRP. □

Property 3. *Let $\mathscr{R}(\mathscr{C})$ denote an optimal partition resulting from APP($\mathscr{C}$) and suppose that $\mathscr{R}(\mathscr{C})$ is a subset of $\mathscr{C}$. 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($\mathscr{C}$). 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 (7g) that involve variables y^t, $t \in \mathcal{R}$, the one corresponding to $C = \mathcal{R}$ is satisfied tightly since, by assumption, $\sum_{t\in\mathcal{R}} y^t = Z(\mathcal{R})$. Since for each $C' \subset \mathcal{R}$ (whether or not C' is in $\mathscr{C}$), the inequality $\sum_{t\in C'} y^t \geq Z(C')$ holds (by Remark 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 (9), $Z(\mathscr{R}(\mathscr{C})) = Z^*$. □

3.3 Cluster generation algorithm

The above properties suggest the following algorithm for solving SSRCDP.

CGA: cluster generation algorithm

Step 0: Specify an initial family of clusters $\mathscr{C}$.
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 $\frac{Z(\mathscr{R}(\mathscr{C}))-Y(\mathscr{C})}{Y(\mathscr{C})} \leq \varepsilon$ then stop: $\mathscr{R}(\mathscr{C})$ is suboptimal (or even optimal) family of routing clusters 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 family $\mathscr{R}(\mathscr{C})$ 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 $\mathscr{R}(\mathscr{C}) \setminus \mathscr{C} \neq \emptyset$ 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 CGA will stop even if $\varepsilon = 0$ is assumed (and then return an optimal partition $\mathscr{R}(\mathscr{C})$ for SSRCDP) in a finite number of steps, because the number of all clusters is finite. This, however, can take an excessive computation time.

3.4 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 partition $\mathscr{R}$ defined by a binary vector $u = (u^t)_{t\in\mathcal{T}}$ feasible for APP($\mathscr{C}$), i.e., fulfilling (7b) and (7c).

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

$$Z(C|1) := \min_{1\leq k\leq l-1} \big\{Z(C(\tau, k)) + Z(C(\tau \oplus k, l-k))\big\}. \tag{10}$$

Then the inequality

$$\textstyle\sum_{t\in C} y^t \geq Z(C|1) \tag{11}$$

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 $\mathscr{R}$ specified by u. Then $C = C(\tau, k) \cup C(\tau \oplus k, l-k)$ for some $1 \leq k \leq l-1$. Let $C' = C(\tau, k) \cap \mathcal{R}'$ and $C'' = C(\tau \oplus k, l-k) \cap \mathcal{R}''$. Since, by Remark 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 (11) is a valid inequality. Note that when in an optimal solution of APP($\mathscr{C}$), $C' = \mathcal{R}'$ and $C'' = \mathcal{R}''$ and inequality (11) becomes tight. □

Clearly, for $U^C = 1$, inequality (11) is tighter than the inequality implied by constraint (7g) (recall that $Y^C := \sum_{t\in C} y^t$) since in general $Z(C|1) > Z(C|\infty)$ (see Remark 1). Thus, substituting constraint (7g) in (7) with

$$Y^C \geq Z(C) + \big(Z(C|1) - Z(C)\big) \cdot U^C \qquad C \in \mathscr{C} \qquad (12)$$

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

Observe however, that for $U^C \geq 2$, inequality (12) is in general not valid. For example, for $U^C = 2$, the value of $Z(C) + \big(Z(C|1) - Z(C)\big) \cdot 2$ can be greater than the proper value given by the following formula (analogous to (10)):

$$\begin{aligned} Z(C|2) := \min_{1 \leq k_1 < k_2 \leq l-1, k_2 - k_1 \geq L} \{Z(C(\tau, k_1)) + \\ + Z(C(\tau \oplus k_1, k_2 - k_1)) + Z(C(\tau \oplus k_2, l - k_1 - k_2))\}. \end{aligned} \qquad (13)$$

It follows that MAPP($\mathscr{C}$) is correct only when the control family $\mathscr{C}$ is a subfamily of $\mathscr{C}(L+1)$ – the family of all clusters of length at most $L+1$ – since only then it is guaranteed that $U^C \leq 1$ for all $C \in \mathscr{C}$, and thus inequality in (12) is valid. Thus, the modified problem cannot be used in the CGA algorithm, as in general the family $\mathscr{R}(\mathscr{C})$ contains clusters with length larger than $L+1$ and such sets cannot be added to the control cluster 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 4, even this (non-iterative) solution gives very good results when applied to SSRCDP.

3.5 Improvements

The efficiency of the CGA algorithm described in Section 3.3 can be improved in two complementary ways.

First, the linear relaxation of formulation (7) can be strengthened (by improving, i.e., increasing, the lower bound 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 $\frac{Z(\mathscr{R}(\mathscr{C})) - Y(\mathscr{C})}{Y(\mathscr{C})}$ between the integer solution and the relaxed solution. The lower bound computed through the linear relaxation of formulation (7) can be increased by improving valid inequalities specified in constraint (7g). In fact, these valid inequalities are tight only for the case $U^C = 0$, i.e., when the control cluster C is contained in a cluster of the constructed family of routing clusters $\mathscr{R}$. (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 (7g) are weaker than the inequality in (7f), 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 (7g) 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 $M(C) := \lceil \frac{l(C)-1}{L} \rceil$ is an upper bound for U^C, and for integer values of the argument U, $G^C(U) = Z(C|U)$, where $Z(C|0) := Z(C)$, $Z(C|1)$ is defined by (10), $Z(C|2)$ by (13) and $Z(C|U)$, $U \geq 3$, are defined analogously. Then, the valid inequality in (7g) 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.)

Second, on top of the family of clusters $\mathscr{R}(\mathscr{C})$ that is added to the control family $\mathscr{C}$ in Step 2 of CGA, we may seek to add extra control sets C' for which constraints (7g) are broken to the largest extent by the the current optimal values y^*.

4 NUMERICAL EXPERIMENT

Below we describe a numerical experiment illustrating the efficiency of the proposed APP($\mathscr{C}$)-based approach for a network linking 47 cities in an European Union country. The network consists of 47 nodes linked with 140 directed links (each of capacity 4 Gbps), and $47 \times 46 = 2162$ traffic demands corresponding to all ordered pairs of nodes. The demand volumes used in the calculations are derived from real traffic measurements (obtained from a network operator) taken every 15 minutes on a selected weekday (a Wednesday in 2018). Thus, the number of considered timepoints equals 96 ($T - 1 = 95$). We set the maximal number of clusters 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.

In the experiment reported below, for solving the semi-stable routing cluster design problem (SSRCDP) we used formulation MAPP($\mathscr{C}$) in the way described in Section 3.4. The procedure was implemented using the platform: Lenovo Thinkpad, Intel i7-6500U, 8GB 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 $2T = 192$ of such clusters, and thus, in the preprocessing phase, for each of them we need to calculate the values $Z(C)$ and $Z(C|1)$ according to formulae (5a) and (11), respectively. For that, the routing problem RP($\mathcal{U}$) (5) is solved $8T = 768$ times, i.e., for all clusters of length between 2 and 9.

In RP($\mathcal{U}$) the delay function $F(z) := \max\{0.1z, z - 0.45, 10z - 8.5\}$ (with $K = 3$ linear pieces) was used, i.e., $b(1) = 0, b(2) = -0.45, b(3) = -8.5$ and $a(1) = 0.1, a(2) = 1, a(3) = 10$. Thus, $F(z)$ 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 bound (column LB) and the upper bound (column UB) for the optimal objective function value, and the current 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 in the task, i.e., clusters for which we solve the routing problem, and in brackets, if applicable, the number of clusters that are contained in the control set of the partitioning problem. Finally, column N_{PATHS} first shows (in brackets, with the plus sign) the total number of paths that we have 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.

In the row STATIC ROUTING, the case when only one routing cluster, i.e., $\mathcal{T}$, is applied. Then an optimized single routing scheme gives the optimal objective equal to $Z(\mathcal{T})$ given in the column UB, as this value is the upper bound for the true SSDRP optimal solution value. The row DYNAMIC ROUTING corresponds to the case when each timepoint is considered as a cluster, i.e., the routing scheme is optimized individually for each timepoint. Hence, the column LB in this row indicates $\sum_{t \in \mathcal{T}} Z(\{t\})$ which is clearly the cheapest solution value to SSRCDP (the case when the partition to the routing clusters is unconstrained). The value in column GAP, equal to $\frac{\text{UB-LB}}{\text{LB}} \times 100\%$ (UB taken for STATIC ROUTING and LB taken for DYNAMIC ROUTING), is indicated. The row PREPROCESSING contains information concerning preparation of the control cluster family $\mathscr{C}$ and initial routing paths (recall the RP($\mathcal{U}$) is solved through path generation). Next, the row PARTITIONING LR shows the results of solving the linear relaxation of the modified APP($\mathscr{C}$) formulation, i.e., of problem MAPP($\mathscr{C}$) described in Section 3.4. The so obtained value of LB happens to be the same as for DYNAMIC ROUTING, although in general it could be larger. Further, the solution of the MIP formulation

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%	1s	(192)	-
PARTITIONING MIP	550.50	-	2.43%	2s	(192)	-
ROUTING	-	551.86	0.25%	1m16s	8	(+1) 8011

MAPP($\mathscr{C}$) is described in the row PARTITIONING MIP. The LB value delivered by this solution is increased with respect to the preceding row and hence the GAP value is decreased. Finally, the row ROUTING shows the results for the partitioning $\mathscr{R}(\mathscr{C})$ obtained with the MIP formulation MAPP($\mathscr{C}$) with the routing scheme optimized for each of the resulting routing clusters $\mathcal{R}$. In particular, UB gives the value of $Z(\mathscr{R}(\mathscr{C}))$. Observe that the gap between this feasible SSRCDP solution and the best lower bound obtained with PARTITIONING MIP is very small and equals 0.25%. In the final solution, the optimal routing cluster family $\mathscr{R}(\mathscr{C})$ 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 SSRDCP in a reasonable time within the optimality gap as small as 0.25%.

5 CONCLUSIONS

In this paper we propose a scalable solution to the problem of designing clusters of the 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 poor linear relaxation. Thus we were considering a number of exact and hybrid approaches (as in [13]) that aimed at separating 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 T is typically between 96 and 288 as the traffic measurement period is either 5 or 15 minutes), our numerical trials show that in practice we cannot analyze all those clusters. Using a link-path formulation combined with path generation and a warm start for the master problem, it took around k seconds to solve the routing problem for a cluster of length k and a 50-node network. And this 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 cluster preprocessing. Our algorithm is capable of providing the upper and the lower objective function value bounds with very low optimality gaps, well below 0.5%, as shown in the presented numerical study (and some other studies not reported here for the lack of space). It also offers the trade-off between the quality of the solution, and the number of clusters in the control set that influences the preprocessing time, and the size and the solution time of the partitioning problem.

In addition, we have proposed two possible ways for improving the efficiency of the approach that lead to interesting future research. First, we can use a stronger formulation of APP($\mathscr{C}$) equipped with improvements described in Section 3.5. Second, we can either implement a full version of the cluster generation algorithm presented in Section 3.3, or, even better, to incorporate cluster generation into a branch-and-bound procedure of solving the partitioning problem, by analyzing relaxed or incumbent solutions and generating appropriate user cuts. We will also aim at testing the resulting optimization procedure on examples with lower correlation among the traffic matrices, which might feature a more substantial gap between the static and dynamic routing solutions than the 3.33% observed in the current example (which our algorithm nonetheless managed to decrease tenfold).

ACKNOWLEDGMENT

The work of the Polish authors was supported by the National Science Center, Poland, grant no. 2015/17/B/ST7/03910 "Logical tunnel capacity control – a traffic routing and protection strategy for communication networks with variable link capacity".

REFERENCES

[1] Yossi Azar, Edith Cohen, Amos Fiat, et al. 2003. Optimal oblivious routing in polynomial time. In *ACM Symp. on Theory of Computing*. 383–388.

[2] Walid Ben-Ameur and Mateusz Żotkiewicz. 2011. Robust routing and optimal partitioning of a traffic demand polytope. *Intl. Trans. in Operational Research* 18, 3 (2011), 307–333.

[3] Walid Ben-Ameur and Mateusz Żotkiewicz. 2013. Multipolar routing: where dynamic and static routing meet. *Electronic Notes in Discrete Mathematics* 41 (2013), 61–68.

[4] Theophilus Benson, Ashok Anand, Aditya Akella, and Ming Zhang. 2011. MicroTE: Fine grained traffic engineering for data centers. In *Proc. ACM CoNext*. 8.

[5] Pedro Casas, Lionel Fillatre, and Sandrine Vaton. 2008. Multi Hour Robust Routing and Fast Load Change Detection for Traffic Engineering. In *Proc. IEEE ICC*. 5777–5782.

[6] Bernard Fortz and Mikkel Thorup. 2002. Optimizing OSPF/IS-IS weights in a changing world. *IEEE JSAC* 20, 4 (2002), 756–767.

[7] Chi-Yao Hong, Srikanth Kandula, Ratul Mahajan, et al. 2013. Achieving high utilization with software-driven WAN. In *ACM SIGCOMM CCR*, Vol. 43. 15–26.

[8] Sushant Jain, Alok Kumar, Subhasree Mandal, et al. 2013. B4: Experience with a globally-deployed software defined WAN. *ACM SIGCOMM CCR* 43, 4 (2013), 3–14.

[9] Murali Kodialam, TV Lakshman, and Sudipta Sengupta. 2004. Efficient and robust routing of highly variable traffic. In *Proc. HotNets*.

[10] Michał Pióro and Deep Medhi. 2004. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan-Kaufmann.

[11] Michael Poss and Christian Raack. 2013. Affine recourse for the robust network design problem: Between static and dynamic routing. *Networks* 61, 2 (2013), 180–198.

[12] Matthew Roughan, Mikkel Thorup, and Yin Zhang. 2003. Traffic engineering with estimated traffic matrices. In *Proc. ACM IMC*.

[13] Davide Sanvito, Ilario Filippini, Antonio Capone, Stefano Paris, and Jeremie Leguay. 2018. Adaptive Robust Traffic Engineering in Software Defined Networks. In *Proc. IFIP Networking*.

[14] Marco Silva, Michael Poss, and Nelson Maculan. 2018. Solving the bifurcated and nonbifurcated robust network loading problem with k-adaptive routing. *Networks* 72, 1 (2018), 151–170.

[15] Vahid Tabatabaee, Abhishek Kashyap, Bobby Bhattacharjee, Richard J La, and Mark A Shayman. 2007. Robust routing with unknown traffic matrices. In *Proc. IEEE INFOCOM*. 2436–2440.

[16] Hao Wang, Haiyong Xie, Lili Qiu, Yang Richard Yang, Yin Zhang, and Albert Greenberg. 2006. COPE: traffic engineering in dynamic networks. In *ACM SIGCOMM CCR*, Vol. 36. 99–110.

[17] Yin Zhang and Zihui Ge. 2005. Finding critical traffic matrices. In *Proc. IEEE DSN*.