Models and methods for Traffic Engineering problems with single-path routing

Martim Joyce-Moniz

Université Libre de Bruxelles
Faculté des Sciences
Département d'Informatique

Models and methods for Traffic Engineering problems with single-path routing

Martim Joyce-Moniz

Thèse présentée en vue de
l'obtention du grade académique
de Docteur en Sciences

October 2016

To my family.

This thesis has been written under the supervision of Prof. Bernard Fortz and Prof. Luís Gouveia. The members of the jury are:

- Prof. Bernard Fortz (Université Libre de Bruxelles, Belgium)
- Prof. Bernard Gendron (Université de Montreal, Canada)
- Prof. Luís Gouveia (Universidade de Lisboa, Portugal)
- Prof. Martine Labbé (Université Libre de Bruxelles, Belgium)
- Prof. Ivana Ljubic (ESSEC Business School Paris, France)
- Prof. Thomas Stützle (Université Libre de Bruxelles, Belgium)

The research of Martim Joyce-Moniz was done with the support of the COMEX (Combinatorial Optimization: Metaheuristics & Exact Methods) project.

Acknowledgements

Over the last 4 years, I've often expressed how lucky I am, and how luck was fundamental for me to arrive where I have. With this, I don't mean to indulge in false modesty - naturally, a mix of hard work and good decisions played a part. Nor do I intent to conjecture about the nature, meaning or reason for said luck. What I do wish, is to simply acknowledge the often unsung role of good fortune, every time one's fate does not merely depend on will and want.

I am extremely lucky to have had a pair of supervisors, Bernard and Luís, who ideally epitomized the job, by providing all the necessary support and orientation to successfully write this thesis. Luís, who I was lucky to have studied under, and learn about network optimization with. Luís, who recognized and understood my yearn to do a Ph.D. abroad, and immediately put me in contact with the right people. Luís, whose great knowledge and good advise I could always count on, only an email or phone call away. Bernard, who I was lucky to have take a chance on me. Bernard, who always gave me the freedom to grow as a researcher, yet always had his door open, ready to offer me guidance and support, when needed. Bernard, whose hours spent with, staring at formulations in the blackboard, were among the most fascinating and enriching in my lifetime. Bernard, who always went a step further to ensure I evolved in the OR community, by allowing me to attend a great number of conferences and PhD schools.

I am very lucky to belong to a thriving and exciting academic community, filled with brilliant and friendly people. Of this community, I must highlight of course, the members of my jury, Bernard G., Ivana, Martine and Thomas, to whom I extend my deepest appreciation. Their precious time spent reading my thesis, and their constructive and helpful comments were invaluable.

I cannot emphasize enough how lucky I am to count on their love, support and upbringing of my parents. All that I am today, is thanks to how they raised me. I am remarkably lucky that they have always pushed me to follow my dreams, even if they take me to far away lands. I am also exceedingly lucky to have a sister and brother-in-law, who I can always count on for inspiration, guidance and friendship.

I am exceptionally lucky to have many great friends, who are always ready to celebrate the good times, and help me go through the tough ones - in Lisbon, Brussels, and other places around the world.

Finally, I am just-the-luckiest, to have an amazing girlfriend, Ashley, who manages to prove to me everyday that love, care and encouragement are not a function of physical distance.

Abstract

Traffic Engineering (TE) uses methods and models from a variety of mathematical fields, such as statistics and optimization, to improve the performance of telecommunication networks. In this thesis, we study TE problems dealing with networks that impose single-path routing. As the name infers, in this type of routing, the traffic flow of each "commodity" cannot be split in its path between its origin and destination. Given its cheap cost, single-path routing is widely used in today's data centers, where thousands of stored servers perform computations or host Internet services. One common case of single-path routing is the one enforced by the Spanning Tree Protocol (STP) in switched Ethernet networks. The STP requires the network to keep its activated links loop-free, while maintaining the other redundant links ready for back-up, in case of link failure. The Multiple Spanning Tree Protocol (MSTP) extends the STP by installing multiple virtual networks compliant with the STP, over a single physical topology. Therefore, the MSTP is greatly beneficial for network service providers, as it allows for a more efficient use of the existing resources.

Network design problems dealing with the MSTP are generally highly combinatorial and very hard to solve. As such, TE literature mainly suggests heuristic methods, which can quickly produce reasonable designs. Notwithstanding, due to a scarce existence of lower bounds to the optimum values of such problems, there is little knowledge about the quality of the solutions provided by these heuristics.

In this sense, we propose mathematical programming models and methods that can provide optimal designs for these networks, or at the very least, obtain valid lower bounds. Taking into mind the goal of avoiding congestion in the network, we focus on two problems that deal with the following load-balancing objectives: the minimization of the worst-case link utilization, and the minimization of flow costs given by piecewise linear functions that penalize heavily-loaded links. The study of both these problems yielded relevant by-products: the first is the study of a MSTP network design problem, where we minimize the total load, and the second is the study of a fundamental unsplittable multicommodity flow problem with piecewise linear costs.

For all the considered problems, we provide studies of complexity, extensive polyhedral studies to compare the proposed formulations, and a wide array of computational experiments to evaluate the performance of the proposed models and methods.

Résumé

L'Ingénierie du Trafic (IT) utilise des méthodes et des modèles de plusieurs domaines mathématiques, comme la statistique et l'optimisation, pour améliorer les performances des réseaux de télécommunication. Dans cette thèse, nous étudions des problèmes de l'IT qui traitent de réseaux qui imposent l'utilisation du mono-routage. Comme le nom l'indique, dans ce type de routage, le flux de trafic de chaque demande ne peut pas tre divisé sur son trajet entre son origine et sa destination. Vu son coût faible, le mono-routage est largement utilisé dans les centres de données actuels, où des milliers de serveurs effectuent des calculs ou hébergent des services Internet. Une utilisation courante du mono-routage est l'usage du *Spanning Tree Protocol* (STP), basé sur les arbres couvrants, dans les réseaux de type *switched Ethernet*. Le STP impose que le réseau maintienne un ensemble de liens activés sans circuits, tandis que les autres liaisons sont redondantes et prtes pour la sauvegarde, en cas de défaillance d'un lien. Le *Multiple Spanning Tree Protocol* (MSTP) étend STP, en installant plusieurs réseaux virtuels se conformant à STP, sur une seule topologie physique. Par conséquent, MSTP est très bénéfique pour les fournisseurs de services réseau, car il permet une utilisation plus efficace des ressources existantes.

Les problèmes de conception de réseau avec MSTP sont généralement très combinatoires et très difficiles. De ce fait, la littérature sur l'IT suggère principalement des méthodes heuristiques, qui peuvent rapidement produire des solutions raisonnables. Néanmoins, en raison de l'existence de rares bornes inférieures aux valeurs optimales de ces problèmes, il y a peu de connaissances sur la qualité des solutions fournies.

En conséquence, nous proposons des modèles de programmation mathématique et des méthodes qui peuvent fournir des plans optimaux pour ces réseaux, ou à tout le moins, obtenir des bornes inférieures valides. Compte tenu de l'objectif d'éviter la congestion dans le réseau, nous nous concentrons sur deux problèmes qui traitent les objectifs d'équilibrage de charge suivants : la minimisation de l'utilisation maximale des liens et la minimisation des coûts des flux donnés par des fonctions linéaires par morceaux qui pénalisent les liens très chargés. L'étude de ces deux problèmes a abouti à des résultats dérivés pertinents: le premier est l'étude d'un problème de conception de réseau utilisant MSTP, o nous minimisons la charge totale, et le second est l'étude d'un problème fondamental de mono-routage avec des coûts linéaires par morceaux.

Pour tous les problèmes examinés, nous fournissons des études de complexité, de vastes études polyédriques pour comparer les formulations proposées, et un large éventail d'expériences de calcul pour évaluer la performance des modèles et des méthodes proposées.

Contents

List of Figures

List of Tables

List of Formulations

Glossary

ASM the aggregated strong model forthe PUMF problem

B-RDMFM the basic rooted directed multicommodity flow problem for the TE-MSTP-p problem

B&B Branch-and-bound

B&C Branch-and-cut

BM1 the basic model 1 for the PUMF problem

BM1-r the basic model 1 for the NCPMF problem

BM2 the basic model 2 for the PUMF problem

BSA Binary search algorithm

DM the disaggregated model for the PUMF problem

Gbps Billions of bits per second

IEEE Institute of Electrical and Electronics Engineers

LAN Local area network

LP Linear program(ming)

MAC address Media access control address

Mbps Megabits per second

MFM the multicommodity flow model for the TE-MSTP problem

MFM-avg the multicommodity flow model for the COCMST problem

MIP Mixed-integer linear program(ming)

MST problem Minimum spanning tree problem

MSTP Multiple Spanning Tree Protocol

NCPMF problem Non-convex piecewise linear multicommodity network flow problem

NPUMF problem Non-convex unsplittable multicommodity flow problem

OCST problem Optimum (optimal) communication spanning tree problem

ODIMCF problem Origin-destination integer multicommodity flow problem

PUMF problem Convex unsplittable multicommodity flow problem

QoS Quality of service

RDM the rooted directed model for the TE-MSTP problem

RDM-0 the rooted directed model for the TE-MSTP decision problem

RDM-avg the rooted directed model for the COCMST problem

RDMFM the rooted directed multicommodity flow model for the TE-MSTP problem

RDMFM-0 the rooted directed multicommodity flow model for the TE-MSTP decision problem

RDMFM-avg the rooted directed multicommodity flow model for the COCMST problem

S-RDMFM the strengthened rooted directed multicommodity flow problem for the TE-MSTP-p problem

SAT problem Boolean satisfiability problem

SEN Switched Ethernet network

SM the strong model for the PUMF problem

SM-n the strong model for the NPUMF problem

SM-r the strong model for the NCPMF problem

SP Sub-problem

STP Spanning Tree Protocol

TE Traffic engineering

TE-MSTP problem Traffic engineering for the Multiple Spanning Tree Protocol problem

TE-MSTP-p problem Traffic engineering for the Multiple Spanning Tree Protocol with piecewise-linear costs problem

ToR Top of Rack

VLAN Virtual local area network

Introduction

In the turn of the century, the Danish mathematician, statistician and engineer, Agner Krarup Erlang (1878 - 1929) first invented the field of telecommunications traffic engineering, teletraffic engineering or simply, traffic engineering (TE) [ML97]. Originally, TE was essentially characterized by the application of probability theory to telephone networks. The tools used in the discipline included stochastic processes, queueing theory and numerical simulation, and the goal was to evaluate, operate and maintain telecommunications networks [I+05].

Ever since then, propelled by the Digital Revolution, the field has increased in importance, as well as in breadth. Nowadays, TE also includes a wide array of optimization methods, which aim at finding the best network configuration, in order to improve different traffic-oriented performance measures, such as delay, delay variation, packet loss and throughput. The ultimate goal of these methods is to optimize the quality of service (QoS), while considering the general cost of the network [Wir97].

TE is, thus, a key element for the design, operation and maintenance of all kinds of telecommunications networks, such as public switched telephone networks, local area networks (LANs), Ethernet networks, wide area networks, cellular telephone networks and the Internet networks [Ho12].

In this thesis, we study TE problems with single-path routing. As the name indicates, in single-path routing, the traffic flow of each "commodity" cannot be split among multiple paths between its origin and destination. Even though the converse, multipath routing, is generally perceived as resulting in better performance for different QoS measures ([GK04]), single-path routing is still widely used in our days, as it is traditionally a cheaper option. One example of single-path routing is enforced by the Spanning Tree Protocol (STP), common in switched Ethernet networks (SENs). The STP requires the network's active links to be loop-free, while keeping redundant links as back-up in case of link failure.

SENs can also implement the Multiple Spanning Tree Protocol (MSTP). This protocol allows for the providers to partition the traffic in a SEN, and assign it to different virtual LANs (VLANs), without infringing the topological requirements of the STP.

The MSTP is highly beneficial for the QoS of these networks, as the traffic can be spread throughout a bigger number of the existing links. Nevertheless, it is hard to use effectively, due to the difficulty of optimizing over multiple spanning trees, juxtaposed in one single capacitated network. For that reason, it is not surprising that the state

of the art concerning the MSTP is comprised, almost exclusively, of heuristic methods (see Section 1.2 for a review of these methods). These methods are reported to be able to efficiently produce good designs with respect to different performance measures. Notwithstanding, little-to-nothing is known about the optimality or quality of these solutions, as lower bounds to the optimum values of the respective problems are seldom provided. Accordingly, in this thesis, we propose models and exact methods for network design problems dealing with the MSTP that yield optimal designs or, at the very least, produce bounds that can shed light on the quality of heuristic solutions.

In the problems considered in this thesis, the goal is to avoid congestion in the network, by selecting network designs whose link utilization (ratio between the load and the link's capacity) is low. In Chapter 2, we study the TE for the MSTP (TE-MSTP) problem, first proposed by Ho *et al.* [HDBF11, Ho12]. In this problem, we aim at designing capacitated networks implementing the MSTP, such that the resulting routing of the traffic demands minimizes the worst-case edge utilization. We show that this problem is $\mathcal{NP}$-hard. We propose three mixed-integer linear programming (MIP) formulations for this problem, which we compare both via polyhedral studies, and extensive computational experiments. We also propose a branch-and-cut (B&C) algorithm for the TE-MSTP problem, based on Benders' decomposition of one of the proposed models.

In Chapter 3, we study an almost identical problem, the capacitated optimum communication multiple spanning tree (COCMST) problem; the difference is the objective, which is to minimize the sum of the loads throughout the network. Any feasible solution for this problem is also feasible for the TE-MSTP problem with a guaranteed worst-case edge utilization. As such, we integrate the solving of the COCMST problem in a binary search algorithm that for some instances is able to efficiently produce near-optimal solutions.

In Chapter 4, we consider a more basic network design problem: a multicommodity flow problem with single-path routing, and with costs on arc load given by piecewise-linear cost functions. Despite its apparent simplicity, we show that this problem is $\mathcal{NP}$-complete, when there is more than one commodity at stake. We propose a strengthened MIP formulation, whose linear relaxation always gives the optimal solution of the problem for the single commodity case. We present a wide array of computational experiments, that show that this formulation also produces very tight linear programming bounds for the multicommodity case. Since solving this formulation with the available MIP solvers (*e.g.* CPLEX), can be a lengthy procedure for some more complicated instances of the problem, we also propose two B&C algorithms: in the first one, we embed a Benders' decomposition method; in the second, we project the strong valid inequalities of our strengthened formulation onto a more compact model. From the interpretation of the Benders' cuts yielded by the latter, we infer a second strengthened formulation, with less variables than the first one.

The structure of the problem described in the above paragraph can be integrated in more complicated network design problems, with single-path routing. Accordingly, in Chapter 5, we study the TE-MSTP-p problem, where the goal is to design networks implementing the MSTP, such that we minimize the total flow cost given by convex

piecewise-linear linear functions. To model this problem, we combine the best formulations for the TE-MSTP and the PUMF problems. Through computational experiments, we analyze how solving this problem compares to solving the TE-MSTP problem, and how the solutions for both problems relate.

Finally, in Chapter 6, we summarize the main contributions of the work described in this thesis, and draw conclusions.

Chapter 1

Background and related work

We begin this chapter by motivating the problems that are studied in this thesis, and introducing the switching protocols on switched Ethernet networks, namely the Multiple Spanning Tree Protocol (Section 1.1). In Section 1.2, we review the state-of-the-art on optimized implementations of the latter. The large majority of the literature hinges on heuristic approaches. As such, in Section 1.4, we make an overview of different mathematical models that have been proposed, to tackle two fundamental spanning tree optimization problems. Lastly, Section 1.5 focuses on the few mathematical programming formulations that have been proposed for problems dealing with the Multiple Spanning Tree Protocol.

1.1 An introduction to switching protocols

In this section, we introduce the basic concepts of telecommunications and traffic engineering (TE), that motivate the problems studied in this thesis.

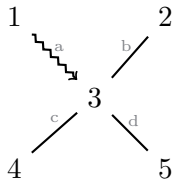
1.1.1 Data Centers and switched Ethernet networks

With the increasing demand for Internet and cloud computing services, the need for large-scale data centers has become paramount. In 2010, [BAM10] reported that the biggest online service providers (*e.g.* Google, Microsoft and Amazon) were building cloud data centers with upwards of 10K servers; it is expected that these numbers will quickly rise in the upcoming years. For the most part, data centers are used to either perform computation or to host Internet services. They support simultaneously multiple applications that run on a set of virtual machines, distributed on physical servers.

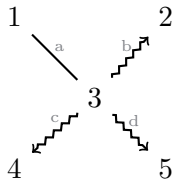
In these data centers, switched Ethernet networks (SENs) are a popular choice, as they offer better port density at a lower price per Gbps (billions of bits per second). Better port density translates into the capacity to carry larger amounts of traffic flow per unit of space in the data center. SENs are very similar to the traditional shared Ethernet networks. In fact, they are identical, with the exception that hubs are replaced by switches. A switch, like the hub, acts as a junction box and a repeater. However,

they differ in the way they transmit data packets (or frames) received by a port (or segment): while the hub retransmits it to all the other segments, the switch is able to identify the destination address of the incoming packet, and forward it only through the segments it needs to [RVJ99]. This means that contrary to what happens in the shared case, in SENs, hosts do not have to compete for the same bandwidth.

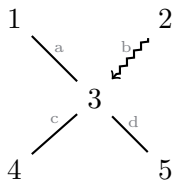
In order to identify which ports to forward the message through, switches store a lookup table, where the "address" of each node in the network is associated with a segment. In this way, when an incoming packet is received, the media access control (MAC) address is read from the frame's header and compared to the list of addresses in the lookup table. To create these lookup tables, Ethernet switches use a process called transparent bridging every time a new node is added to the network [Tys11]. Typically, these nodes are computers, but other examples can include a printer, a digital telephone handset, or even a switch. Figure 1.1 illustrates an example of the transparent bridging process. The process begins when node 1 sends a frame to node 2. In Figure 1.1a, when switch 3 receives the frame, it reads the MAC address of node 1, and saves it on its lookup table for further use. As switch 3 does not have the information regarding the location of node 2, it forwards the frame through every segment (Figure 1.1b). Eventually the frame arrives at node 2, that acknowledges it, by sending it back to node 1. Now, as the frame passes once again through switch 3, the latter can store the MAC address of node 2 on the lookup table.



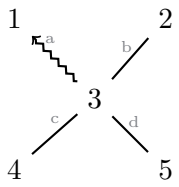
(a) $1 \to \{-, ?, ?\}$, $2 \to \{?, -, ?\}$, $3 \to \{a, ?, -\}$



(b) $1 \to \{-, ?, ?\}$, $2 \to \{b, -, b\}$, $3 \to \{a, ?, -\}$



(c) $1 \to \{-, ?, ?\}$, $2 \to \{b, -, b\}$, $3 \to \{a, b, -\}$



(d) $1 \to \{-, a, a\}$, $2 \to \{b, -, b\}$, $3 \to \{a, b, -\}$

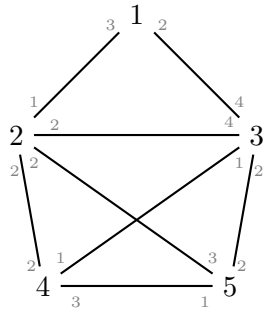
Figure 1.1: Example of transparent bridging. The caption of each subfigure represents the lookup tables of nodes 1 to 3, at each moment of the process. Notation *e.g.* $1 \to \{-, x, z\}$ indicates that in order to send a message to node 2 and 3, node 1 must relay it through, respectively, segments x and z.

If there exists two or more disjoint routes between a switch and a given node, the lookup tables will associate different ports with the destination node's MAC address. Therefore, when the switch receives a packet destined for that node, it forwards it through multiple segments. However, as the switch is part of a loop, the packet will ultimately return to it, and be treated as a new incoming frame. In this sense, loops in the topology of SENs can result in broadcast storms, *i.e.* the accumulation of broadcast and multicast traffic. This happens when the switches in the loop repeatedly rebroadcast the data packets, flooding the network [KCR08]. Ultimately, broadcast radiation can have a high impact on the performance of the network and should, therefore, be avoided at all costs.

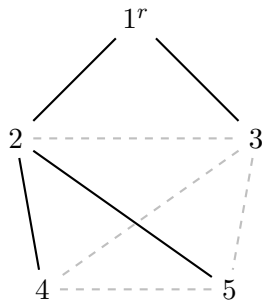
1.1.2 Spanning Tree Protocol

In SENs, it is important to keep redundant links to ensure automatic backup paths in case of link failure. However, as seen before, it is important to avoid loops in the network. As such, SENs only activate at a given time, a loop-free subset of the existing links. In this sense, these networks implement the Institute of Electrical and Electronics Engineers (IEEE) 802.1d standard [80298], also known as the Spanning Tree Protocol (STP). As the protocol's name indicates, the topology of a network using the 802.1d standard must be a spanning tree.

To configure the spanning trees, each switch is assigned two types of integer values: a BridgeID, and a PortCost for every port. The switch with the lowest BridgeID is chosen to be the Root Bridge. Then, the active links are deduced from the PortCosts: a link is selected, if it is part of the minimum cost path between each switch and the Root Bridge. The cost of each path is calculated by summing the PortCosts of the forwarding ports. If there exists two paths with the same minimum cost, between a given switch and the Root Bridge, the BridgeID of the second switch in the path is used to break the tie. An example for this procedure is depicted in Figure 1.2.



(a) Assignment of PortCosts (in gray) and BridgeIDs (same value as node id).



(b) Activated spanning tree, highlighted in black.

Figure 1.2: Example of PortCost and BridgeID assignment and resulting spanning tree. Node 1 is chosen as the Root Bridge. Even though path {1,2,4} and {1,3,4} have both length 8, the first is chosen because the BridgeID of node 2 is smaller than of node 3.

In graph theory, a spanning tree of a graph G is a connected, acyclic subgraph that spans all the nodes of G. As a consequence, a spanning tree of G contains only $n - 1$ edges, where n is the number of nodes in G. Hence, one of the drawbacks of the STP is that the network is only able to use a small portion of the existing links.

1.1.3 Multiple Spanning Tree Protocol

The IEEE 802.1q standard [80206] enables large SENs to be partitioned into multiple, smaller virtual LANs (VLANs), simplifying the network design. This allows for the isolation of different applications and/or data center customers, as two nodes belonging to a given VLAN can only communicate between each other, through the links established for the same VLAN.

The Multiple Spanning Tree Protocol (MSTP), standardized as 802.1s [80202], allows for service providers to install different spanning trees (one per VLAN) over a single physical topology. In this way, the network can make use of a larger number of links to send traffic, while satisfying the STP.

In this thesis, we study network design problems involving the MSTP. Note that we do not consider the assignment of the BridgeIDs and PortCosts to the switches. It is possible to do this *a posteriori*, such that the spanning trees selected in the optimization are implemented by the MSTP [dSS07]. Nonetheless, the optimization problems alone are highly combinatorial, as the number of potential designs can be huge for large networks. In practice, today's implementations of the MSTP circumvent this, by computing a small number of spanning trees, and mapping the VLAN onto them. Often, when many VLANs are defined, a spanning tree is generated, not for each single VLAN, but for a whole subset of them [Ho12]. In all likelihood, these implementation result in designs that do not make the best use of the network resources. Many methods have been proposed in the literature to improve the implementation of the MSTP, with respect to different traffic-oriented measures. In the next section, we present an overview of such methods.

Extensions to this protocol, better suited for traffic management, have been proposed in the engineering literature (see e.g. [KSI04, Med08]), but these are outside the scope of this thesis and could be a topic for further research. Nevertheless, we think the MSTPtechnology is worth studying as it is in use in many networks today, and it raises challenging problems for optimizers.

Other technologies like Software Defined Networks (SDNs) might be better suited for large data centers. However, SDNs are much more complex to manage and the commercially available solutions are expensive and quite limited in their traffic engineering capabilities. On the other hand, switched Ethernet equipment is a mature technology, it is quite inexpensive and its management requires less expertise human resources (which also has an impact on the operational costs).

1.2 Review of methods for the MSTP

Ho [Ho12] wrote an extensive review on several approaches from the literature, regarding TE problems for SENs implementing the MSTP.

In a first approach, [HZC06, LYD+03, Med06] proposed optimization techniques, that map a set of VLANs to a given number of spanning trees. Meddeb [Med06] developed an algorithm to generate a set of spanning trees with a small number of links in common, and then introduced another greedy algorithm to map each VLAN to those spanning trees, while attempting to minimize the number of used links. Lim *et al.* [LYD+03] proposed a Quality of Service (QoS)-aware mechanism that maps VLANs, with the objective of minimizing network load and delay. He *et al.* [HZC06] used an admission control algorithm to assign a group of VLANs to each given spanning tree, and then map each service to a VLAN, such that it minimizes the link load.

Sousa *et al.* [dSS07] and Santos *et al.* [SdSA+09, SdSA+10], introduced heuristic schemes, that aim to balance the load in networks using the MSTP. In [SdSA+09, SdSA+10], the heuristic procedure solves relaxed mixed-integer linear programs (MIPs), in order to obtain feasible solutions and lower bounds. Different criteria were taken into consideration, including service disruption and network load balancing.

Chen *et al.* [CJZ06] proposed an algorithm that designs a spanning tree for every source node in the network, while trying to achieve a good trade-off between load balance and average delay.

One last approach was suggested by Padmaraj *et al.* [PNM+05] and Mirjalily *et al.* [MSS09]. In this approach, costs are assigned to the links in the network, and the "cheapest" spanning trees are selected. In the first paper, the proposed heuristic updates weights assigned to the links in the network, in order to find a set of spanning trees with a good load balancing. In the second one, the suggested algorithm tries to find the best set of edge-disjoint spanning trees, and the best mapping of VLANs to that set.

Ho [Ho12] argued that all these proposals were not applicable for large networks. Hence, he proposed a local search based algorithm that aims at minimizing the worst-case link utilization for data center networks.

In Section 1.5, we review the existing literature on the use of mixed-integer programming (MIP) formulations to models problems dealing with the MSTP.

1.3 Notation and definitions

A SEN can be represented by a graph, where the nodes typically represent computers or switches, and the edges represent the bi-directional links connecting the latter. In this section, we introduce some concepts of graph theory and combinatorial optimization, that will be used throughout this thesis.

Consider an undirected graph $G = (N, E)$, where N is the set of nodes, with size n, and E the set of edges, with size m. Edge $e = \{i, j\} \in E$ represents an undirected link between the two end nodes, $i \in N$ and $j \in N$. Given a set $W \subset N$, $\delta(W) = \{\{i, j\} \in E : i \in W, j \in N \backslash W\}$ denotes the cut induced by W. The degree of a node v is defined as the cardinality of $\delta(v)$, where $\delta(v) = \delta(\{v\})$. Similarly, let $W_0, W_1, ..., W_k$ be a family of disjoint non-empty sets of nodes, whose union is N. A multicut is a subset of E containing edges with an end node in one of these sets of nodes, and the other end node in another; it is denoted by $\delta(W_0, W_1, ..., W_k)$. We also define the set $E(W) = \{\{i, j\} \in E : i, j \in W\}$ as the set of edges having both end nodes in W.

A sequence of edges, connecting a sequence of nodes is called a path. The first and last node in the sequence are named the end nodes. A cycle is defined as a path, that starts and ends at the same node. A graph that has a path between each pair of nodes is a connected graph.

Consider as well the set of arcs $A = \{(i, j), (j, i) : \{i, j\} \in E\}$. The graph $G' = (N, A)$ is the *directed* version of graph G. For such a graph, and for a given set $W \subset N$, we can define the two following cuts: $\delta^-(W) = \{(i, j) \in A : i \in N \backslash W, j \in W\}$ and $\delta^+(W) = \{(j, i) \in A : i \in N \backslash W, j \in W\}$. The cardinality of $\delta^-(v)$ and $\delta^+(v)$ is respectively named the indegree and outdegree of node v.

For simplicity, in many situations, we use the same designation for concepts defined in the directed graph as for the equivalent concepts defined in the in undirected graph, *e.g.* "paths", "cycles". This is not the case, however, for spanning trees and arborescences. A spanning tree of G is a connected subgraph, that includes all the nodes in N and contains exactly $n - 1$ edges. Consequently, a spanning tree is also acyclic, in the sense that it contains no cycles. An r-arborescence is a subset of A, such that there is no arc entering the root node r, and there is an unique path between r and every other node in N.

In this thesis, we propose several MIP formulations. In order to keep this thesis consistent, we try to assign the same notation to variables that have similar meanings, for different problems and/or formulations. As such, the formulations presented in the surveys of Sections 1.4 and 1.5 may not appear in the notation they were presented on, in their original papers. We associate the notation:

- w with undirected design variables;
- y with directed design variables;
- z^r with directed design variables for r-arborescences;

- x with directed flow variables, that define a path between between two nodes;
- l with variables that indicate the load on an arc/edge.

Moreover, letters of the greek alphabet will be assigned to variables of dual formulations (see below). Note, however, that there may occur exceptions to this notation, either when formulations require a lot of notation (*e.g.* in Section 1.5), or when referring to problems that are not network design problems (*e.g.* in Section 2.1).

Let us consider a MIP formulation F, where the goal is to minimize a given objective function. Consider the linear programming (LP) relaxation of F, where the integrality of the variables are relaxed, and which we denote by F_{LP}. We denote as $B_{LP}(F)$ as the bounds provided by solving F_{LP}, and as Gap_{LP} the ratio $\frac{B^* - B_{LP}(F)}{B^*}$, where B^* is the value/cost of the best known primal solution. We also denote $\mathcal{P}_F$ as the polyhedron defined by the set of feasible solutions of F_{LP}. The strength of the LP relaxation of two different MIP formulations with the same objective function, F^1 and F^2, can then be compared by examining their respective polyhedra, $\mathcal{P}_{F^1}$ and $\mathcal{P}_{F^2}$. The LP relaxation of F^1 is as strong as (stronger than) F_2 if $\mathcal{P}_{F^1} \subseteq \mathcal{P}_{F^2}$ ($\mathcal{P}_{F^1} \subset \mathcal{P}_{F^2}$). They are regarded as equivalent, if $\mathcal{P}_{F^1} = \mathcal{P}_{F^2}$.

We consider as well the concept of projection, which provides a connection between different formulations. Given a polyhedron $Q = \{(u, x) \in \mathbb{R}^p \times \mathbb{R}^q : Au + Bx \leq b\}$, the projection of Q onto $\mathbb{R}^q$, or onto the x-space, is defined as $Proj_x(Q) = \{x \in \mathbb{R}^q : \exists u \in \mathbb{R}^p : (u, x) \in Q\}$. This is useful, as it allows us to compare formulations, as seen in the previous paragraph, but in different variable spaces.

Lastly, we recall some basic notions of duality theory. Let us redefine the LP formulation F_{LP} as $\min\{z = cx : Ax \geq b, x \geq 0\}$, where z is the cost of the solution, A are the constraints' coefficients, b the right-hand sides and c objective function coefficients. Then, we denote as the dual of F_{LP} to the LP formulation defined as $\max\{\omega = \pi b : \pi A \leq c, \pi \geq 0\}$, where π are the dual variables. We refer to this dual formulation as F_D. Dual formulations have an important property, in that the value of the optimal solution of F_{LP}, is the same of the optimal solution of F_D: $z^* = \omega^*$. This theory will be important when we introduce the Benders' decomposition algorithm, in Section 1.6.

1.4 MIPs for problems with spanning trees

Spanning tree optimization problems are numerous, and many papers have been published to study these special structures. By allowing the connection of all nodes in a network, while eliminating edge redundancy, spanning trees naturally arise in a large number of applications, such as computer networking, energy distribution, facility location, manufacturing and telecommunications. Moreover, as the spanning tree is the simplest type of network design model, its structure appears regularly embedded in other problems. As such, the study of these trees can prove to be valuable for other network design problems [MW95].

Although the amount of optimization problems dealing with spanning trees is indeed vast, their corresponding mathematical programming formulations can all be traced back to the ones proposed for a few fundamental problems. Therefore, the way the latter have been modeled is an essential first clue, when looking into a new spanning tree optimization problem. In this section, we give an overview of MIP formulations that have been proposed for two of these basic problems: the minimum spanning tree (MST) problem, and the optimum communication spanning tree (OCST) problem.

1.4.1 Minimum spanning tree problem

In the MST problem, the objective is to find a tree that spans all the nodes of a given graph, such that the sum of the weights/costs of the edges in the tree is minimized. This simple problem does not impose any restrictions on the topology of the tree, nor on the capacity of the links. This is one of the most well-known problems in combinatorial optimization, with studies dating back to 1926 [NMN01]. Many greedy algorithms have been proposed, with the most famous being Kruskal's [Kru56] and Prim's [Pri57]. These algorithms are able to achieve the optimal tree in polynomial time. For our purposes, we are exclusively interested in exploring the different mathematical programming formulations that have been proposed for the MST problem [MW95].

Let c_e be the cost of selecting edge $e \in E$ for the spanning tree. We denote as w_e, the set of binary variables which have value 1 if edge $e \in E$ is selected as part of the spanning tree, and 0 otherwise. In this thesis, this type of variables will also be referred to as "design variables".

The first MIP formulation, 1.1, is named after the "packing" constraints in (1.1b). These constraints ensure that the set of selected edges is acyclic. Along with the cardinality constraint of (1.1c), they imply that the chosen edges form a spanning tree. Note that constraints (1.1d) define the design variables w as 0-1. Consider, however, the LP relaxation of the packing formulation where the integrality of these constraints is relaxed, and we have $w_e \in [0, 1]$ instead. This relaxed formulation has an important property: it describes the spanning tree polytope, *i.e.* the convex hull of characteristic vectors of spanning trees. Consequently, for any instance of the MST problem, the optimal solution of the LP relaxation of the packing formulation is also integer.

Formulation 1.1 regards spanning tree as acyclic subgraphs with $n - 1$ edges. Alternatively, a spanning tree can be seen as a connected subgraph with $n - 1$ edges. The

$$\min_{w} \sum_{e \in E} c_e w_e \tag{1.1a}$$

s.t

$$\sum_{e \in E(W)} w_e \leq |W| - 1, \qquad W \subseteq N : 2 \leq |W| \leq n-1 \tag{1.1b}$$

$$\sum_{e \in E} w_e = n - 1 \tag{1.1c}$$

$$w_e \in \{0, 1\}, \qquad e \in E \tag{1.1d}$$

Formulation 1.1: MST problem: packing formulation.

cutset Formulation 1.2, substitutes the "packing" constraints with constraints (1.2a), which imply that there is at least one edge connecting the nodes of any subset of N, and all the others.

Let $\mathcal{P}_{sub}$ and $\mathcal{P}_{cut}$ be the polyhedra defined by the LP relaxations of, respectively, the packing and the cutset formulations. Then, $\mathcal{P}_{sub} \subset \mathcal{P}_{cut}$. This serves as a motivation to look at what is "missing" in the cutset formulation. The multicut Formulation 1.3 extends the cutset formulation, by connecting any partition of N onto i sets, for $2 \leq i \leq n-1$ (1.3a). The polyhedron P_{mcut} defined by the LP relaxation of the multicut formulation is such that $P_{mcut} = P_{sub}$.

Formulations 1.1, 1.2 and 1.3 are "natural" formulations, as they only use the "natural" design variables. Nevertheless, these formulations can be very large, as the number of constraints (1.1c), (1.2a) and (1.3a) is exponential, with respect to n. This motivates the creation of extended formulations, that ensure connectivity by modelling flows, that need to sent between the nodes of the network. In this context, the design variables w_e are redefined to indicate whether or not edge e is selected, so that it can carry any flow. Note that, in flow models, although the edges continue to be undirected, the flow

$$\min_{w} \sum_{e \in E} c_e w_e \tag{1.1a}$$

s.t

$$\sum_{e \in \delta(W)} w_e \geq 1, \qquad W \subseteq N : 2 \leq \mid W \mid \leq n-1 \tag{1.2a}$$

$$\sum_{e \in E} w_e = n - 1 \tag{1.1c}$$

$$w_e \in \{0, 1\}, \qquad e \in E \tag{1.1d}$$

Formulation 1.2: MST problem: cutset formulation.

$$\min_{w} \sum_{e \in E} c_e w_e \tag{1.1a}$$

s.t

$$\sum_{e \in \delta(W_0, W_1, ..., W_i)} w_e \geq i, \qquad W_0, W_1, ..., W_i = N : 2 \leq i \leq n-1 \tag{1.3a}$$

$$\sum_{e \in E} w_e = n - 1 \tag{1.1c}$$

$$w_e \in \{0, 1\}, \qquad e \in E \tag{1.1d}$$

Formulation 1.3: MST problem: multicut formulation.

variables are typically directed, and consequently defined in G'.

There are different flow formulations for the MST problem, and one way to categorize them is either as single commodity or as multicommodity. In the single commodity case, one node (typically node 1) is chosen as the root, from where we must send one unit of flow to every other node. We define variables x_{ij}, that measure the quantity of flow, originated at the root node, traversing arc $(i, j) \in A$. The single commodity model is formulated as seen in Formulation 1.4. Constraints (1.4a-1.4b) ensure the conservation of flow from the root node to every other node, while (1.4c-1.4d) ensure that the flow only travels through the selected edges.

$$\min_{w,x} \sum_{e \in E} c_e w_e \tag{1.1a}$$

s.t

$$\sum_{a \in \delta^+(1)} x_a = n - 1 \tag{1.4a}$$

$$\sum_{a \in \delta^-(i)} x_a - \sum_{a \in \delta^+(i)} x_a = 1, \qquad i \in N : i \neq 1 \tag{1.4b}$$

$$x_{ij} \leq (n-1) \cdot w_e, \qquad e = \{i, j\} \in E \tag{1.4c}$$

$$x_{ji} \leq (n-1) \cdot w_e, \qquad e = \{i, j\} \in E \tag{1.4d}$$

$$\sum_{e \in E} w_e = n - 1 \tag{1.1c}$$

$$w_e \in \{0, 1\}, \qquad e \in E \tag{1.1d}$$

$$x_{ij} \geq 0, \qquad (i, j) \in A \tag{1.4e}$$

Formulation 1.4: MST problem: single commodity flow formulation.

This model is "compact", in the sense that the number of constraints (in addition

to the number of variables) is polynomial, and thus, smaller than for the "natural" formulations. Nevertheless, comparing with the packing and multicut formulations, the single commodity model is weaker, in the sense that the polyhedron defined by its LP relaxation ($\mathcal{P}_{flo}$) can be quite a poor representation of the integer program. Consider the projection of $\mathcal{P}_{flo}$ onto the w-space, defined as seen in Section 1.3. Then $\mathcal{P}_{cut} \subseteq Proj_w(\mathcal{P}_{flo})$.

In the multicommodity flow case, the flow variables x are disaggregated, such that they also contain information about the destination of each flow. As such, we redefine variables x_a^v that measure the quantity of flow traversing arc $a \in A$, originated at the root node and destined to node $v \in N$. The formulation seen in 1.5 is denoted as the undirected multicommodity flow model.

$$\min_{w,x} \sum_{e \in E} c_e w_e \tag{1.1a}$$

s.t

$$\sum_{a \in \delta^+(1)} x_a^v = 1, \qquad v \in N : v \neq 1 \tag{1.5a}$$

$$\sum_{a \in \delta^-(i)} x_a^v - \sum_{a \in \delta^+(i)} x_a^v = 0, \qquad v, i \in N : i \neq \{1, v\}, v \neq 1 \tag{1.5b}$$

$$\sum_{a \in \delta^-(v)} x_a^v = 1, \qquad v \in N : v \neq 1 \tag{1.5c}$$

$$x_{ij}^v \leq w_e, \qquad v, i, j \in N : e = \{i, j\} \in E, v \neq 1 \tag{1.5d}$$

$$\sum_{e \in E} w_e = n - 1 \tag{1.1c}$$

$$w_e \in \{0, 1\}, \qquad e \in E \tag{1.1d}$$

$$x_{ij}^v \geq 0, \qquad v \in N : v \neq 1, (i, j) \in A \tag{1.5e}$$

Formulation 1.5: MST problem: undirected multicommodity flow formulation.

Constraints (1.5a-1.5c) imply that one unit is sent from the root node to every other node. Naturally, the flow can only be carried by the edges selected by the design variables (1.5d). Let $\mathcal{P}_{mcflo}$ denote the polyhedron defined by the LP relaxation of Formulation 1.5, and $Proj_w(\mathcal{P}_{mcflo})$ represent the projection of $\mathcal{P}_{mcflo}$ onto the w-space. Although $Proj_w(\mathcal{P}_{mcflo})$ is as strong as $Proj_w(\mathcal{P}_{flo})$ ($Proj_w(\mathcal{P}_{mcflo}) \subseteq Proj_w(\mathcal{P}_{flo})$), it is still not equivalent to $\mathcal{P}_{sub}$ and $\mathcal{P}_{mcut}$. In order to strengthen this Formulation 1.5, it is also possible to consider design variables that are defined in the directed network, G'. Let y_a be 1 if arc $a \in A$ is selected to carry flow, and 0 otherwise. The directed multicommodity flow model is formulated as shown in 1.6. Note that constraints (1.6c) imply that an edge can only carry flow originated at the root, in one of the two directions.

We denote as $\mathcal{P}_{dflo}$ the polyhedron defined by the LP relaxation of Formulation 1.6,

$$\min_{w,x,y} \sum_{e \in E} c_e w_e \tag{1.1a}$$

s.t

$$\sum_{a \in \delta^+(1)} x_a^v = 1, \qquad v \in N : v \neq 1 \tag{1.5a}$$

$$\sum_{a \in \delta^-(i)} x_a^v - \sum_{j \in \delta^+(i)} x_a^v = 0, \qquad v, i \in N : i \neq \{1, v\}, v \neq 1 \tag{1.5b}$$

$$\sum_{a \in \delta^-(v)} x_a^v = 1, \qquad v \in N : v \neq 1 \tag{1.5c}$$

$$x_{ij}^v \leq y_{ij}, \qquad v \in N : v \neq 1, (i,j) \in A \tag{1.6a}$$

$$\sum_{a \in A} y_a = n - 1 \tag{1.6b}$$

$$y_{ij} + y_{ji} = w_e, \qquad e = \{i,j\} \in E \tag{1.6c}$$

$$y_a \in \{0,1\}, \qquad a \in A \tag{1.6d}$$

$$w_e \in \{0,1\}, \qquad e \in E \tag{1.1d}$$

$$x_{ij}^v \geq 0, \qquad v \in N : v \neq 1, (i,j) \in A \tag{1.5e}$$

Formulation 1.6: MST problem: directed multicommodity flow formulation.

and $Proj_w(\mathcal{P}_{dflo})$ as the projection of $\mathcal{P}_{dflo}$ onto the w-space. Then, $Proj_w(\mathcal{P}_{dflo}) = \mathcal{P}_{sub} = \mathcal{P}_{mcut}$.

Alternatively, it is possible to obtain yet another multicommodity flow formulation equivalent to Formulation 1.6, without using directed design variables. Formulation 1.7 is designated as the extended multicommodity flow formulation, and the corresponding polyhedron of feasible solutions of the LP relaxation as $\mathcal{P}_{mc'flo}$. As mentioned above, an edge can only carry flow originated at the root, in one of the two directions. Hence, it is possible to replace constraints (1.5d) by the tighter constraints (1.7a).

A similar extended formulation was proposed by Kipp Martin in [Mar91]. This was, in fact, the first polynomial-sized formulation proposed to describe the spanning tree polytope. Consider variables z_{ij}^u, that indicate the quantity of flow with origin in node $u \in N$ and destination in node $j \in N$ and travelling through arc $(i,j) \in A$. The MST problem can be formulated as seen in 1.8. Note that in his original paper, Kipp Martin defined variables z_{ij}^u in a slightly different way: they indicate instead the quantity of flow with origin in $i \in N$ and destination in $u \in N$, and traversing arc $(i,j) \in A$. We have changed the formulation here, for the sake of the thesis' consistency. Constraints (1.8a) state that the flow entering node $j \in N$, cannot surpass 1, whereas constraints (1.8b) imply that no flow can enter the origin node u. The set of constraints (1.8c) implies that we can only have flow originated at node u travelling in one direction of any given edge $e \in E$; and the quantity of flow is precisely the value of w_e.

$$\min_{w,x} \sum_{e \in E} c_e w_e \tag{1.1a}$$

s.t

$$\sum_{a \in \delta^+(1)} x_a^v = 1, \qquad v \in N : v \neq 1 \tag{1.5a}$$

$$\sum_{a \in \delta^-(i)} x_a^v - \sum_{a \in \delta^+(i)} x_a^v = 0, \qquad v, i \in N : i \neq \{1, v\}, v \neq 1 \tag{1.5b}$$

$$\sum_{a \in \delta^-(v)} x_a^v = 1, \qquad v \in N : v \neq 1 \tag{1.5c}$$

$$x_{ij}^v + x_{ji}^{v'} \leq w_e, \qquad v, v', i, j \in N : e = \{i, j\} \in E, v \neq \{1, v'\} \tag{1.7a}$$

$$\sum_{e \in E} w_e = n - 1 \tag{1.1c}$$

$$w_e \in \{0, 1\}, \qquad e \in E \tag{1.1d}$$

$$x_{ij}^v \geq 0, \qquad v \in N : v \neq 1, (i, j) \in A \tag{1.5e}$$

Formulation 1.7: MST problem: extended multicommodity flow formulation.

$$\min_{w,z} \sum_{e \in E} c_e w_e \tag{1.1a}$$

s.t

$$\sum_{a \in \delta^-(j)} z_a^u \leq 1, \qquad u, j \in N : i \neq u \tag{1.8a}$$

$$\sum_{a \in \delta^-(u)} z_a^u \leq 0, \qquad u \in N \tag{1.8b}$$

$$z_{ij}^u + z_{ji}^u = w_e, \qquad e = \{i, j\} \in E \tag{1.8c}$$

$$\sum_{e \in E} w_e = n - 1 \tag{1.1c}$$

$$w_e \in \{0, 1\}, \qquad e \in E \tag{1.1d}$$

$$z_a^u \geq 0, \qquad u \in N, a \in A \tag{1.8d}$$

Formulation 1.8: MST problem: Kipp Martin's formulation.

Formulation 1.8 can also be understood in a different way. Recall the notion of arborescence, introduced in the previous section. We can perceive variables z_a^u instead as directed design variables, which indicate whether or not arc $a \in A$ is selected for an arborescence rooted at node $u \in N$. Thus, the constraints in Formulation 1.8 imply that

there is an u-arborescence for every node $u \in N$, and that they all use the same edges (1.8c). Each u-arborescence is designed by indicating that there is an arc entering every node (1.8a), with the exception of the root u (1.8b).

Let $Proj_w(\mathcal{P}_{arb})$ be the projection of the polyhedron defined by the LP relaxation of Formulation 1.8 onto the w-space. Figure 1.3 details the relationship between the polyhedra defined by the LP relaxations of each model. All the polyhedra in the left-most set describe the spanning tree polytope.

$$\left\{ \begin{array}{c} \mathcal{P}_{sub} \\ \mathcal{P}_{mcut} \\ Proj_w(\mathcal{P}_{dflo}) \\ Proj_w(\mathcal{P}_{mc'flo}) \\ Proj_w(\mathcal{P}_{arb}) \end{array} \right\} \subseteq \left\{ \begin{array}{c} \mathcal{P}_{cut} \\ Proj_w(\mathcal{P}_{mcflo}) \end{array} \right\} \subseteq Proj_w(\mathcal{P}_{flo})$$

Figure 1.3: Relationship between underlying polyhedra.

As it was implied in the beginning of this section, the MST problem is probably the most fundamental of all problem dealing with spanning trees. As such, the formulations in Figure 1.3 are of great important, as they are the base for modelling the many other, more complicated, spanning tree problems.

1.4.2 Optimum communication spanning tree problem

The optimum (optimal) communication spanning tree (OCST) problem was first introduced by Hu [Hu74]. It is a particular extension of the minimum spanning tree problem, where there is a set of communications' traffic demands between the nodes of the network, that have to be routed in a spanning tree. The main novelty of the OCST problem is in the cost of each edge, which is linearly dependent on its load (*i.e.* the total traffic flowing through it). Despite of its apparent simplicity, this is actually a very challenging problem, belonging to the class of $\mathcal{NP}$-hard problems [JLK78].

Consider the set of traffic commodities K, each $k \in K$ with a given origin o_k, destination d_k, and traffic demand ρ_k. Alternatively, we designate as $\rho_{uv} = \sum_{k \in K: o_k = u, d_k = v} \rho_k$. Without loss of generality we assume that $u < v$, and redefine ρ_{uv} as $\rho_{uv} + \rho_{vu}$.

Although the problem was originally proposed in 1974, to the best of our knowledge, it was only recently that it was first modeled as a MIP [Rot08]. For this first formulation, Rothlauf chose to formulate the problem with multicommodity flows (see Section 1.4.1). As mentioned above, in the OCST problem there are multiple sources of traffic flow. Rothlauf defines flow variables x_e^{uv}, that indicate whether or not edge $e \in E$ is in the path between $u \in N$ and $v \in N$, such that $u < v$. Note that the author chooses to define these variables on the undirected space, which motivates a new set of variables h_i^{uv}, that indicate whether node $i \in N$ is on the path between $u \in N$ and $v \in N$ ($u < v, i \notin \{u, v\}$). Rothlauf models the OCST problem as Formulation 1.9.

Constraints (1.9b-1.9c) define the spanning tree, while (1.9f-1.9h) define the path between each pair of nodes. Note that this is done in a different way than in Formulations

$$\min_{h,w,x} \sum_{u,v\in N:u<v} \rho_{uv} \sum_{e\in E} c_e x_e^{uv} \tag{1.9a}$$

s.t

$$\sum_{e\in\delta(i)} w_e \geq 1, \qquad i \in N \tag{1.9b}$$

$$\sum_{e\in E} w_e = n-1 \tag{1.9c}$$

$$\sum_{u,v\in N:u<v} x_e^{uv} \geq w_e, \qquad e \in E \tag{1.9d}$$

$$\sum_{u,v\in N:u<v} x_e^{uv} \leq M w_e, \qquad e \in E \tag{1.9e}$$

$$\sum_{e\in\delta(u)} x_e^{uv} = 1, \qquad u,v \in N : u < v \tag{1.9f}$$

$$\sum_{e\in\delta(v)} x_e^{uv} = 1, \qquad u,v \in N : u < v \tag{1.9g}$$

$$\sum_{e\in\delta(i)} x_e^{uv} = 2h_i^{uv}, \qquad i,u,v \in N : u < v, i \notin \{u,v\} \tag{1.9h}$$

$$w_e \in \{0,1\}, \qquad e \in E \tag{1.9i}$$

$$h_i^{uv} \in \{0,1\}, \qquad i,u,v \in N : u < v, i \notin \{u,v\} \tag{1.9j}$$

$$x_e^{uv} \in \{0,1\}, \qquad u,v \in N : u < v, e \in E \tag{1.9k}$$

Formulation 1.9: OCST problem: Rothlauf's formulation.

1.5 and 1.6; this is due to Rothlauf's choice of using undirected flow variables x. In (1.9d-1.9e) the design and flow variables are associated. In (1.9e) a *big-M* strategy is used, which is typically associated with weak LP relaxations.

Later, Contreras [Con09] presented three formulations for the same problem. The first model is also a flow Formulation (1.10). Notwithstanding, in this formulation there are no flow variables, as defined for Formulations 1.4-1.9. While for those formulations, flows were used exclusively to ensure the connectivity of the spanning trees, for the OCST problem it is necessary to know the load flowing through each edge/arc. Accordingly, Contreras defines continuous variables l_a^u that measure the load in arc $a \in A$, and originated in $u \in N$.

The LP relaxation of Formulation 1.10 yields poor lower bounds and, therefore, the same author proposed two other formulations, which he designates as "path-based". The first of these formulation is, in fact, a multicommodity flow formulation, similar to the one in [Rot08], but with the flow variables x being directed. The second "path-based" Formulation 1.11 requires to generate, *a priori*, all the possible paths between each pair of nodes, $P_{uv} : u,v \in N, u < v$. Variables p_{uv}^q indicate whether path $q \in P_{uv}$ is chosen

$$\min_{l,w} \sum_{u \in N, a \in A} c_a l_a^u \tag{1.10a}$$

s.t

$$\sum_{a \in \delta^-(i)} l_a^u - \sum_{a \in \delta^+(i)} l_{ij}^u = \rho_{ui}, \qquad u, i \in N : u \neq i \tag{1.10b}$$

$$l_{ij}^u + l_{ji}^u \leq \sum_{v \in N : u < v} \rho_{uv} \cdot w_e^s, \qquad u \in N, e = \{i, j\} \in E \tag{1.10c}$$

$$\sum_{e \in E} w_e = n - 1 \tag{1.9c}$$

$$w_e \in \{0, 1\}, \qquad e \in E \tag{1.9i}$$

$$l_a^{us} \geq 0, \qquad u \in N, a = (i, j) \in A : j \neq u \tag{1.10d}$$

Formulation 1.10: OCST problem: Contreras' flow formulation.

to connect nodes $u \in N$ and $v \in N$.

The set of constraints (1.11b) implies that for each pair of nodes, only one path can be chosen. Constraints (1.11c) state that these paths must use the edges selected for the spanning tree. Contreras showed that the two "path-based" formulations are equivalent. In [CFM10] tests were made using Formulation 1.10 against benchmark instances, namely to measure the Gap_{LP}. These tests revealed that for small-sized instances, the Gap_{LP} is close to 0%, and that for medium-sized ones, the Gap_{LP} does not exceed 15%.

In [FLMH$^+$13], another flow formulation is proposed. Consider the set of variables w_e, l_a^u and z_a^u, defined as before. The OCST problem can be formulated as detailed in 1.12. This model builds on the Formulation 1.10, by adding "arborescence design" variables z, like the ones used in Kipp Martin's Formulation 1.8 for the MST problem.

While the two "path-based" models in [Con09] had variables with up to 4 indexes, this model only requires 3 indexes variables. However, although no comparison is made with the previous models, the authors seem to suggest that this Formulation 1.12 is weaker than the aforementioned path models 1.10 and 1.11. In order to make the LP relaxation of the model tighter, the authors propose several families of valid inequalities, namely: vertex cutset inequalities, set cutset inequalities, min-cut values inequalities and inequalities that ensure the minimum flows through arcs. They report that, according to the numerical tests, the ones with a greater impact are the min-cut values inequalities.

The optimum communication spanning tree problem is an important foundation for the problems we study, as it introduces the concept of traffic demands.

$$\min_{p,w} \sum_{u,v\in N:u<v} \rho_{uv} \sum_{q\in P_{uv}} \sum_{e\in q} c_e p^q_{uv} \tag{1.11a}$$

s.t

$$\sum_{q\in P_{uv}} p^q_{uv} = 1, \qquad u,v \in N : u < v \tag{1.11b}$$

$$\sum_{q\in P_{uv}:e\in q} p^q_{uv} \leq w_e, \qquad u,v \in N : u < v, e \in E \tag{1.11c}$$

$$p^q_{uv} \in \{0,1\}, \qquad u,v \in N, q \in P_{uv} \tag{1.11d}$$

$$\sum_{e\in E} w_e = n-1 \tag{1.9c}$$

$$w_e \in \{0,1\}, \qquad e \in E \tag{1.9i}$$

(1.11e)

Formulation 1.11: OCST problem: Contreras' path formulation.

$$\min_{l,x,w} \sum_{u\in N,a\in A} c_a l^u_a \tag{1.10a}$$

s.t

$$\sum_{a\in A:a\notin\delta^-(u)} z^u_a = n-1, \qquad u \in N \tag{1.12a}$$

$$l^u_a \leq M z^u_a, \qquad u \in N, a \in A : a \notin \delta^-(u) \tag{1.12b}$$

$$\sum_{j:(j,i)\in A} l^u_{ji} - \sum_{j:(i,j)\in A, j\neq u} l^u_{ij} = \rho_{ui}, \qquad u,i \in N : u \neq i \tag{1.10b}$$

$$l^u_{ij} + l^u_{ji} \leq \sum_{v\in N:u<v} \rho_{uv} \cdot w^s_e, \qquad u \in N, e = \{i,j\} \in E \tag{1.10c}$$

$$\sum_{e\in E} w_e = n-1 \tag{1.9c}$$

$$w_e \in \{0,1\}, \qquad e \in E \tag{1.9i}$$

$$l^{us}_a \geq 0, \qquad u \in N, a \in A : j \neq u \tag{1.10d}$$

$$z^u_a \geq 0, \qquad u \in N, a \in A : a \notin \delta^-(u) \tag{1.12c}$$

Formulation 1.12: OCST problem: Fernandez's flow formulation.

1.5 MIPs for problems with MSTP

As it was expressed in Section 1.2, the literature dealing with the optimization of MSTP is extensive. Notwithstanding, due to the great complexity of computing many spanning trees over a single physical network, research has been mainly focused on heuristic approaches. As far as we know, the only works that have studied problems dealing with the MSTP, by using exact methods are the ones reviewed in this section.

In [CKM05], SENs using the MSTP are optimized with respect to QoS, while ensuring network protection, in case of link failure. For the proposed problem, the authors divide the set of nodes N, in three disjunct subsets: the access nodes, N_A; the edge nodes, N_E; and the bridge nodes, N_B. Recall the set of commodities K, introduced in the previous section, for the OCST problem. In [CKM05], each $k \in K$ is assumed to be such that $o_k \in N_A$ and $d_k \in N_E$. Moreover, set K is partitioned in four subsets ($K = K^1 \bigcup K^2 \bigcup K^3 \bigcup K^4$), representing different QoS classes, with different requirements. The first class entails the commodities whose traffic has the highest priority; as such, commodities in this class are assigned the highest weight in the objective function ($c^k = 4, k \in K^1$). The three other classes are assigned decreasing weights: 3, 2 and 1, respectively. Let C_e define the bandwidth capacity of each edge $e \in E$, bounding the traffic volume in both directions. The first class is also assumed to require the lowest bandwidth volume, and so its load can use at most 10% of the link's capacity. The three other classes are assigned increasing limits: 20%, 30% and 40%, respectively.

The authors also define a set of trees T, that can have up to 64 elements. Each tree can only be assigned one edge node, that is regarded as the root, whereas the leaves can only be access nodes. In order to ensure the protection of the network, the authors define for each demand $k \in K$, both a working tree in $T_w(k) \subset T$ and a protection tree in $T_p(k) \subset T$. Even though Cinkler *et al.* consider that most of commodities need to be protected, not all have that need. In that sense, they define a binary parameter π^k, which is 1 if and only if commodity k needs to be protected. With multiple spanning trees, the design variables previously used, have to include one more index. Therefore, y_a^t indicates whether or not arc $a \in A$ is used by tree $t \in T$. Cinkler *et al.* also define the variables $\hat{y}_a^k$ and $\check{y}_a^k$, that indicate whether or not demand $k \in K$ uses arc $a \in A$ as, respectively, part of the working tree or the protection tree. The model is thus formulated as shown in 1.13. In [CMK+05] a very similar model is presented, without the concern for tree protection.

The objective is to minimize a weighted sum of the chosen resources in the network (1.13a). Constraints (1.13b-1.13f) bound the load for each class of commodities. For both the working and protection tree, (1.13g-1.13l) define a path between the origin and destination of each commodity. The set of constraints (1.13m-1.13n) imply that the working and protection trees for each commodity are edge-disjoint. The remaining constraints (1.13o-1.13t) are responsible for defining the trees for each commodity, such that only edge nodes may be the root, and access nodes the leaves.

The authors do not seem to be so much focused on optimizing large-sized instances, as on exploring the impact of different protection and design strategies on the QoS of toy

networks. Moreover, despite dealing with MSTP, the problems in [CKM05, CMK+05] clearly differ from the problems considered in the context of this thesis, where our objective is to minimize link congestion. Although they are also distinct, the problems in [SdSA+09, SdSA+10] are closer to the one we propose to study. In both papers, different objectives are regarded; nevertheless, the same set of constraints seen in 1.14 are enforced.

In their formulations, Santos *et al.* use five different sets of binary variables (z, w, x, $\hat{x}$, ϕ) and one continuous (μ). There are two sets of design variables: w_e^t are undirected variables that indicate whether edge $e \in E$ is used for tree $t \in T$; z_a^t design an arborescence for each $t \in T$. There are also two sets of flow variables: $\hat{x}_a^{ut}$ define a path between the root node r of arborescence $t \in T$, and node $u \in N$; and x_a^{kt} that define a path between the origin and destination of commodity $k \in K$, if that commodity is assigned to tree $t \in T$. Note that the authors do not pre-assign the commodities to the set of trees (VLANs); this is done via variables ϕ. Finally, μ_a measure the utilization of an edge. Observe the formulation in 1.14. Constraints (1.14a-1.14b) impose the connectivity of the arborescences. The arcs on the arborescence defined in (1.14c-1.14e) must coincide with the undirected trees (1.14f). Constraints (1.14h-1.14j) define a path between the origin and destination of each commodity, in the trees they are assigned to; naturally, each commodity must be assigned to one and only one tree (1.14g). Finally, the edge utilization is calculated in (1.14l).

Using this set of constraints, Santos *et al.* consider different objectives. In [SdSA+09], two load balancing problems are tackled. The first one is the minimization of the average edge utilization, with a guaranteed worst-case scenario: $\min \frac{1}{|E|} \sum_{e \in E} \mu_e$, subject to (1.14a-1.14r) and $\{\mu_e \leq \mu_{max}, e \in E\}$. The second objective is to minimize the worst-case edge utilization, with a guaranteed average value: $\min \mu_{max}$, subject to (1.14a-1.14r) and $\{\frac{1}{|E|} \sum_{e \in E} \mu_e \leq \mu_{avg}; \mu_e \leq \mu_{max}, e \in E\}$. Moreover, in both [SdSA+10] and [SdSA+09] another problem is proposed, dealing with a lexicographical minimization of the edge's utilization. In this lexicographical objective, the authors begin to minimize the worst-case edge utilization. Afterwards, that value is fixed, and the second largest edge utilization is minimized. The procedure continues in this fashion, for all edges. In [SdSA+10] another similar problem was proposed, that also considers the minimization of the average utilization.

Although both [SdSA+09, SdSA+10] propose MIP formulations for their problem, the focus of these works is clearly on heuristic procedures. In fact, Santos *et al.* claim, as Cinkler *et al.* had before, that the computational cost of using exact methods for these problems motivate the need for heuristic methods. In this sense, in [SdSA+09, SdSA+10], the proposed MIPs are relaxed and embedded in an heuristic procedure. Although the authors also suggest exact methods *per se*, these seem to be used only as a means of measuring the quality of the proposed heuristics.

Recently, in [LLL15] have proposed a novel protection scheme for the MSTP, with the purpose of quickly configuring a back-up VLAN in case of link failure. In his scheme, the authors require that the switches adjacent to a given protected link, must be leaf nodes if this link fails. The authors propose a MIP formulation for the problem of designing load-

balanced and resilient SENs implementing the MSTP. In this formulation, the authors suggest transforming the original graph onto a two-layer graph, such that the load-balanced working trees are set on the top layer, and the back-up trees on the bottom one. A back-up tree is selected for all failure scenarios, following the aforementioned protection scheme. We do not present this MIP here since, contrary to what happens in Formulations 1.13 and 1.14, the design of the spanning trees is not considered; trees are instead, provided as input. Lee *et al.* suggest that their proposed MIP is to large to solve. Therefore, they partition the problem in two phases: in the first phase, the heuristic algorithm proposed in [dSS07] is used, to design the load-balanced working trees; in the second phase, given the set of working trees, the original MIP can be decomposed onto smaller ones, where the protection for each failure scenario is ensured.

This review implies that, when it comes to the use of exact methods for the optimized implementation of the MSTP, there is still plenty of room for improvement. The objective of this thesis, is to develop more efficient models that, by themselves or through the use of high-end mathematical programming techniques (*e.g.* decomposition methods), can overcome the demanding computational costs mentioned in the works reviewed in this section.

$$\min_{y,\hat{y},\check{y}} \sum_{a\in A}\left[\alpha \sum_{t\in T} y_a^t + \frac{1-\alpha}{C_a}\sum_{k\in K} c^k(\hat{y}_a^k + \check{y}_a^k)\rho_k\right] \tag{1.13a}$$

s.t

$$\sum_{k\in K^1}(\hat{y}_{ij}^k + \hat{y}_{ji}^k + \check{y}_{ij}^k + \check{y}_{ji}^k) \le 0.1C_e, \qquad e=\{i,j\}\in E \tag{1.13b}$$

$$\sum_{k\in K^2}(\hat{y}_{ij}^k + \hat{y}_{ji}^k + \check{y}_{ij}^k + \check{y}_{ji}^k) \le 0.2C_e, \qquad e=\{i,j\}\in E \tag{1.13c}$$

$$\sum_{k\in K^3}(\hat{y}_{ij}^k + \hat{y}_{ji}^k + \check{y}_{ij}^k + \check{y}_{ji}^k) \le 0.3C_e, \qquad e=\{i,j\}\in E \tag{1.13d}$$

$$\sum_{k\in K^4}(\hat{y}_{ij}^k + \hat{y}_{ji}^k + \check{y}_{ij}^k + \check{y}_{ji}^k) \le 0.4C_e, \qquad e=\{i,j\}\in E \tag{1.13e}$$

$$\sum_{k\in K}(\hat{y}_{ij}^k + \hat{y}_{ji}^k + \check{y}_{ij}^k + \check{y}_{ji}^k) \le C_e, \qquad e=\{i,j\}\in E \tag{1.13f}$$

$$\sum_{a\in\delta^+(o_k)}\hat{y}_a^k - \sum_{a\in\delta^-(o_k)}\hat{y}_a^k = 1, \qquad k\in K \tag{1.13g}$$

$$\sum_{a\in\delta^+(i)}\hat{y}_a^k - \sum_{a\in\delta^-(i)}\hat{y}_a^k = 0, \qquad k\in K, i\in N\backslash\{o_k,d_k\} \tag{1.13h}$$

$$\sum_{a\in\delta^+(d_k)}\hat{y}_a^k - \sum_{a\in\delta^-(d_k)}\hat{y}_a^k = -1, \qquad k\in K \tag{1.13i}$$

$$\sum_{a\in\delta^+(o_k)}\check{y}_a^k - \sum_{a\in\delta^-(d_k)}\check{y}_a^k = \pi_k, \qquad k\in K \tag{1.13j}$$

$$\sum_{a\in\delta^+(i)}\check{y}_a^k - \sum_{a\in\delta^-(i)}\check{y}_a^k = 0, \qquad k\in K, i\in N\backslash\{o_k,d_k\} \tag{1.13k}$$

$$\sum_{a\in\delta^+(d_k)}\check{y}_a^k - \sum_{a\in\delta^-(d_k)}\check{y}_a^k = -\pi_k, \qquad k\in K \tag{1.13l}$$

$$\hat{y}_{ij}^k + \check{y}_{ij}^k \le 1, \qquad k\in K, \{i,j\}\in E \tag{1.13m}$$

$$\hat{y}_{ij}^k + \check{y}_{ji}^k \le 1, \qquad k\in K, \{i,j\}\in E \tag{1.13n}$$

$$\hat{y}_a^k \le y_a^t \qquad k\in K, a\in A, t\in T_w(k) \tag{1.13o}$$

$$\check{y}_a^k \le y_a^t \qquad k\in K, a\in A, t\in T_p(k) \tag{1.13p}$$

$$\sum_{a\in\delta^+(i)} y_a^t = 0, \qquad i\in N_E, t\in T \tag{1.13q}$$

$$\sum_{a\in\delta^+(i)} y_a^t \le 1, \qquad i\in N\backslash N_E, t\in T \tag{1.13r}$$

$$y_{ji}^t \le \sum_{a\in\delta^+(i)} y_a^t, \qquad i\in N\backslash N_E, j\in\delta^-(i), t\in T \tag{1.13s}$$

$$y_{ij}^t \le \sum_{a\in\delta^-(i)} y_a^t, \qquad i\in N\backslash N_A, j\in\delta^+(i), t\in T \tag{1.13t}$$

$$y_a^t \in \{0,1\}, \qquad a\in A, t\in T \tag{1.13u}$$

$$\hat{y}_a^k, \check{y}_a^k \in \{0,1\}, \qquad a\in A, k\in K \tag{1.13v}$$

Formulation 1.13: MSTP: Cinkler's formulation.

$$\sum_{a\in\delta^+(u)} \hat{x}_a^{ut} = 1, \qquad u \in N\backslash\{r\}, t \in T \tag{1.14a}$$

$$\sum_{a\in\delta^+(i)} \hat{x}_a^{ut} - \sum_{a\in\delta^-(i)} \hat{x}_a^{ut} = 0, \qquad u, i \in N\backslash\{r\} : u \neq i, t \in T \tag{1.14b}$$

$$z_a^t \geq \hat{x}_a^{ut}, \qquad u \in N\backslash\{r\}, a \in A, t \in T \tag{1.14c}$$

$$\sum_{a\in\delta^+(r))\in A} z_a^t = 0, \qquad t \in T \tag{1.14d}$$

$$\sum_{a\in\delta^+(i)} z_a^t = 1, \qquad i \in N\backslash\{r\}, s \in S \tag{1.14e}$$

$$z_{ij}^t + z_{ji}^t = w_{\{i,j\}}^t, \qquad \{i,j\} \in E, t \in T \tag{1.14f}$$

$$\sum_{t\in T} \phi_k^t = 1, \qquad k \in K \tag{1.14g}$$

$$\sum_{a\in\delta^+(o_k)} x_a^{kt} = \phi_k^t, \qquad k \in K, t \in T \tag{1.14h}$$

$$\sum_{a\in\delta^+(i)} x_a^{kt} - \sum_{a\in\delta^-(i)} x_a^{kt} = 0, \qquad k \in K, i \in N\backslash\{o_k, d_k\}, t \in T \tag{1.14i}$$

$$\sum_{a\in\delta^-(d_k)} x_a^{kt} = \phi_k^t, \qquad k \in K, t \in T \tag{1.14j}$$

$$x_{ij}^{kt} + x_{ji}^{kt} \leq w_{\{i,j\}}^t, \qquad k \in K, \{i,j\} \in E, t \in T \tag{1.14k}$$

$$\sum_{t\in T}\sum_{k\in K} \rho_k(x_{ij}^{kt} + x_{ji}^{kt}) = C_{\{i,j\}}\mu_{\{i,j\}}, \qquad \{i,j\} \in E \tag{1.14l}$$

$$z_a^t \in \{0,1\}, \qquad a \in A, t \in T \tag{1.14m}$$

$$w_e^t \in \{0,1\}, \qquad e \in E, T \in T \tag{1.14n}$$

$$x_a^{kt} \in \{0,1\}, \qquad a \in A, k \in K, t \in T \tag{1.14o}$$

$$\hat{x}_a^{ut} \in \{0,1\}, \qquad a \in A, u \in N\backslash\{r\}, t \in T \tag{1.14p}$$

$$\phi_k^t \in \{0,1\}, \qquad k \in K, t \in T \tag{1.14q}$$

$$f_e \in [0,1], \qquad e \in E \tag{1.14r}$$

Formulation 1.14: MSTP: Santos' set of constraints.

1.6 Benders' decomposition

In this section, we introduce the Benders' decomposition algorithm, first proposed in [Ben62]. This review follows the notes on [CP08] and [Mar12]. The Benders' decomposition algorithm allows for a faster solving of some optimization problems, by exploring special substructures in their mathematical programming formulations. Namely, at each iteration of this algorithm, we fix certain variables with the purpose of making the resulting sub-problem easier to solve. The decision of which variables to fix is, thus, key to the efficiency of the algorithm. Unfortunately, it is also very much problem dependent.

Let us extend the definition of a LP formulation F_{LP}, introduced at the end of section 1.3, such that it contains two types of variables, x and y: $\min\{z = cx + dy : Ax + By \geq b,\ \ x, y \geq 0\}$, where B and d are respectively, the constraints' and objective function's coefficients for variables y. If we fix variables y to some trial values $\bar{y}$, we obtain the following slave problem $F_{LP}(\bar{y})$: $\min\{z = cx + d\bar{y} : Ax \geq b - B\bar{y}, x \geq 0\}$. Consider the dual of this sub-problem, $F_D(\bar{y})$: $\max\{\omega = \pi(b - B\bar{y}) + d\bar{y} : \pi A \leq c, \pi \geq 0\}$. The optimal value of $F_D(\bar{y})$ provides a lower bound to the objective function value on $F_{LP}(\bar{y})$. What we would like, however, is to derive a function $\beta_{\bar{y}}(y)$, that provides a lower bound that is valid for any fixing of variables y, and not just the current $\bar{y}$.

If ω is finite, the dual problems gives us a bound of the form $z \geq \pi(b - B\bar{y}) + d\bar{y}$. Moreover, note that the solution π is feasible for any fixing of y, and so, we can generalize this bound to be $z \geq \beta_{\bar{y}}(y) = \pi(b - By) + d\bar{y}$. We call this bound a Benders optimality cut.

However, the dual problem $F_D(\bar{y})$ can also be unbounded, if we choose $\bar{y}$ such that $F_{LP}(\bar{y})$ is infeasible. In that case, we can infer a Benders feasibility cut, of the form $\beta_{\bar{y}}(y) = \vec{\pi}(b - By) + d\bar{y} \leq 0$, where $\vec{\pi}$ is an extreme ray for $F_D(\bar{y})$. Note that in practice, many times the objective function in $F_D(\bar{y})$ is fixed to a normalization positive value, so that the dual sub-problem is not unbounded. This allow us to find the source of infeasibility, without having to find extreme rays, which is very hard from a computational point of view [FSZ10].

Let $\bar{y}^k$ be the trial values for variables y, fixed at the k^{th} iteration of the Benders' decomposition algorithm. We name as Benders' master problem (or simply master problem) to the problem F_M^k, that aggregates all the cuts generated until the k^{th} iteration: $\min\{z : z \geq \beta_{\bar{y}^k}(y) \vee \beta_{\bar{y}^k}(y) \geq 0\ \ \forall k,\ \ y \geq 0\}$.

Algorithm 1.1 describes the general procedure for the Benders' decomposition algorithm. This procedure can be integrated in a branch-and-cut (B&C) framework, by solving the Benders' decomposition at each node of a branch-and-bound (B&B) algorithm. Recently, this approach has been used successfully to solve efficiently hard MIP problems (*e.g.* [BFGP13], [FLS15]). The efficiency of a B&C depends on many different factors. For instance, as mentioned above, the choice of which variables to fix can have a great impact: in some cases, the correct choice allows the slave problem to be decomposable in many, easier-to-solve problems; in other cases, the correct choice can lead to a slave problem which is known, and for which there exists fast and efficient algorithms. Another important factor is the strength of the generated cuts; *e.g.* if they are facet

defining. In [FLS15], Fischetti *et al.* apply Benders' decomposition to the Uncapacitated Facility Location problem. They claim that the success of their approach is in great part due to the implementation of a procedure that stabilizes the cut loop at the root node of the B&C algorithm. The role of this procedure is compared to the one of the bundle method in the usual Lagrangian dual minimization. In their work, the authors do not exactly implement a real bundle method, but an in-out variant.

The in-out procedure is described in Algorithm 1.2. The main idea is that at each cut loop iteration, there are two y points: $\bar{y}$, the optimal solution of the LP relaxation of the current master problem, F_{MLP}; and a stabilizing point $\tilde{y}$. In [FLS15], this stabilizing point is initialized to $(1, ..., 1)$, but alternatively it can be initialized to a integer feasible solution, obtained by a heuristic method. At each iteration, $\tilde{y}$ moves halfway towards $\bar{y}$. An intermediate point between $\tilde{y}$ and $\bar{y}$ is fed to the dual slave problem, in order to obtain a Benders' cut that is added to the master. The procedure finishes either if this intermediate point is the optimal solution to F_{MLP}, or after 15 consecutive iterations with no LP bound improvement.

As integrating the cut loop within CPLEX is not immediate, in practice, the procedure is implemented as a pre-processing to the B&C algorithm. Consequently, the in-out procedure is only used for the root node. For the other nodes of the tree, a limit of 20 consecutive cut loop iterations is set.

Algorithm 1.1: Benders' decomposition algorithm.

```
1  Choose initial ȳ
2  z̄ ← −∞
3  k ← 0
4  Solve F_D(ȳ)
5  while ω* ≥ z̄ do
6      k ← k + 1
7      y^k ← ȳ
8      if ω* = ∞ then
9          Add β_ȳ(y) ≤ 0 to F_M^k
10     if ω* is finite then
11         Add z ≥ β_ȳ(y) to F_M^k
12     Solve F_M^k
13     if F_M^k is infeasible then
14         Stop. F_LP is infeasible
15     else
16         z̄ ← z*
17         ȳ ← y*
18     Solve F_D(ȳ)
```

Algorithm 1.2: Fischetti *et al.* in-out cut loop stabilization algorithm.

$\lambda \leftarrow 0.2$
$\delta \leftarrow 2\varepsilon$
$k \leftarrow 0$
$\check{k} \leftarrow 0$
$\bar{z} \leftarrow -\infty$
Choose stabilizing point $\tilde{y}$.
while $\check{k} \leq 15$ **do**
    $k \leftarrow k + 1$
    if $\check{k} = 5$ **then**
        $\lambda \leftarrow 1$
    if $\check{k} = 10$ **then**
        $\delta \leftarrow 0$
    Solve F^k_{MLP}
    $\bar{y} \leftarrow$ solution of F^k_{MLP}
    $\bar{z}^k \leftarrow$ optimal solution of F^k_{MLP}
    if $\bar{z}^k - \bar{z}^{k-1} \leq \varepsilon$ **then**
        $\check{k} \leftarrow \check{k} + 1$
    else
        $\check{k} \leftarrow 0$
    $\tilde{y} \leftarrow \frac{1}{2}(\bar{y} + \tilde{y})$
    $\check{y} \leftarrow \lambda\bar{y} + (1 - \lambda)\tilde{y} + \delta(1, ..., 1)$
    Solve $F_D(\check{y})$
    if $\omega^* \geq \bar{z}$ **then**
        Add Benders' cut(s) to F^k_{MLP}
    else
        Stop.
    if $k \pmod 5 = 0$ **then**
        Remove cuts with positive slack from F^k_{MLP}

Chapter 2

MSTP: minimization of worst-case link utilization

In this chapter, we study the problem of finding optimal designs for switched Ethernet networks implementing the MSTP, first proposed by Ho *et al.* [HDBF11, Ho12]. We denote it as the TE for the MSTP (TE-MSTP) problem. Let $G = (N, E)$ be an undirected graph, as defined in the Section 1.3. Moreover, let C_e be the capacity on the load of edge $e \in E$. This capacity is regarded as symmetric, in the sense that it limits the traffic flowing in both directions, (i, j) and (j, i), together.

As for the OCST problem, we define the set of commodities K, each $k \in K$ with a given origin o_k, destination d_k, and traffic demand ρ_k. We also define T as the set of VLANs in the SEN. For each VLAN $t \in T$, let $\rho^t_{uv} = \sum_{k \in K^t : o_k = u, d_k = v} \rho_k$, where $K^t \subseteq K$ is the set of commodities assigned to VLAN $t \in T$. We assume that $u < v, u, v \in N$, and that ρ_{uv} stands for the sum of traffic to be sent, both from u to v and from v to u.

The TE-MSTP problem consists of finding a design for all VLANs $t \in T$, such that we minimize the worst-case link utilization - the ratio between the link's load and its capacity. Furthermore, a feasible set of designs must satisfy the following properties:

- the topology of each VLAN is a spanning tree;
- all given traffic demands in a VLAN are routed;
- the total traffic flowing through a link does not exceed its given capacity.

The TE-MSTP problem is shown to be $\mathcal{NP}$-hard in Section 2.1. Then, we propose three different MIP formulations to model the TE-MSTP problem, in Section 2.2. In Section 2.3, we compare the LP relaxations of these formulations. Computational experiments are presented in Section 2.4, to further compare the proposed formulations. In Section 2.5, we propose a B&C algorithm for the problem. Finally, in Section 2.6, we draw some conclusions about the research described in this chapter.

2.1 Problem complexity

In this section, we show that the TE-MSTP problem is $\mathcal{NP}$-hard. A similar problem, the OCST problem was proved to be $\mathcal{NP}$-hard [JLK78]. Note however that the OCST problem does not consider flow capacities, and therefore cannot be used to show that the TE-MSTP problem is also $\mathcal{NP}$-hard.

We begin our proof by considering the corresponding decision problem, which is defined as follows: Let $G = (N, E)$ be an undirected graph, with capacity C_e assigned to each edge $e \in E$; and T a set of VLANs, where in each $t \in T$, we have traffic demand ρ_{uv}^t, between nodes $u \in N$ and $v \in N$. The TE-MSTP decision problem asks the question, "Is it possible to design each VLAN as a spanning tree, such that all the traffic demands are routed without exceeding the capacities on the link?".

We use the boolean satisfiability (SAT) problem to demonstrate that this decision problem is $\mathcal{NP}$-complete. Then we conclude the proof by showing that if the decision problem is $\mathcal{NP}$-complete, the TE-MSTP problem is $\mathcal{NP}$-hard .

Theorem 1. *The TE-MSTP decision problem is $\mathcal{NP}$-complete, for $|T| \geq 1$.*

Proof. Let (X, C) be an instance of the SAT problem. $X = \{x_1, x_2, ..., x_\nu\}$ is the set of boolean variables, where each $x_i \in X$ can either be a positive literal (denoted by χ_i) or a negative literal (denoted by $\overline{\chi}_i$). $C = \{c_1, c_2, ..., c_\mu\}$ stands for the set of clauses, where each clause c is a disjunction of literals. In this version of the problem, each clause can have any given number of literals, from 1 to ν. A truth assignment $\{x_1^*, x_2^*, ..., x_\nu^*\}$ is defined as an assignment of the boolean value *true* or *false* to each variable $x \in X$. A given clause is said to be satisfied under that truth assignment, if it contains the literal χ and $x^* := true$, or the literal $\overline{\chi}$ and $x^* = false$. The objective of the SAT problem is to find out if there is a truth assignment such that all clauses in C are satisfied. Cook and Levin proved that the SAT problem is $\mathcal{NP}$-complete [Coo71, Lev73].

We show that the TE-MSTP decision problem is polynomially reducible to the SAT problem, and therefore, it is $\mathcal{NP}$-complete too. Consider an instance of the SAT problem (X, C), as defined above. We now describe how to construct an undirected graph $G = (N, E)$, to model this instance.

For each variable $x \in X$ we consider a triple of nodes denoted as x^0, x^T and x^F. The first node acts as a "representative" of the variable, whereas the two latter imply the choice of assignment - respectively, *true* or *false*. Moreover, we consider a node for each clause $c \in C$ and a root node, denoted by r.

Let $C_{\chi_i} = \{c \in C : \chi_i \text{ appears in } c\}$ and $C_{\overline{\chi}_i} = \{c \in C : \overline{\chi}_i \text{ appears in } c\}$. The set E is comprised of the following group of edges, for each variable $x \in X$:

- $\{r, x^0\}$, with capacity μ;
- $\{x^0, x^T\}$, with capacity $|C_\chi|$;
- $\{x^0, x^F\}$, with capacity $|C_{\overline{\chi}}|$;

- $\{x^T, x^F\}$, with capacity $\mu + 1$;
- $\{x^T, c\}, c \in C_\chi$, with capacity 1;
- $\{x^F, c\}, c \in C_{\overline{\chi}}$, with capacity 1.

Finally, we assume that there exists the following set of demands: $\rho_{rc} = 1$, for every clause $c \in C$; and $\rho_{x^T x^F} = \mu + 1$, for every variable $x \in X$.

It is easy to see that G can be constructed in polynomial time. Figure 2.1 depicts what this graph looks like for a SAT instance with four variables, and the following five clauses: $(\chi_1 \vee \overline{\chi}_4)$, $(\overline{\chi}_1 \vee \overline{\chi}_2 \vee \chi_3)$, $(\chi_3 \vee \chi_4)$, $(\chi_2 \vee \chi_4)$ and $(\overline{\chi_3} \vee \overline{\chi_4})$. The thicker edges represent the following feasible solution: $x_1 = false$, $x_2 = true$, $x_3 = true$, $x_4 = false$.

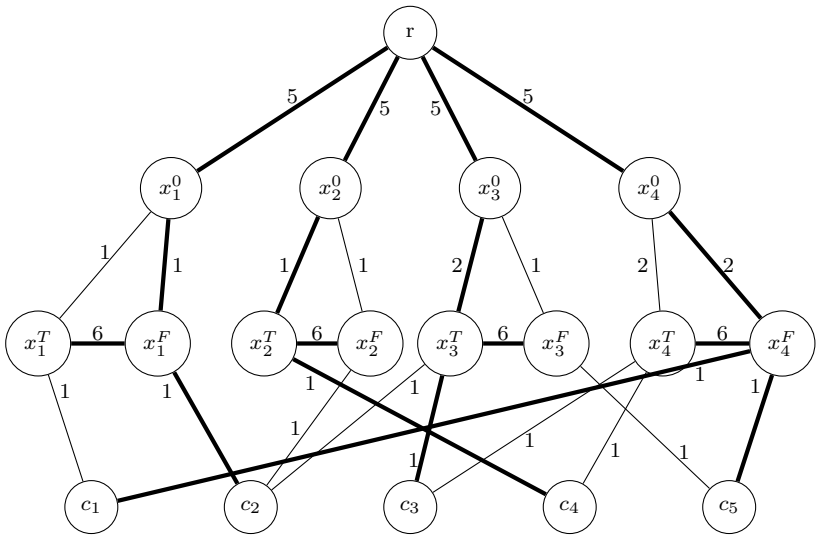


Figure 2.1: Graph construction for an example of a SAT instance.

We now show that there exists a feasible truth assignment for the SAT problem, if and only if the answer for the TE-MSTP decision problem is positive for G and the aforementioned set of demands.

Consider the feasible truth assignment $\{x_1^*, x_2^*, ..., x_\nu^*\}$. This solution can be represented by a sub-graph $S^* = (N, E^*) : E^* \subseteq E$, that routes the demands described above, whilst not surpassing the capacities on the edges. The assignment of a given variable $x \in X$ to *true* or *false*, is expressed by selecting respectively the edge $\{x^0, x^T\}$ or $\{x^0, x^F\}$. Each clause $c \in C$ has to be satisfied by, at least, one literal. This is ensured by the existence of a path in E^*, between the corresponding node and the root, such that the demand ρ_{rc} can be routed. The last edge on the path connects the clause node, with the node that indicates which variable assignment ensures the clause's satisfiability. Even if the clause is satisfied by more than one literal, we only include one of these edges in E^*; the others are redundant. Finally, $\{x^T, x^F\} \in E^*$, as the direct edge is the only path through which we can send the corresponding demand, without

exceeding the capacity. If the case arises that the sub-graph constructed in this fashion is not connected (*e.g.* if there are more variables than clauses), we include in S^* the edges needed to ensure connectivity; they will be necessarily of the form $\{r, x^0\}$. Note that no demands will flow through these edges, so the capacity is always satisfied.

It is easy to observe that the topology S^* is a spanning tree, and that the demands flowing through its edges do not surpass the capacities. Therefore S^* is such that the answer to the TE-MSTP decision problem is "yes".

Consider now, a spanning tree S^*, such that it satisfies the TE-MSTP decision problem.

For each variable $x \in X$, only one edge of $\{x^0, x^T\}$ or $\{x^0, x^F\}$ belongs in S^*. This owes to $\{x^T, x^F\}$ naturally being in S^*, as it is the only one through which we can send the demand between those two nodes, such that that capacity is met; and S^* being a spanning tree - if all three edges were to be in S^*, we would have a cycle.

Therefore, we can construct a truth assignment, by assigning to each variable $x \in X$ the value *true* if $\{x^0, x^T\} \in S^*$, the value *false* if $\{x^0, x^F\} \in S^*$, and an arbitrary value if $\{x^0, x^T\}, \{x^0, x^F\} \notin S^*$.

Moreover, there is an unique path between the root node r and the clause node $c \in C$. This path is necessarily of the form $\{r, x^0, x^T, c\}$ or $\{r, x^0, x^F, c\}$. Let us consider the absurd case where there exists a path p between r and c for which this is not true. The first possibility is that there exists another clause node $c' \in C$ on the path p, *e.g.* $p = \{r, x^0, x^{T'}, c', x^T, c\}$. Note however, that the demands both from r to c, and from r to c' ($\rho_{rc} + \rho_{rc'} = 2$), flow through edge $\{x^{T'}, c'\}$, exceeding the capacity ($C_{\{x^{T'}, c'\}} = 1$). The same justification holds for the cases where there is more than one clause on path p. The second possibility is that there exists an edge of the type $\{x^T, x^F\}$ on path p, *e.g.* $\{r, x^0, x^T, x^F, c\}$. Nonetheless, the demands both from x^T to x^F and r to c ($\rho_{x^T x^F} + \rho_{rc'} = \mu + 2$), flow through edge $\{x^T, x^F\}$ exceeding the capacity ($C_{\{x^T, x^F\}} = \mu + 1$).

Thus, as all the clauses are satisfied, S^* also describes a solution for the SAT problem. Therefore, the TE-MSTP decision problem is polynomial reducible to the SAT problem. □

Corollary 1. *TE-MSTP problem is $\mathcal{NP}$-hard, for $|T| \geq 1$.*

Proof. Consider the TE-MSTP(Λ) decision problem, defined as the TE-MSTP decision problem where all the capacities are multiplied by Λ. Solving the TE-MSTP problem is equivalent to solving a succession of TE-MSTP(Λ) decision problems, with Λ given by a binary search algorithm. The algorithm iteratively updates either the upper bound for the optimal solution, if the decision problem is feasible for Λ, or the lower bound, otherwise; until the difference between the two is not larger than the desired precision. Hence, as the TE-MSTP decision problem is $\mathcal{NP}$-complete, the TE-MSTP problem is $\mathcal{NP}$-hard. □

2.2 Problem formulation

In Section 1.4, we presented a review of mathematical programming formulations for the MST and OCST problems. Those formulations provide an important basis, that can be extended and/or adapted in the interest of modelling other spanning tree design problems, namely the one studied in this chapter. In this section, we propose three MIP formulations for the TE-MSTP problem. In order to facilitate the exposition of these formulations, we divide the TE-MSTP problem into three sub-problems, which we begin by looking at individually. We consider the following sub-problems:

Sub-problem 1: Designing spanning trees;

Sub-problem 2: Routing the traffic demands;

Sub-problem 3: Ensuring edges' capacities and calculating edge utilization.

In the next three sections, we propose MIP formulations for each of these sub-problems. In Section 2.2.4 we present the objective function of this problem. Finally, in Section 2.2.5 we describe complete formulations for the TE-MSTPP, that are obtained by combining adequately the formulations proposed in Sections 2.2.1, 2.2.2 and 2.2.3.

2.2.1 Sub-problem 1: Designing spanning trees

There are different ways to model spanning trees as a MIPs. In this section, we present two MIP formulations, that model the design of multiple spanning trees. These build on Formulations 1.7 and 1.8, proposed for the MST problem and described in Section 1.4. Note that in this thesis, we do not consider single commodity flow models for the TE-MSTP problem. As it was seen in Section 1.4, these tend to be weak formulations. In addition, in our preliminary tests we verified that these models were not efficient in solving instances of the TE-MSTP problem.

The set of variables that are used in the proposed formulations are defined as follows:

- $x_a^{uvt} = 1$ if arc $a \in A$ is used on the unique path from node u to node v, in VLAN $t \in T$; 0 otherwise;
- $z_a^{ut} = 1$ if arc $a = (i, j) \in A$ is used on the unique path from root node u to node j, in VLAN $t \in T$; 0 otherwise;
- $w_e^t = 1$ if edge $e \in E$ is used in VLAN $t \in T$; 0 otherwise.

Note that the first two set of variables are defined in the directed graph G', whereas variables w are defined on the original graph. The latter are common to both formulations; in constraints (2.1), we define them as binary.

The first model is a multicommodity flow formulation, that extends Formulation 1.7 in order to design multiple spanning trees. Moreover, we take into attention that we

$$w_e^t \in \{0,1\}, \qquad e \in E, t \in T \tag{2.1}$$

Formulation 2.1: TE-MSTP SP1: defining design variables w.

also extend that formulation, so as to have multiple sources of commodities, in addition to multiple destinations. The reason for this will become apparent further on.

Given the set of variables $\{x_a^{uvs}, w_e^s\}$, any feasible solution for the problem must verify the set of constraints in 2.2, in addition to (2.1).

$$\sum_{a \in \delta^+(u)} x_a^{uvt} = 1, \qquad u, v \in N : u < v, t \in T \tag{2.2a}$$

$$\sum_{a \in \delta^-(j)} x_a^{uvt} - \sum_{a \in \delta^+(j)} x_a^{uvt} = 0, \qquad u, v, i \in N : u < v, i \neq \{u, v\}, t \in T \tag{2.2b}$$

$$x_{ij}^{uvt} + x_{ji}^{uvt} \leq w_e^t, \qquad u, v \in N : u < v, e = \{i, j\} \in E, t \in T \tag{2.2c}$$

$$\sum_{e \in E} w_e^t = n - 1, \qquad t \in T \tag{2.2d}$$

$$x_{ij}^{uvt} \in \{0,1\}, \qquad (i,j) \in A, u, v \in N : u < v, j \neq u, i \neq v, t \in T \tag{2.2e}$$

Formulation 2.2: TE-MSTP SP1: multicommodity flow formulation.

The flow conservation constraints (2.2a) and (2.2b) define for each VLAN, a path between every pair of nodes, through which the traffic is routed. Constraints (2.2c) guarantee that, in each VLAN, the paths only make use of the edges chosen in the according VLAN. The set (2.2c) are edge-cardinality constraints, for each VLAN. Finally, constraint set (2.2e) define the integrality of the x variables. Observe that we can define instead as continuous and non-negative.

As it was seen for 1.7, constraints (2.2c) can be extended as seen in 2.3. These type of valid inequalities were proposed in [BMW89], so as to improve previous formulations for the fixed-charge network design problems. The first set of constraints, (2.3a), states that when edge $e = \{i, j\} \in E$ is used in a given VLAN, then all traffic originated in a given node $u \in N$, will flow either from i to j, or from j to i. (2.3b) describes an equivalent situation, for all traffic flowing to a given node $v \in N$.

$$x_{ij}^{uvt} + x_{ji}^{uv't} \leq w_e^t, \qquad u, v, v' \in N : u < v, u < v', e = \{i, j\} \in E, t \in T \tag{2.3a}$$

$$x_{ij}^{uvt} + x_{ji}^{u'vt} \leq w_e^t, \qquad u, u', v \in N : u < v, u' < v, e = \{i, j\} \in E, t \in T \tag{2.3b}$$

Formulation 2.3: TE-MSTP SP1: tightening for (2.2c).

Observe that in the specific cases described in 2.4, the previous valid inequalities can instead be re-written as equalities.

$$x_{ij}^{ujt} + x_{ji}^{uit} = w_e^t, \qquad u \in N, e = \{i,j\} \in E : u < j, u < i, t \in T \tag{2.4a}$$
$$x_{ij}^{ivt} + x_{ji}^{jvt} = w_e^t, \qquad v \in N, e = \{i,j\} \in E : i < v, j < v, t \in T \tag{2.4b}$$

Formulation 2.4: TE-MSTP SP1: further tightening for special cases of (2.2c).

Constraints (2.4a) assert that if edge $e = \{i,j\}$ is used in VLAN $t \in T$, this necessarily means that either the traffic flowing between a given node $u \in N$ and j travels from i to j, or the traffic flowing between u and i travels in the opposite direction. Constraints (2.4b) describes an equivalent situation, where the common node between the two traffic demands is not the origin, but the destination. Empirical evidence has revealed that for some instances, by using the valid inequalities in 2.3 along with the equalities in 2.4, we are able to strengthen the bound of the LP relaxation of the previous model.

The second model can be obtained by adapting Formulation 1.8 for the multiple spanning tree case. Given the set of variables $\{z_a^{us}, w_e^s\}$, any feasible solution must verify the set of constraints in 2.5, in addition to (2.1).

$$\sum_{a \in \delta^-(j)} z_a^{ut} = 1, \qquad u, j \in N : u \neq j, t \in T \tag{2.5a}$$
$$z_{ij}^{ut} + z_{ji}^{ut} = w_e^t, \qquad u \in N, e = \{i,j\} \in E, t \in T \tag{2.5b}$$
$$z_{ij}^{ut} \in \{0,1\}, \qquad u \in N, (i,j) \in A : j \neq u, t \in T \tag{2.5c}$$

Formulation 2.5: TE-MSTP SP1: rooted directed formulation.

In each VLAN, we design an arborescence rooted at each node $u \in V$, by defining an unique path between the root node and every other node. As such, constraints (2.5a) state that in each arborescence, every node, with the exception of the root, has an indegree of 1. In constraint set (2.5b), we ensure that all arborescences, in each VLAN, use the same edges. In (2.5c), we define variables z as binary. Once again, these variables can be defined instead as continuous and non-negative.

2.2.2 Sub-problem 2: Routing the traffic demands

Once the design for each VLAN is established, it is possible to route the traffic flows according to the demands. To properly model this routing, we can look at the traffic demands in each VLAN in two distinct ways: either by considering the traffic demands between each pair of nodes separately, or by aggregating traffic that shares a single origin (or destination).

As seen in Section 2.2.1, the x variables describe a path between each pair of nodes.

Therefore, it is natural to associate the first above-mentioned strategy with the use of these variables, as they allow for the immediate calculation of the load in each edge, of VLAN $t \in T$: $\sum_{u,v \in N: u<v} \rho_{uv}^t (x_{ij}^{uvt} + x_{ji}^{uvt})$. In this sense, to model this second sub-problem, we need to define for every VLAN, the unique path between each pair of nodes, in constraints 2.6

$$\sum_{a \in \delta^+(u)} x_a^{uvt} = 1, \qquad u, v \in N : u < v, t \in T \quad (2.2a)$$

$$\sum_{a \in \delta^-(j)} x_a^{uvt} - \sum_{a \in \delta^+(j)} x_a^{uvt} = 0, \qquad u, v, i \in N : u < v, i \neq \{u, v\}, t \in T \quad (2.2b)$$

$$x_{ij}^{uvt} \in \{0, 1\}, \qquad (i, j) \in A, u, v \in N : u < v, j \neq u, i \neq v, t \in T \quad (2.2e)$$

Formulation 2.6: TE-MSTP SP2: multi-source-multi-destination routing.

Observe that the constraints in 2.6 are the same as in 2.2. We repeat them here for the sake of completeness, since they are also used to model the routing of the traffic demands. Nevertheless, when the two sub-problems are combined, the redundancy is naturally omitted.

For the second strategy, in order to aggregate traffic that has a common origin, the following load variable set is defined:

- l_a^{ut} = traffic load originated from node $u \in N$, in arc $a \in A$, in VLAN $t \in T$.

These variables are related to the x variables, since $l_a^{ut} = \sum_{v \in N \setminus \{u,i\}} \rho_{uv}^t \cdot x_a^{uvt}$. They must verify the sets of constraints in 2.7, so that the traffic flows are correctly distributed. Constraint set (2.7a) defines the quantity of aggregated traffic flow leaving from each origin, to be sent to all other nodes. Constraints (2.7b) state that the difference between the traffic quantity originated from node $u \in N$, in VLAN $t \in T$, entering and exiting a given node $i \in N$ (different from u) must match, exactly, the traffic demand between nodes u and i, on that same VLAN.

$$\sum_{a \in \delta^+(u)} l_a^{ut} = \sum_{v \in N: v \neq u} \rho_{uv}^t, \qquad u \in N, t \in T \quad (2.7a)$$

$$\sum_{a \in \delta^-(i)} l_a^{ut} - \sum_{a \in \delta^+(i)} l_a^{ut} = \rho_{ui}^t, \qquad u, i \in N : u \neq i, t \in T \quad (2.7b)$$

$$l_{ij}^{ut} \geq 0, \qquad u \in N, (i, j) \in A : j \neq u, t \in T \quad (2.7c)$$

Formulation 2.7: TE-MSTP SP2: single-source-multi-destination routing.

2.2.3 Sub-problem 3: Edge utilization and capacity constraints

The last sub-problem to consider deals, in fact, with two different components. Firstly, it guarantees that the distribution of the traffic flows, that was made in the second sub-problem, does not exceed the capacity of each edge. Secondly, it calculates the utilization of each link. Nevertheless, as both these components can be dealt within the same constraint set, we see it as one sub-problem.

In order to calculate the maximum edge utilization on the network, the following variable is defined:

- U^{max} = maximum value of edge utilization.

As mentioned in the last section, the edge load (total sum of traffic travelling through a link) can be calculated via either the x variables or l variables. Hence, there are two distinct constraint sets which can be used, depending on which variables are used: (2.8) or (2.9). In both cases, in the left-hand side of each constraint, the edge load is calculated. That way, it is possible to determine, as well, each edge utilization. As this is done for every existing edge on E, the variable U^{max} is attributed the value of the maximum utilization. At the same time, (2.8) and (2.9) bound the traffic quantity flowing through each edge to the given capacity, as $U^{max} \in [0, 1]$ (2.10).

$$\sum_{t \in T} \sum_{u,v \in N : u<v} \rho_{uv}^{t} (x_{ij}^{uvt} + x_{ji}^{uvt}) \leq C_e \cdot U^{max}, \qquad e = \{i, j\} \in E \tag{2.8}$$

Formulation 2.8: TE-MSTP SP3: disaggregated case.

$$\sum_{t \in T} \sum_{u \in N} (l_{ij}^{ut} + l_{ji}^{ut}) \leq C_e \cdot U^{max}, \qquad e = \{i, j\} \in E \tag{2.9}$$

Formulation 2.9: TE-MSTP SP3: aggregated case.

$$0 \leq U^{max} \leq 1 \tag{2.10}$$

Formulation 2.10: TE-MSTP SP3: bound on edge utilization.

2.2.4 Objective function

The objective of the problem is to minimize the worst-case edge utilization. As it was seen in the last section, every formulation uses variable U^{max}. Hence, their common

objective function is (2.11).

$$\min \quad U^{max} \tag{2.11}$$

Formulation 2.11: TE-MSTP problem: objective function.

2.2.5 Complete formulations

Having modelled each one of the three sub-problems, it is now possible to go back and look at the original problem as a whole. We do this by combining adequately the formulations for the different sub-problems, which were presented in Sections 2.2.1 to 2.2.3. The result are three complete models: the multicommodity flow formulation (MFM), the rooted directed formulation (RDM), and the rooted directed multicommodity flow formulation (RDMFM).

MFM formulates the first sub-problem via (2.1,2.2a-2.2b,2.2d-2.2e), along with strengthened constraints (2.3a-2.3b,2.4a-2.4b). To route the traffic demands, MFM also uses (2.2a-2.2b,2.2e). Notwithstanding, as mentioned above, we do not repeat them in the complete formulation. Finally for the third sub-problem, MFM uses (2.8,2.10).

RDM uses constraint sets (2.1,2.5a-2.5c) to define each VLAN as a spanning tree and (2.7a-2.7c) to route the traffic demands. To link the variables involved in the formulations of these two sub-problems, we add constraints (2.12). These constraints imply that the traffic demands originated at node $u \in N$ and VLAN $t \in T$, can only flow through arcs selected for an arborescence rooted at node u of the same VLAN. Finally, RDM uses (2.9,2.10) to calculate edge utilization, whilst ensuring edge capacity.

$$l_{ij}^{ut} \leq \sum_{v \in N: u<v, u \neq i} \rho_{uv}^{t} \cdot z_{ij}^{ut}, \qquad (i,j) \in A, u \in N \backslash \{j\}, t \in T \tag{2.12}$$

Formulation 2.12: TE-MSTP problem: linking constraints for RDM.

Finally, it is also possible to combine the formulations proposed for Sub-problem 1 and 2 in a different way: we can opt to model the spanning trees as arborescences with (2.1,2.5a-2.5c), but use the multicommodity flows defined in (2.2a-2.2b,2.2e) to route the traffic demands instead. The variables used for these two sub-problems are linked via the constraints (2.13).

$$x_{ij}^{uvt} \leq z_{ij}^{ut}, \qquad (i,j) \in A, u, v \in N : u < v, j \neq u, i \neq v, t \in T \tag{2.13}$$

Formulation 2.13: TE-MSTP problem: linking constraints for RDMFM.

$$x_{ij}^{uvt} \leq z_{ji}^{vt}, \qquad (i,j) \in A, u, v \in N : u < v, j \neq u, i \neq v, t \in T \tag{2.14}$$

Formulation 2.14: TE-MSTP problem: linking constraints for RDMFM.

These constraints imply that the traffic demands directed from node $u \in N$ to $v \in N$ can only flow through a given arc, if that arc is selected in the arborescence rooted at node u. Empirical evidence has revealed that the bounds of LP relaxation of this formulation can be improved for some instances, by also adding the linking valid inequalities (2.14). This third complete formulation is named RDMFM.

For each model, listed in the first column, Table 2.1 describes the variable set, and enumerates which sets of constraints are used in the model to solve each sub-problem (*SP1, SP2, SP3*), as well as sets of constraints used to link the first two sub-problems (*Link*). In addition, all models use the objective function described in Section 2.2.4.

Model	Variables	SP1	SP2	Link	SP3
MFM	$\{U^{max}, x_a^{uvt}, w_e^t\}$	(2.1) (2.2a-2.2b,2.2d-2.2e) (2.3a-2.3b) (2.4a-2.4b)	(2.2a-2.2b) (2.2e)	-	(2.8,2.10)
RDM	$\{U^{max}, l_a^{ut}, z_a^{ut}, w_e^t\}$	(2.1) (2.5a-2.5c)	(2.7a-2.7c)	(2.12)	(2.9,2.10)
RDMFM	$\{U^{max}, x_a^{uvt}, z_a^{ut}, w_e^t\}$	(2.1) (2.5a-2.5c)	(2.2a-2.2b) (2.2e)	(2.13) (2.14)	(2.8,2.10)

Table 2.1: TE-MSTP: composition of each complete formulation.

2.3 Polyhedral comparison of formulations

Consider the LP relaxations of the models introduced in the last section. In this section, we compare the strength of the LP relaxed models. An introduction of some of the concepts used in this section, can be found in Section 1.3.

Let $\mathcal{P}_{RD}$, $\mathcal{P}_{MF}$ and $\mathcal{P}_{RDMF}$ be the polyhedron defined by the set of feasible solutions of the LP relaxation of RDM, MFM and RDMFM, respectively.

Theorem 2. $Proj_{U^{max},w}(\mathcal{P}_{MF}) \subseteq Proj_{U^{max},w}(\mathcal{P}_{RD})$.

Proof. Theorem 2 can be proven by showing that any solution of the projection of $\mathcal{P}_{MF}$ onto the space of U^{max}, z, and w, can be transformed to a solution in $\mathcal{P}_{RD}$. That is to say, any feasible solution of the LP relaxation of MFM can be converted to a solution that verifies all the constraints of the LP relaxation of RDM. Consider a generic solution of $\mathcal{P}_{MF}$, $\tilde{\mathcal{S}} = \{\tilde{w}_e^t, \tilde{x}_a^{uvt}, \tilde{U}^{max}\}$. Consider, as well, $\hat{\mathcal{S}} = \{\tilde{w}_e^t, \hat{z}_a^{ut}, \hat{l}_a^{ut}, \tilde{U}^{max}\}$, such that:

- $\hat{z}_a^{ut} := \tilde{x}_a^{ujt}, \quad u \in N, a = (i,j) \in A : j \neq u, t \in T$

- $\hat{l}_a^{ut} := \sum_{v \in V \setminus \{u,i\}} \rho_{uv}^t \cdot \tilde{x}_a^{uvt}, \quad u \in N, a = (i,j) \in A : j \neq u, T \in T$

The w and U^{max} variables, having the same meaning in both models, can be directly converted. The values of the z variables can be deduced from the $\tilde{x}$ variables in $\mathcal{P}_{MF}$. Finally, given the routing of traffic demands between each pair of nodes in a VLAN, it is easy to compute the load in each link. Thus, in a given VLAN, the traffic quantity originated in node $u \in N$ and flowing through arc $a = (i,j) \in A$, is equal to the sum of the fraction of traffic demands with origin in node u and destination in every other node $v \in N$, routed through that arc. From this sum we exclude $v = u$ and $v = i$, as the x variables are not defined for these indexes. In the following paragraphs, we demonstrate that $\hat{\mathcal{S}}$ satisfies all the constraints in the LP relaxation of RDM.

Naturally, constraints (2.1,2.10) are satisfied by $\hat{\mathcal{S}}$, as they are common to both models.

By (2.2a-2.2b), we show that (2.5a) is satisfied, for every $u, j \in N : u \neq j, t \in T$:

$$\sum_{a \in \delta^-(j)} \hat{z}_a^{ut} = \sum_{a \in \delta^-(j)} \tilde{x}_a^{ujt} \overset{(2.2a+2.2b)}{=} 1$$

Through (2.4a), it can be seen that $\hat{\mathcal{S}}$ satisfies constraint set (2.5b), for all $u \in N, e = \{i,j\} \in E, t \in T$:

$$\hat{z}_{ij}^{ut} + \hat{z}_{ji}^{ut} = \tilde{x}_{ij}^{ujt} + \tilde{x}_{ji}^{uit} \overset{(2.4a)}{=} \tilde{w}_e^t$$

Moreover, $\hat{\mathcal{S}}$ naturally satisfies the LP relaxation of (2.5c).

By (2.2a), we show that (2.7a) are satisfied, for every $u \in N : u, t \in T$:

$$\begin{aligned}
\sum_{a \in \delta^+(u)} \hat{l}_a^{ut} &= \sum_{a \in \delta^+(u)} \sum_{v \in N \setminus \{u,i\}} \rho_{uv}^t \cdot \tilde{x}_a^{uvt} \\
&= \sum_{v \in N : v \neq u} \rho_{uv}^t \sum_{a \in \delta^+(u)} \tilde{x}_a^{uvt} \\
&\overset{(2.2a)}{=} \sum_{v \in N : v \neq u} \rho_{uv}^t \cdot 1 = \sum_{v \in N : v \neq u} \rho_{uv}^t
\end{aligned}$$

We prove that $\hat{\mathcal{S}}$ verifies constraint set (2.7b), for all $u, v, i \in N : u < v, i \neq \{u,v\}, t \in T$, as follows:

$$
\begin{aligned}
&\sum_{a\in\delta^-(i)} \hat{l}_a^{ut} - \sum_{a\in\delta^+(i)} \hat{l}_a^{ut} \\
&= \sum_{a\in\delta^-(i)} \sum_{v\in N\setminus\{u,j\}} \rho_{uv}^t \cdot \tilde{x}_a^{uvt} - \sum_{a\in\delta^+(i)} \sum_{v\in N\setminus\{u,i\}} \rho_{uv}^t \cdot \tilde{x}_a^{uvt} \\
&= \sum_{v\in N: v\neq u} \rho_{uv}^t \cdot \Big(\sum_{a\in\delta^-(i)} \tilde{x}_a^{uvt} - \sum_{a\in\delta^+(i)} \tilde{x}_a^{uvt} \Big) \\
&= \rho_{uv}^t \cdot \Big(\sum_{a\in\delta^-(i)} \tilde{x}_a^{uit} - \sum_{a\in\delta^+(i)} \tilde{x}_a^{uit} \Big) \\
&+ \sum_{v\in N\setminus\{u,i\}} \rho_{uv}^t \cdot \Big(\sum_{a\in\delta^-(i)} \tilde{x}_a^{uvt} - \sum_{a\in\delta^+(i)} \tilde{x}_a^{uvt} \Big) \\
&\overset{(2.3a+2.3b)}{=} \rho_{ui}^t \cdot 1 + \sum_{v\in N\setminus\{u,i\}} \rho_{uv}^t \cdot 0 = \rho_{ui}^t
\end{aligned}
$$

As the demands are all non-negative, it is obvious that $\hat{\mathcal{S}}$ satisfies (2.7c).

To prove that $\hat{\mathcal{S}}$ verifies constraints (2.12), for all $a = (i,j) \in A, u \in N\setminus\{j\}, t \in T$, it is first necessary to show that $\tilde{x}_a^{uvt} \leq \tilde{x}_a^{ujt}$ for all cases. This can be achieved by replacing $\tilde{w}_{\{i,j\}}^t$, in the particular case of (2.3a) where $v' = i$, by the value given in (2.4a). The proof is completed as follows:

$$
\hat{l}_a^{ut} = \sum_{v\in N\setminus\{u,i\}} \rho_{uv}^t \cdot \tilde{x}_a^{uvt} \overset{(2.3a+2.4a)}{\leq} \sum_{v\in N\setminus\{u,i\}} \rho_{uv}^t \cdot \tilde{x}_a^{ujt} = \sum_{v\in N\setminus\{u,i\}} \rho_{uv}^t \cdot \hat{z}_a^{ut}
$$

Finally, as $\tilde{\mathcal{S}}$ satisfies (2.8), $\hat{\mathcal{S}}$ satisfies (2.9), for all $e = \{i,j\} \in E$.

$$
\sum_{t\in T} \sum_{u\in N} (\hat{l}_{ij}^{ut} + \hat{l}_{ji}^{ut}) = \sum_{t\in T} \sum_{u\in N} \sum_{v\in N\setminus\{u,i\}} \rho_{uv}^t \cdot (\tilde{x}_{ij}^{uvt} + \tilde{x}_{ji}^{uvt}) \overset{(2.9)}{\leq} C_e \cdot \tilde{U}^{max} \qquad \square
$$

Theorem 3. *$Proj_{U^{max},x,w}(\mathcal{P}_{RDMF}) \subseteq \mathcal{P}_{MF}$.*

Proof. Following the idea used to prove Theorem 2, Theorem 3 can be proved by showing that any feasible solution of the LP relaxation of RDMFM has a corresponding solution in the LP relaxation of MFM. Consider a generic solution on $\mathcal{P}_{RDMF}$, $\check{\mathcal{S}} = \{\check{w}_e^t, \check{x}_a^{uvt}, \check{U}^{max}, \check{z}_a^{ut}\}$. We now show that the solution $\tilde{\mathcal{S}} = \{\check{w}_e^t, \check{x}_a^{uvt}, \check{U}^{max}\}$, belongs to $\mathcal{P}_{MF}$.

As RDMFM uses the same variables as MFM, in addition to variable z, the transformation of $\check{\mathcal{S}}$ to $\tilde{\mathcal{S}}$ can be made directly. Moreover, since many constraint sets of RDMFM are common to MFM, namely (2.1, 2.2a-2.2b, 2.2e, 2.8-2.10), we only need to show that $\tilde{\mathcal{S}}$ satisfies constraints (2.2d,2.3a-2.3b,2.4a-2.4b).

To prove that $\tilde{\mathcal{S}}$ satisfies constraint set (2.2d), for all $t \in T$, we use (2.5a) and (2.5b):

$$\sum_{e\in E} \check{w}_e^t \overset{(2.5b)}{=} \sum_{e=\{i,j\}\in E} (\check{z}_{ij}^{ut} + \check{z}_{ji}^{ut})$$
$$= \frac{1}{2} \cdot (\sum_{(i,j)\in A} \check{z}_{ij}^{ut} + \sum_{(j,i)\in A} \check{z}_{ji}^{ut}) = \frac{1}{2} \cdot (\sum_{j\in N} \sum_{a\in\delta^-(j)} \check{z}_a^{ut} + \sum_{i\in N} \sum_{a\in\delta^-(i)} \check{z}_a^{ut})$$
$$\overset{(2.5a)}{=} \frac{1}{2} \cdot (\sum_{j\in N:j\neq u} 1 + \sum_{i\in N:i\neq u} 1) = n-1$$

Below, we show that $\tilde{\mathcal{S}}$ satisfies constraints (2.3a) for all $u, v, v' \in N : u < v, u < v', e = \{i, j\} \in E, t \in T$, by (2.5b,2.13,2.14). The proof that $\tilde{\mathcal{S}}$ verifies (2.3b) can achieved in the same fashion.

$$\check{x}_{ij}^{uvt} + \check{x}_{ji}^{uv't} \overset{(2.13+2.14)}{\leq} \check{z}_{ij}^{ut} + \check{z}_{ji}^{ut} \overset{(2.5b)}{=} \check{w}_e^t$$

Finally, in order to prove that $\tilde{\mathcal{S}}$ verifies (2.4a), we begin by demonstrating that $\check{x}_a^{ujt} = \check{z}_a^{us}$ for all $a = (i, j) \in A, u \in N : u < j, t \in T$. We already known that $\check{x}_a^{ujt} \leq \check{z}_a^{us}$, by (2.13). Hence, it is necessary to show that $\check{x}_a^{ujt} \geq \check{z}_a^{ut}$:

$$\check{x}_a^{ujt} \overset{(2.2a+2.2b)}{=} 1 - \sum_{(i',j)\in A:i'\neq i} \check{x}_{i'j}^{ujt} \overset{(2.13)}{\geq} 1 - \sum_{(i',j)\in A:i'\neq i} \check{z}_{i'j}^{ut} \overset{(2.5a)}{=} \check{z}_a^{ut}$$

Thus, the proof that $\tilde{\mathcal{S}}$ verifies (2.4a), for all $u \in N, e = \{i, j\} \in E : u < j, u < i, t \in T$, is the following:

$$\check{x}_{ij}^{ujt} + \check{x}_{ji}^{uit} = \check{z}_{ij}^{ut} + \check{z}_{ji}^{ut} \overset{(2.5b)}{=} \check{w}_e^t$$

The proof that $\tilde{\mathcal{S}}$ verifies (2.4a) can be achieved in a similar way.

□

As a consequence of Theorems 2 and 3, the LP relaxation of RDMFF+ is also as strong as the LP relaxation of the RDF.

Corollary 2. $Proj_{U^{max},w}(\mathcal{P}_{RDMF}) \subseteq Proj_{U^{max},w}(\mathcal{P}_{RD})$.

A natural consequence of Theorems 2 and 3 is that $B_{LP}(RDMFM) \geq B_{LP}(MFM) \geq B_{LP}(RDM)$. Our computation experiments (see the next section) will reveal that for some instances, the LP bounds obtained can actually be different, in the sense that $B_{LP}(RDMFM) > B_{LP}(MFM)$, $B_{LP}(RDMFM) > B_{LP}(RDM)$ or $B_{LP}(MFM) > B_{LP}(RDM)$.

2.4 Computational experiments

In this section, we discuss the results of computational experiments, that can help to further evaluate the quality of the proposed formulations to solve the TE-MSTP problems. The results are presented in Appendix A. All the tests were performed on a Intel Core

i7 CPU 960 @ 3.20GHz (x8) with 12GB of memory with 64 bits, and running Ubuntu 14.04.2 LTS (GNU/Linux $3.2.0-26$−generic x86_64). The tests were done using the MIP solver ILOG CPLEX 12.6, implemented in Java programming language. We allow CPLEX to use all the threads of the machine's processor. In Section 2.4.1 we describe the test sets that were used for these experiments. In Section 2.4.2 and 2.4.3 we analyze the results for the two different test sets.

2.4.1 Test sets for the TE-MSTP problem

These experiments were conducted using two distinct test sets of randomly generated instances, which aim at emulating two different types of network topologies. Instances in both test sets were created such that they span different values for the number of nodes and number of VLANs.

In the first test set, T_{rand}, the set of available edges was randomly distributed in the network; the number of edges is calculated according to a given network density. Each edge was given a traffic capacity value of either 50, 75 or 100 Mbps. For each VLAN, the traffic demands (in Mbps) between nodes were generated following an adaptation of the formula proposed in [FT04]:

$$\rho_{uv}^t = \alpha(O_u^t D_v^t + O_v^t D_u^t) R_{uv}^t e^{\frac{-L_2(u,v)}{2\Delta}} \tag{2.15}$$

For each node $u \in N$ and VLAN $t \in T$, two random numbers, O_u^t and D_v^t are randomly generated in the interval $[0, 1]$. These values reflect, respectively, the attractiveness of each node as a sender and as a receiver. Another value, R_{uv}^t, is generated in the same interval, for each pair of nodes. α is a parameter given as input. In these tests, the Euclidian distance (L_2) was substituted by the length of the shortest path between each pair of nodes, with respect to the number of edges. Δ is the largest distance in the network. The final values were rounded to the nearest integer.

The second test set, T_{3tc}, follows the 3-Tier Cisco architecture, which is a common network topology in private enterprise data centers [Inf07, HDBF11]. This hierarchical architecture consists in a core, an aggregation and an edge tier. The core, at the top of the hierarchy, provides a gateway to the data center, from the extranet, wide area network, or Internet edge. The switches in the second tier, serve as a bridge between the core and the nodes in the edge tier, aggregating the in and outflows. At the lowest level, the edge tier consists of racks of servers, interconnect by a Top of Rack (ToR) switch. We mimic this topology by generating a tripartite graph, in which around 1% of the nodes belong to the set representing the core tier, around 15% belong to the set representing the aggregation tier, and the remaining nodes stand for the ToR switches, in the edge tier. For each ToR we assign between 20 and 80 servers. As each server is only connected to the respective ToR switch, it is not necessary to represent them in the graph. Nevertheless, they are relevant for the generation of the traffic demands. Each ToR has 2 to 8 uplinks, depending on the size of the network. Each core node is connected to every node in the aggregation tier. Every link has a capacity value of

10 Gbps. For each VLAN, the traffic demands between the core nodes and the servers, or between servers, were calculated using the formula described above. All the traffic demands directed at (originating from) servers belonging to a given rack, are considered to have as a destination (origin) the node representing the corresponding ToR switch.

For each class of instances, $T^k_{rand} \in T_{rand}$, $T^k_{3tc} \in T_{3tc}$, five instances were generated and tested. Table 2.2 describes these classes of instances, in terms of number of nodes in the network ($\#nodes$), network density (for T_{rand} only), and number of VLANs in the network ($\#VLANs$). α was given a value of 0.1 for every instance in T_{rand}, and of 10 for every instance in T_{3tc}, so that there was a low chance of having infeasible instances.

Class ID	#nodes	#VLANs	density
T^1_{rand}	8	3	0.4
T^2_{rand}	8	6	0.4
T^3_{rand}	8	3	0.6
T^4_{rand}	8	2	0.8
T^5_{rand}	8	3	0.8
T^6_{rand}	10	2	0.5
T^7_{rand}	10	4	0.5
T^8_{rand}	12	2	0.3
T^9_{rand}	12	4	0.3
T^{10}_{rand}	12	2	0.5
T^1_{3tc}	12	4	
T^2_{3tc}	12	7	
T^3_{3tc}	15	4	
T^4_{3tc}	15	7	
T^5_{3tc}	20	2	
T^6_{3tc}	20	4	

Table 2.2: TE-MSTP: description of each class of instances.

2.4.2 Analysis of the results of test set T_{rand}

In this section, we analyze the results for test set T_{rand}, described in Tables A.1 and A.2. For the sake of clarity these results are also represented in the performance profiles of Figures 2.2 to 2.6. Performance profiles are graphs that depict the percentage of instances (spanned over the y-axis) that are solved under different values of each criteria, detailed in the x-axis.

For instance, Figure 2.2 describes the performance profile for the Gap_{LP} of the three models. The Gap_{LP} is a measure of the quality of the lower bounds obtained by the LP relaxation of each model; this concept was introduced back in Section 1.3. In this performance profile we can observe what had been suggested at the end of Section 2.3: for some instances, $B_{LP}(RDMFM) > B_{LP}(MFM)$ and/or $B_{LP}(MFM) > B_{LP}(RDM)$. Notwithstanding, the difference between the average Gap_{LP} is not hugely significant:

the average Gap_{LP} is 34.3% for RDM, 32.5% for MFM and 31.3% for RDMFM.

Moreover, an important aspect that can be observed is the high fluctuation of these Gap_{LP} values. For example, with RDMFM, the Gap_{LP} vary between 0% and 74.5%. Even among each class for instances, which aggregate instances with the same basic characteristics, the Gap_{LP} can be quite distinct, with an average standard deviation of over 11%. Another curious remark that can be made with respect to the test sets at hand is that, even though this gap tends to increase for bigger and/or denser networks, as expected, it tends to decrease when the number of VLANs increases.

Next, we analyze the performance of the different models, when used by CPLEX to solve the MIPs of T_{rand} instances. For these experiments, we set a time limit of one hour. The results are depicted in the performance profile in Figure 2.3. The graph implies that even though using RDM with CPLEX is faster at solving "easy" instances (in terms of solving time), for the more "demanding" ones, using CPLEX with RDMFM proves to be faster, being able to solve more instances (74%) in the given time limit. Moreover, for the instances that were unsolved by both models, the average gap between the best upper bound and lower bounds found by CPLEX at the end of the time limit (Gap_{end}) is slightly smaller for RDMFM (38%), than for RDM (41%). Using CPLEX with MFM is significantly slower than with the other two formulations. These results reflect the difficulty of solving the TE-MSTP problem, even for small instances.

The relative efficiency of the three models in solving the MIPs, is explained by their performance in other criteria, illustrated in Figure 2.2 and Figures 2.4 to 2.6. Namely, despite having larger LPs that take longer to solve (see Figure 2.4), the fact that the lower bounds of RDMFM are better than the ones of RDM (as seen in Figure 2.2) explains its superior efficiency. Moreover, the lower bounds obtained by the LP relaxation are even slightly improved by CPLEX's automatically-generated cuts: the average gap at the end of the root node's processing (Gap_0) is 30.7% for RDMFM. The smaller gaps of RDMFM also impact the size of the B&B tree, which is significantly smaller for RDMFM than for other models (see Figure 2.5). The weak performance of MFM in solving the MIPs is explained by: very slow-to-solve LPs (see 2.4); large B&B trees (see 2.5); and the superior work of CPLEX's automatically-generated cuts for RDM than for MFM - the average Gap_0 is respectively 32.7% and 32.5%.

2.4.3 Analysis of the results of test set T_{3tc}

In this section, we analyze the results for test set T_{3tc}, described in Table A.3. Furthermore, we represent these results as performance profile graphs, in Figures 2.7 to 2.10.

In Figure 2.7, we can observe a first and important fact: for instances of test set T_{3tc}, all three models produce the same LP bound. Moreover, as it can be seen in Table A.3, CPLEX's automatic cuts do not yield any improvements for these bounds, for any of the models. Note however, that even though there are some instances with large Gap_{LP}, this result is mainly explained by the large portion of instances of T_{3tc} (80%) that have null Gap_{LP}.

The results for the Gap_{LP} help us interpret the results in Figure 2.8, regarding

the MIP solving times. While for test set T_{rand}, RDMFM was patently the "fastest" model for the more "demanding" instances, due to its better lower bounds; for T_{3tc} the results are less clear. When implemented in CPLEX, RDM seems to be more efficient formulation for most instances. However, RDMFM is able to solve more instance in the time limit of one hour, which implies for the more "demanding" instances, it might perform better. The average Gap_{end} for instances unsolved both by RDM and RDMFM is very similar, 16% and 15% respectively.

The results in Figures 2.9 and 2.10 might shine some light on the results discussed above. The LPs of RDM are significantly faster to solve than the LPs from RDMFM; however, the B&B trees yielded by RDMFM tend to be slimmer than the trees yielded by RDM. The large B&B trees and the long LP solving times also help explain the overall bad performance of CPLEX implementing MFM.

One final observation, is that the instances of T_{3tc} are seemingly easier to solve than the ones of T_{rand}, with the average Gap_{LP} and solving times being clearly lower. This is likely explained by the less dense structure of networks that follow the 3-Tier Cisco architecture.

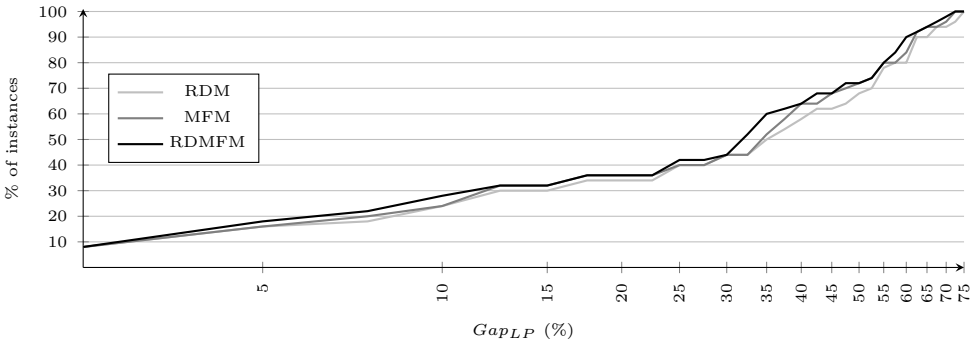


Figure 2.2: TE-MSTP, T_{rand}: performance profile of the Gap_{LP} (%).

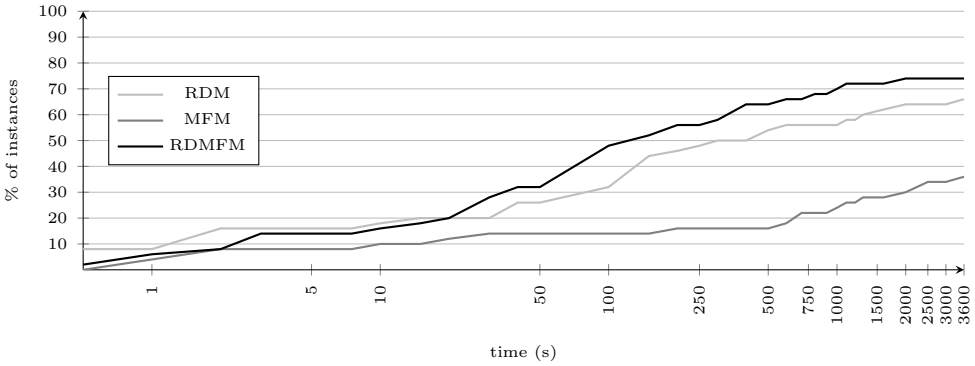


Figure 2.3: TE-MSTP, T_{rand}: performance profile of the MIP solving time (s).

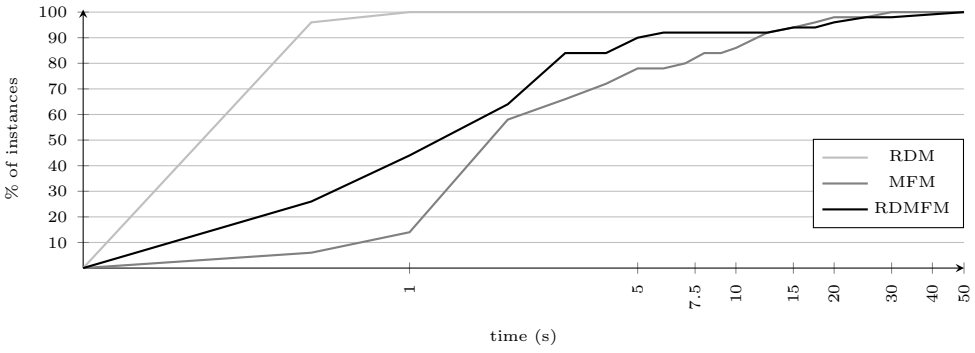


Figure 2.4: TE-MSTP, T_{rand}: performance profile of the LP solving time (s).

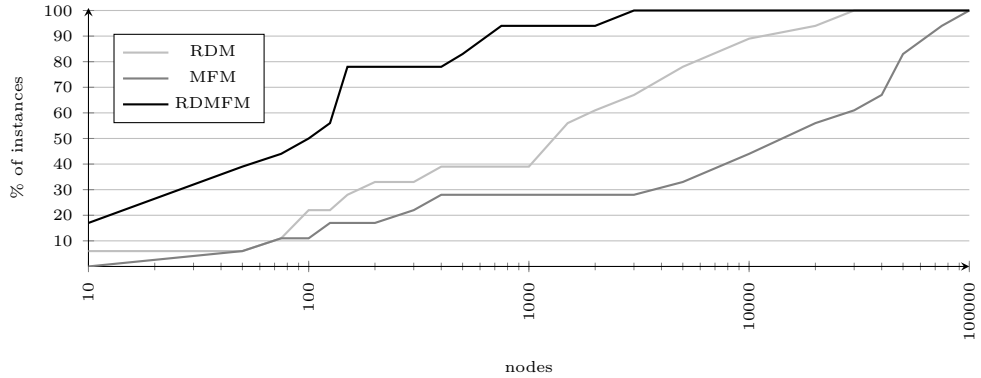


Figure 2.5: TE-MSTP, T_{rand}: performance profile of the B&B tree nodes.

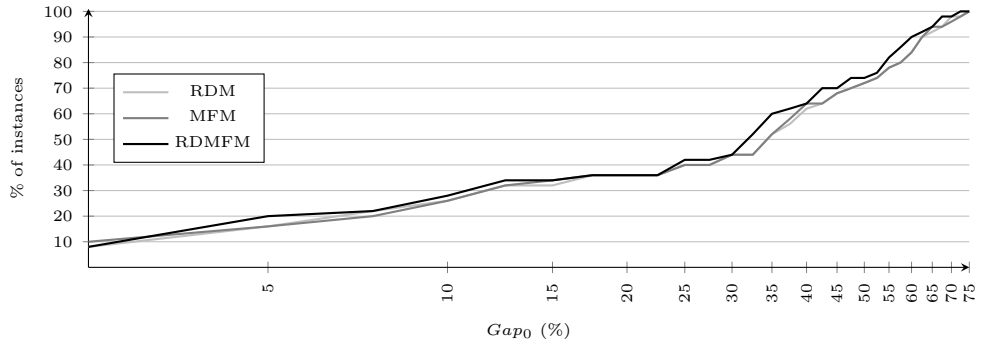


Figure 2.6: TE-MSTP, T_{rand}: performance profile of Gap_0 (%).

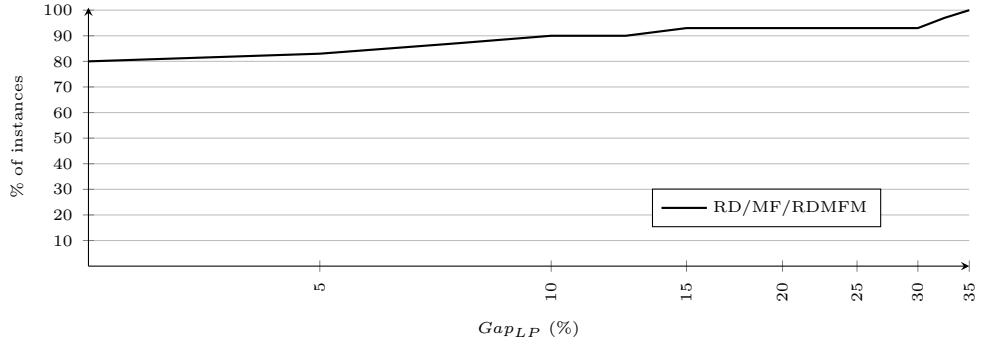


Figure 2.7: TE-MSTP, T_{3tc}: performance profile of Gap_{LP} (%).

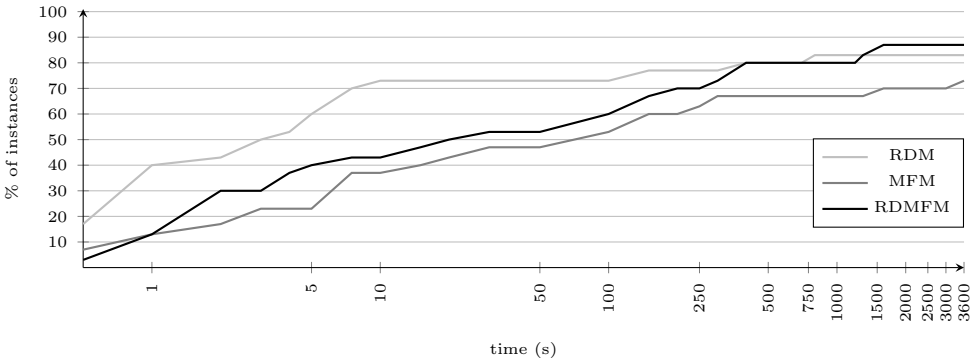


Figure 2.8: TE-MSTP, T_{3tc}: performance profile of the MIP solving time (s).

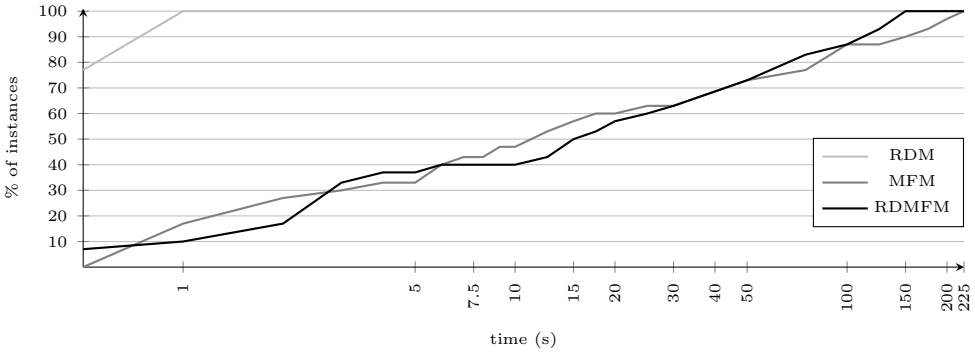


Figure 2.9: TE-MSTP, T_{3tc}: performance profile of the LP solving time (s).

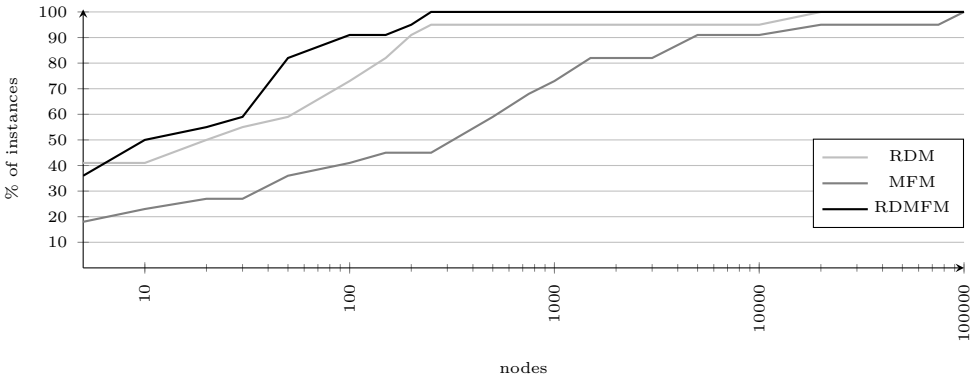


Figure 2.10: TE-MSTP, T_{3tc}: performance profile of the B&B tree nodes.

2.5 B&C algorithm

In this section, we propose a B&C algorithm for the TE-MSTP problem. In this algorithm, the cuts used at the different nodes of the B&B tree are Benders' cuts. For that purpose, we solve at each node a Benders' decomposition algorithm, as described in Section 1.6. In Section 2.5.1 we present a Benders' decomposition of RDMFM. In Section 2.5.2 we describe the details of the implementation of the B&C algorithm, and analyze the results of computational experiments, done in order to evaluate the efficiency of this algorithm.

2.5.1 Benders' decomposition

In this section, we discuss a decomposition of RDMFM based on Benders' method. First, for the sake of clarity, we repeat the original model in Formulation 2.15. As the results of computational experiments, presented in the last section, imply, this model benefits from better lower bounds when compared with the other models. However, the large LPs can be slow to solve, especially when compared with the LPs of RDM. Naturally, this too has an impact on the MIP solving time. It is easy to see that what makes the LPs large, are the x variables in RDMFM, that make it such that this model has $\mathcal{O}(n^4 \cdot |T|)$ variables and constraints, whereas RDM only has $\mathcal{O}(n^3 \cdot |T|)$.

As such, we fix variables z, w and U^{max} in RDMFM, and we consider the slave problem RDMFM$_{LP}(\bar{z}_{ij}^{ut}, \bar{U}^{max})$, described in Formulation 2.16. This sub-problem is, in essence, a feasibility problem, that aims at verifying if it is possible to route each VLAN's demands, based on the tree design, given by $\bar{z}$, and the edge capacity, implied by $\bar{U}^{max}$. We consider this sub-problem as a maximization problem, so as to have RDMFM$_{LP}(\bar{z}_{ij}^{ut}, \bar{U}^{max})$ in the standard form, while keeping constraints (2.16d-2.16f) in their original format. Note that we do not explicitly enforce an upper bound on variables x, as this is implied by constraints (2.16e-2.16f).

Let α, γ, ζ and η be dual variables associated with the constraints of RDMFM$_{LP}(\bar{z}_{ij}^{ut}$, $\bar{U}^{max})$. α links to constraints (2.16b-2.16c), η to constraints (2.16d), γ to constraints (2.16e), and ζ to (2.16f). Then, RDMFM$_D(\bar{z}_{ij}^{ut}, \bar{U}^{max})$ is the dual of RDMFM$_{LP}(\bar{z}_{ij}^{ut}$, $\bar{U}^{max})$; see Formulation 2.17. By solving this dual problem, we are able to infer Benders' cuts $\beta_{\bar{U}^{max},\bar{z}}(U^{max}, z)$, that we iteratively add to the master problem RDMFM$_M$ (see Formulation 2.18), according to the procedure described in Algorithm 1.1, back in Section 1.6. $\beta_{\bar{U}^{max},\bar{z}}(U^{max}, z)$ will be of the form $\bar{\boldsymbol{\eta}} \cdot U^{max} + \sum_{t \in T} \sum_{u,v \in N: u<v} \sum_{(i,j) \in A: j \neq u, i \neq v} \big(\bar{\gamma}_{ij}^{uvt} z_{ij}^{ut} + \bar{\zeta}_{ij}^{uvt} z_{ji}^{vt}\big) + \bar{\boldsymbol{\alpha}}$, where $\bar{\boldsymbol{\eta}} = \sum_{e \in E} C_e \bar{\eta}_e$, and $\bar{\boldsymbol{\alpha}} = \sum_{t \in T} \sum_{u,v \in N: u<v} \bar{\alpha}_u^{uvt}$.

Note that if RDMFM$_{LP}$ is infeasible, RDMFM$_D$ is an unbounded problem. As it was mentioned in Section 1.6, finding extreme rays can be a complicated task from a computational point of view. As such, we fix the objective of RDMFM$_D$ to a negative normalization point. We remark that, due to (2.5a) in RDMFM$_M$, the network given to RDMFM$_{LP}$ will always be connected. As such, the only reason why the routing problem in RDMFM$_{LP}$ might be infeasible is due to the lack of edge capacity implied by $\bar{U}^{max}$.

$$\min_{U^{max},w,x,z} \quad U^{max} \tag{2.11}$$

s.t

$$\sum_{a\in\delta^-(j)} z_a^{ut} = 1, \qquad u,j \in N : u \neq j, t \in T \tag{2.5a}$$

$$z_{ij}^{ut} + z_{ji}^{us} = w_e^t, \qquad u \in N, e = \{i,j\} \in E, t \in T \tag{2.5b}$$

$$\sum_{a\in\delta^+(u)} x_a^{uvt} = 1, \qquad u,v \in N : u < v, t \in T \tag{2.2a}$$

$$\sum_{a\in\delta^-(j)} x_a^{uvt} - \sum_{a\in\delta^+(j)} x_a^{uvt} = 0, \qquad u,v,i \in N : u < v, i \neq \{u,v\}, t \in T \tag{2.2b}$$

$$\sum_{t\in T} \sum_{u,v\in N:u<v} \rho_{uv}^t (x_{ij}^{uvt} + x_{ji}^{uvt}) \leq C_e \cdot U^{max}, \qquad e = \{i,j\} \in E \tag{2.8}$$

$$x_{ij}^{uvt} \leq z_{ij}^{ut}, \qquad (i,j) \in A, u,v \in N : u < v, j \neq u, i \neq v, t \in T \tag{2.13}$$

$$x_{ij}^{uvt} \leq z_{ji}^{vt}, \qquad (i,j) \in A, u,v \in N : u < v, j \neq u, i \neq v, t \in T \tag{2.14}$$

$$x_{ij}^{uvt} \in \{0,1\}, \qquad (i,j) \in A, u,v \in N : u < v, j \neq u, i \neq it \in T \tag{2.2e}$$

$$z_{ij}^{ut} \in \{0,1\}, \qquad u \in N, (i,j) \in A : j \neq u, t \in T \tag{2.5c}$$

$$w_e^t \in \{0,1\}, \qquad e \in E, t \in T \tag{2.1}$$

$$0 \leq U^{max} \leq 1 \tag{2.10}$$

Formulation 2.15: TE-MSTP problem: RDMFM.

Accordingly, if we were to also remove variables U^{max} from the master, and leave them to the slave problem, RDMFM$_{LP}$ would always be feasible, and the corresponding dual bounded. As such, we add constraints $\boldsymbol{\eta} = 1$ to the dual sub-problem; this is the term of the dual objective function associated with variables U^{max}. This makes it such that the Benders' cut will be $U^{max} + \sum_{t\in T} \sum_{u,v\in N:u<v} \sum_{(i,j)\in A:j\neq u,i\neq v} \left(\bar{\gamma}_{ij}^{uvt} z_{ij}^{ut} + \bar{\zeta}_{ij}^{uvt} z_{ji}^{vt}\right) \geq -\bar{\boldsymbol{\alpha}}$. As $\bar{\boldsymbol{\alpha}}$ is typically negative, this normalization tends to push the value of U^{max} up, along the iterations of the Benders' algorithm.

We conclude this section with a remark about alternative reformulations of RDMFM. Naturally, it is possible to decompose RDMFM in a different way; namely, by fixing instead the x and U^{max} variables in the master problem, leaving the z and w variables to the slave problem. This corresponds to routing the demands in the master, and ensuring the tree topology of each VLAN in the slave problem. This has the advantage that it would allow us to decompose the slave problem by VLAN. Nevertheless, the order of the cardinality of integer variables would remain unchanged, with respect to RDMFM. Preliminary tests show that this solution does not improve on the proposed decomposition. Alternatively, a Dantzig-Wolfe reformulation [DW60] of RDMFM could be used, and implemented in a branch-and-price algorithm [BJN$^+$98]. Regardless, it is

easy to understand that the pricing problem would be similar to the OCST problem, which is known to be very hard to solve (see Section 1.4.2). Indeed, in [Cre12], the authors investigate the efficiency of using such an algorithm for the problem proposed in [SdSA+09] (see Section 1.5 for a review of this problem), only to conclude that it does not perform well.

$$\max_{x} \quad 0 \tag{2.16a}$$

s.t

$$\sum_{a\in\delta^+(u)} x_a^{uvt} = 1, \qquad u,v \in N : u < v, t \in T \tag{2.16b}$$

$$\sum_{a\in\delta^-(j)} x_a^{uvt} - \sum_{a\in\delta^+(j)} x_a^{uvt} = 0, \qquad u,v,i \in N : u < v, i \neq \{u,v\}, t \in T \tag{2.16c}$$

$$\sum_{t\in T} \sum_{u,v\in N:u<v} \rho_{uv}^t (x_{ij}^{uvt} + x_{ji}^{uvt}) \leq C_e \cdot \bar{U}^{max}, \qquad e = \{i,j\} \in E \tag{2.16d}$$

$$x_{ij}^{uvt} \leq \bar{z}_{ij}^{ut}, \qquad (i,j) \in A, u,v \in N : u < v, j \neq u, i \neq v, t \in T \tag{2.16e}$$

$$x_{ij}^{uvt} \leq \bar{z}_{ji}^{vt}, \qquad (i,j) \in A, u,v \in N : u < v, j \neq u, i \neq v, t \in T \tag{2.16f}$$

$$x_{ij}^{uvt} \geq 0, \qquad (i,j) \in A, u,v \in N : u < v, j \neq u, i \neq it \in T \tag{2.16g}$$

Formulation 2.16: TE-MSTP problem: RDMFM$_{LP}(\bar{z}_{ij}^{ut}, \bar{U}^{max})$.

$$\min_{\alpha,\beta,\gamma,\zeta} \quad \sum_{t\in T} \sum_{u,v\in N:u<v} \left[\alpha_u^{uvt} + \sum_{(i,j)\in A:j\neq u,i\neq v} \left(\bar{z}_{ij}^{ut} \gamma_{ij}^{uvt} + \bar{z}_{ji}^{vt} \zeta_{ij}^{uvt} \right) \right] + \sum_{e\in E} C_e \cdot \bar{U}^{max} \eta_e \tag{2.17a}$$

s.t

$$\alpha_i^{uvt} - \alpha_j^{uvt} + \gamma_{ij}^{uvt} + \zeta_{ij}^{uvt} + \rho_{uv}^t \eta_{\{i,j\}} \geq 0, (i,j) \in A, u,v \in N : u < v, j \neq u, i \neq v, t \in T \tag{2.17b}$$

$$\eta_e \geq 0, \qquad e \in E \tag{2.17c}$$

$$\gamma_{ij}^{uvt} \geq 0, \qquad (i,j) \in A, u,v \in N : u < v, j \neq u, i \neq v, t \in T \tag{2.17d}$$

$$\zeta_{ij}^{uvt} \geq 0, \qquad (i,j) \in A, u,v \in N : u < v, j \neq u, i \neq v, t \in T \tag{2.17e}$$

Formulation 2.17: TE-MSTP problem: RDMFM$_D(\bar{z}_{ij}^{ut}, \bar{U}^{max})$.

2.5.2 Computational experiments for the B&C algorithm

In this section, we discuss the implementation Benders' decomposition proposed in the previous section in a B&C framework. This algorithm was implemented in Java, and CPLEX was used to solve both the master and the slave problems. CPLEX distinguishes two types of cuts: `user cuts` and `lazy constraints`. The first are used to cut off

$$\min_{U^{max},w,z} \quad U^{max} \tag{2.11}$$

s.t

$$\sum_{a\in\delta^-(j)} z_a^{ut} = 1, \qquad u,j \in N : u \neq j, t \in T \tag{2.5a}$$

$$z_{ij}^{ut} + z_{ji}^{us} = w_e^t, \qquad u \in N, e = \{i,j\} \in E, t \in T \tag{2.5b}$$

$$\beta_{U^{\bar{m}ax k'},\bar{z}^{k'}}(U^{max}, z) \geq 0, \qquad k' = 1...k \tag{2.18a}$$

$$z_{ij}^{ut} \in \{0,1\}, \qquad u \in N, (i,j) \in A : j \neq u, t \in T \tag{2.5c}$$

$$w_e^t \in \{0,1\}, \qquad e \in E, t \in T \tag{2.1}$$

$$0 \leq U^{max} \leq 1 \tag{2.10}$$

Formulation 2.18: TE-MSTP problem: RDMFM$_M^k$.

fractional solutions, by redefining the polyhedron implied by the LP relaxation of the model. These cuts do not, however, exclude feasible integer solutions. By contrast, `lazy constraints` are employed when the input integer model is incomplete and it allows integer infeasible solutions to be considered. As such, `lazy constraints` are set, such that they cut off these solutions. The search for unsatisfied `user cuts` is done every time CPLEX finishes processing a B&B tree node, whereas unsatisfied `lazy constraints` are added to the model whenever an integer feasible solution is found. We implement our Benders' decomposition method both in the `user cuts callback` and in the `lazy constraints callback`. Notwithstanding, we limit the number of `user cuts` added in each node of the tree to 20; with the exception of the root node, where we embed in our B&C algorithm the in-out cut loop proposed in [FLS15], and presented in Section 1.6.

Recall that the in-out cut loop procedure requires one to define *a priori*, a stabilizing point. For this purpose, we implemented a greedy heuristic algorithm that quickly generates a solution to the TE-MSTP problem, which is "fed" to the in-out cut loop procedure. This greedy heuristic is described in Algorithm 4.1. We consider a VLAN at a time, following an order based on the sum of all demands assigned to each VLAN. At each iteration of the algorithm, we create a cost matrix $\mathcal{W}_e$, that penalizes heavily loaded edges. We then run Prim's algorithm with respect to this cost matrix, to calculate the minimum cost spanning tree. Given a tree, we can assign the appropriate values for the respective w and z. We also update the total load on each edge, accordingly. We repeat this procedure for every VLAN. In the end, we calculate the value for the worst-case edge utilization of the solution generated by the heuristic. Note that it is possible that this solution is not feasible for the TE-MSTP problem as we defined it, in the sense that U^{max} can be greater than 1. Nevertheless, this is not a problem, as we can simply relax the upper bound on U^{max} and have an equivalent problem, for which the Algorithm 4.1 always provides a feasible solution.

Algorithm 2.1: Greedy heuristic for the TE-MSTP problem.

$\mathcal{T} \leftarrow \{1, ..., |T|\}$
$\mathcal{F}_e \leftarrow 0, e \in E$
while $|\mathcal{T}| > 0$ **do**
    $t \leftarrow \arg\max_{t \in \mathcal{T}} \sum_{k \in K^t} \rho_k$
    $\mathcal{T} \leftarrow \mathcal{T} \backslash \{t\}$
    $\mathcal{W}_e \leftarrow \frac{\mathcal{F}_e}{1.1C_e - \mathcal{F}_e}, e \in E$
    $\mathcal{W}_e \leftarrow \infty, e \notin E$
    Infer minimum cost spanning tree, with respect to $\mathcal{W}$
    Assign values for w^t and z^t accordingly
    Update loads $\mathcal{F}$
Assign value for U^{max} accordingly

We ran the algorithm for all instances of test sets T_{rand} and T_{3tc}, described in Section 2.4.1. The results are described in Tables A.4 and A.5, in Appendix A. It is clear that our B&C algorithm performs very poorly; its solving times are far greater than the ones of CPLEX solving the MIPs *per se*. We believe that there are two main reasons for the inefficiency of this algorithm. First, the Benders' cuts seem to be very weak, and for most instances a lot of these cuts are necessary to improve the lower bound. This is made clear by the values of the Gap_{io}; the Gap_{io} measures the relative gap between the worst-case utilization of the best known solution (value provided by the experiments described in the previous section), and the lower bound at the end of the root node, after the in-out cut loop is terminated. For the majority of the instances, we are unable to increase this lower bound at all, and its value is 0. Secondly, RDMFM$_D$ is not decomposable, and therefore, solving it is not sufficiently fast as to make the process of generating many cuts efficient. In our remarks in Chapter 6, we expand on this idea, and discuss on why this is a problem common to other capacitated network flow problems.

2.6 Summary and remarks

In this chapter, we studied the TE problem for the MSTP, that was first proposed in [HDBF11, Ho12] and that we denote as the TE-MSTP problem. We showed that this problem is $\mathcal{NP}$-hard (see Section 2.1).

We proposed three different MIP formulations: RDM, MFM and RDMFM. We compared the strength of the LP relaxation of these models, and we showed that $B_{LP}(RDMFM) \geq B_{LP}(MFM) \geq B_{LP}(RDM)$.

Moreover, we presented a comprehensive array of computational experiments that were done, in order to further compare the proposed formulations. These experiments seem to point out RDMFM as the most promising formulation, when used with CPLEX: the B&B trees are typically more compact; the gaps at the root node are, in most cases, smaller; and the more "difficult" instances are solved faster.

The results of the computational experiments also emphasize the difficulty of solving

the TE-MSTP optimally, even for relatively small instances. This is evidenced not only by the lengthy computation times, but also by the often weak LP bounds. As such, we implemented a B&C algorithm, based on a Benders' decomposition of RDMFM. Unfortunately, this algorithm did not prove to be efficient, being much slower than the standard B&B algorithm of CPLEX, when applied to the MIP of RDMFM. This seems to be a result of weak Benders' cuts, and the fact that the dual slave problem is not decomposable.

We hypothesize that weak lower bounds obtained by the LP relaxation of our proposed models might be related to the "min-max" structure of the objective function, as the same has been found to be true for other problems with this type of objective. As such, in the next chapters we consider other MSTP network design problems, with different objective functions.

Chapter 3

MSTP: minimization of total load

The results analyzed in the previous section emphasize the difficulty of solving the TE-MSTP problem optimally, even for relatively small instances. This is evidenced not only by the lengthy computation times, but also by the often weak LP relaxations. This might be partially explained by the structure of the problem's objective: it is not uncommon for a problem with a "min-max" objective function to be associated with large Gap_{LP} values, when compared with the "min-average" or "min-sum" counterpart. An example of this phenomenon can be observed in [GPdS11].

In this chapter, we consider a similar problem to the TE-MSTP problem, where instead of minimizing the worst-case edge utilization, we minimize a weighted sum of all edge loads, or total load; all other requirements are kept. This problem can be seen as an extension of the OCST problem, reviewed in Section 1.4.1, with multiple spanning trees, and capacities on the edge loads. In this sense, we name this problem the capacitated optimum communication multiple spanning tree (COCMST) problem. Recall that the OCST problem is $\mathcal{NP}$-hard; thus, the COCMST problem is also $\mathcal{NP}$-hard.

Let us assume that we use the same edge capacity and traffic demand values for the COCMST problem, as we do for the TE-MSTP problem. Now, let us also define the COCMST(Λ) problem, as the COCMST problem where all these capacities are multiplied by Λ. We can observe that the COCMST(Λ) problem relates to the TE-MSTP problem in an interesting way: any feasible solution to this first problem, is also feasible for the second one, with a guaranteed worst-case objective value of Λ.

In Section 3.1, we show how we adapt the formulations proposed in the previous chapter to solve the COCMST problem. We test the performance of this formulations in Section 3.2. In Section 3.3, we propose a method to solve the TE-MSTP problem, based on the sequential solving of COCMST sub-problems. In Section 3.4 we analyze the efficiency of this method, by delving into the results of computational experiments. Finally, in Section 3.5, we summarize the work described in this chapter, and present some conclusions.

3.1 Problem formulation

It is easy to adapt the formulations proposed in Section 2.2, such that they solve the COCMST problem. We define as RDM-t, to the adaptation of RDM as defined in Table 2.1, but with objective function (3.1a) instead of (2.11), and constraints (3.1b) in substitution of constraints (2.9).

$$\min \sum_{t\in T, u\in N, \{i,j\}\in E} c_{\{i,j\}}(l_{ij}^{us} + l_{ji}^{us}) \tag{3.1a}$$

$$\sum_{t\in T}\sum_{u\in N}(l_{ij}^{ut} + l_{ji}^{ut}) \leq \Lambda \cdot C_e, \qquad e = \{i,j\} \in E \tag{3.1b}$$

Formulation 3.1: COCMST problem: RDM-t.

Similarly, we define MFM-t and RDMFM-t to the adaptation of MFM and RDMFM, respectively, such that the objective function is instead (3.2a), and constraints (2.8) are replaced by constraints (3.2b). Note that variable U^{max} becomes obsolete in RDM-t, MFM-t, and RDMFM-t.

$$\min \sum_{t\in T, u\in N, \{i,j\}\in E} c_{\{i,j\}}\rho_{uv}^{t}(x_{ij}^{uvt} + x_{ji}^{uvt}) \tag{3.2a}$$

$$\sum_{t\in T}\sum_{u,v\in N: u<v} \rho_{uv}^{t}(x_{ij}^{uvt} + x_{ji}^{uvt}) \leq \Lambda \cdot C_e, \qquad e = \{i,j\} \in E \tag{3.2b}$$

Formulation 3.2: COCMST problem: capacity constraints for the aggregated-flows case.

3.2 Computational experiments for the COCMST problem

In order to evaluate the performance of formulations RDM-t, MFM-t and RDMFM-t in solving the COCMST problem, we present in this section the results of computation experiments, that were done using test sets T_{rand} and T_{3tc}, described in Section 2.4. We consider traffic costs $c_e = 1$ for every edge $e \in E$. Moreover, we define $\Lambda = U^{max*} + \epsilon$, for $\epsilon = \{-0.05, 0.01, 0.05, 0.2\}$. U^{max*} is the value of the best integer solution found for each instance, with respect to the TE-MSTP problem. For ease of notation, we denote as the COCMST(0.01^{ϵ}) problem, when $\epsilon = 0.01$; we use an analogous notation for other values of ϵ. Our objective is to test the performance of our models in solving the COCMST problem when capacities are tighter ($\epsilon = 0.05$ and $\epsilon = 0.01$) and looser ($\epsilon = 0.2$), and when the problem is infeasible ($\epsilon = -0.05$). In the following sections we analyze the

results of the computational experiments done with each of these values. Specifications about the machine and framework used to run the experiments are described in Section 2.4.

3.2.1 Analysis of the results for $\epsilon = 0.2$

In this section, we analyze the results for the computational experiments done for the COCMST(0.2^ϵ) problem. The complete results are detailed in Tables B.1 to B.3 of Appendix B.

In Figure 3.1, we depict the performance profile for the solving time of the MIPs of each model. This graph reveals that by using RDM-t, MFM-t and RDMFM-t to solve the COCMST(0.2^ϵ) problem, we obtain similar trends, than by using the corresponding models to solve the TE-MSTP problem. In particular, choosing MFM-t clearly leads to larger solving time, than by choosing RDM-t and RDMFM-t. The results for the two latter models are mixed: Using CPLEX with RDM-t tends to be faster for "easier" instances, whereas using it with RDMFM-t allows us to solve more instances in the one hour time limit.

Looking at the Gap_{LP}, illustrated in the performance profile in Figure 3.2 one can be surprised by the reasonably good performance of RDM-t, described in the previous paragraph: the lower bounds obtained by solving the LP relaxation of RDM-t are much lower than the ones obtained by solving the LP relaxation of RDMFM-t. Namely, the average Gap_{LP} for RDM-t is 25.6% for instances of T_{rand} and 10.8% for instances of T_{3tc}, whereas the average Gap_{LP} for RDMFM-t is just 3.6% for instances of T_{rand} and 0.6% for instances of T_{3tc}. The efficiency of CPLEX using RDM-t in solving the MIPs of both test sets can, however, be party explained by the very fast solving of the LPs, as seen in Figure 3.3. While solving all the LPs of RDM-t is done under 1 second, solving the LPs of RDMFM-t can take up to 7 seconds for instances of T_{rand} and 125 seconds for instances of T_{3tc}. Moreover, the aforementioned weak LP bounds of RDM-t are significantly improved by CPLEX's automatically generated cuts, which narrow the distance between the average Gap_0 for RDM-t (5.6% for T_{rand} and 1.1% for T_{3tc}) and for RDMFM-t (2.9% for T_{rand} and 0.4% for T_{3tc}).

Finally, one interesting remark is that, conversely to what happened with the TE-MSTP problem, for the COCMST(0.2^ϵ) problem the instances of T_{rand} are easier to solve than the instances of T_{3tc}.

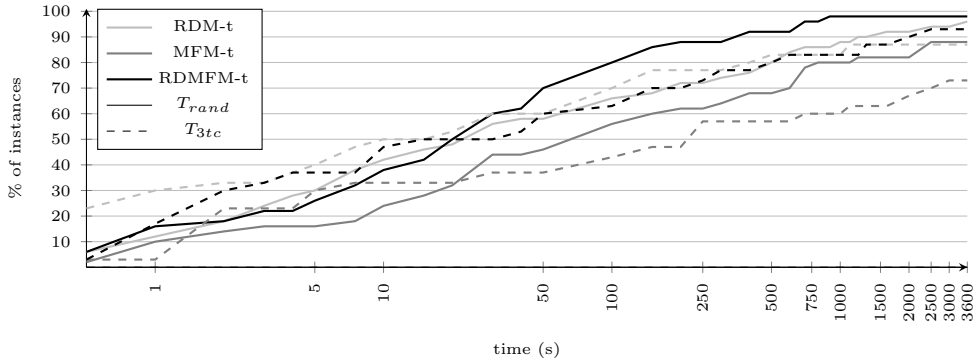


Figure 3.1: COCMST(0.2^{ϵ}): performance profile of the MIP solving time (s).

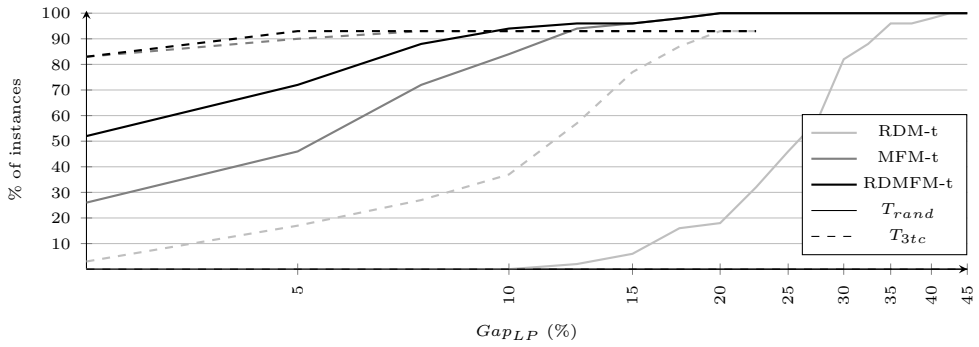


Figure 3.2: COCMST(0.2^{ϵ}): performance profile of Gap_{LP} (%).

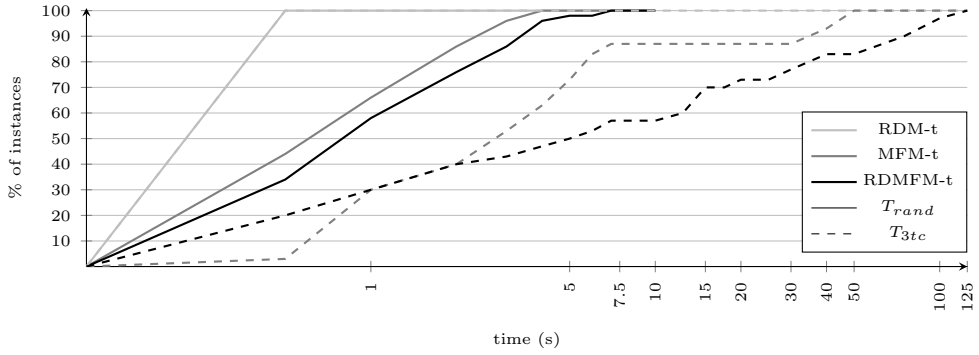


Figure 3.3: COCMST(0.2^{ϵ}): performance profile of the LP solving time (s).

3.2.2 Analysis of the results for $\epsilon = 0.05$

In this section, we consider tighter capacities, by setting $\epsilon = 0.05$. The results of the computational experiments for the COCMST(0.05^ϵ) problem can be seen in Tables B.4 to B.6, in Appendix B. They show that for for this value of ϵ, the relative performance of our proposed models, in the considered criteria, is comparable to the one observed for $\epsilon = 0.2$. Ultimately, this results in a similar relative efficiency in solving MIPs of the instances in T_{rand} and T_{3tc}, as it can be observed in Figure 3.4: CPLEX with MFM-t is very slow to solve the MIPs, CPLEX with RDM-t performs well for "easy" instances, but in the end "loses" to CPLEX with RDMFM-t, when it comes to solving more "difficult" instances.

However, the relative efficiency of RDMFM-t in solving instances of the COCMST (0.05^ϵ) problem, when compared to RDM-t, is even more significant that when $\epsilon = 0.2$: While for $\epsilon = 0.2$, CPLEX with RDMFM-t solves only 2% (6%) more instances of T_{rand} (T_{3tc}) than CPLEX with RDM-t; for $\epsilon = 0.05$ CPLEX with RDMFM-t solves more 16% (10%) instances of T_{rand} (T_{3tc}). It is interesting to see that this happens despite CPLEX with RDM-t still being significantly faster than CPLEX with RDMFM-t in solving LPs, and despite the distance between the lower bounds obtained at the end of root node for both models being narrower for the COCMST(0.05^ϵ) problem, than it was for the COCMST(0.01^ϵ) problem.

3.2.3 Analysis of the results for $\epsilon = 0.01$

In this section, we analyse the results of computational experiments done for the COCMST (0.01^ϵ) problem, presented in Tables B.7 to B.9 of Appendix B. In this case, the capacities are very tight; in all likelihood, in order to be able to route all the demands, it is necessary to use the optimal designs with respect to the corresponding TE-MSTP problem.

Figure 3.5 depicts the performance profile of the MIP solving time for every model. Again, RDMFM-t is able to solve more instances of both test sets in the time limit. For instances of T_{rand}, the difference between the performance of RDMFM-t and RDM-t continues to broaden, with the first model solving more 18% instances than the latter. Notwithstanding, for T_{3tc} we observe the converse, in the sense that the efficiency of RDM-t is relatively similar to the one of RDMFM-t.

Another important remark is that the COCMST(0.01^ϵ) problem seems to be easier to solve than the TE-MSTP problem for instances of T_{rand}, but harder for instances of T_{3tc}. For the COCMST(0.01^ϵ) problem, RDMFM-t is able to solve 88% of the instances of T_{rand}, whereas for the TE-MSTP problem, the corresponding model is only able to solve 74%. Regardless, for instances of T_{3tc} RDMFM-t only solves 80%, falling short of the mark of 87% TE-MSTP instances solved by RDMFM. The Gap_{LP} in the COCMST(0.01^ϵ) problem, whose performance profile can be seen in Figure 3.6, are however, smaller for both test sets. While for the TE-MSTP problem, RDMFM has Gap_{LP} up to 71% (35%) for instances of T_{rand} (T_{3tc}), the largest gap observed for RDMFM-t and the COCMST(0.01^ϵ) problem is of 34% (8%).

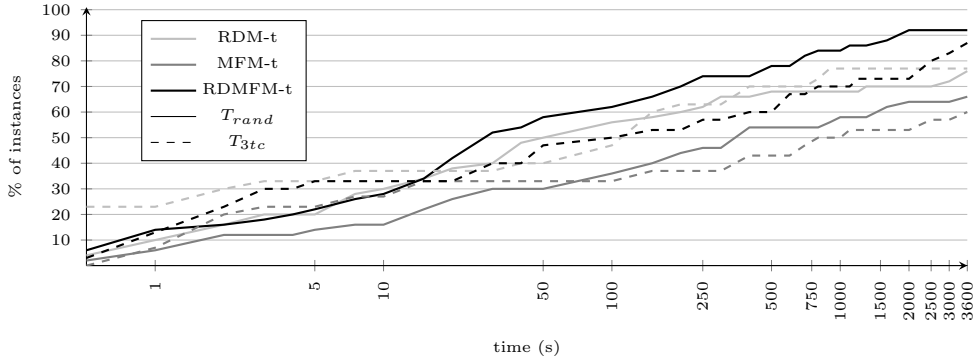


Figure 3.4: COCMST(0.05^{ϵ}): performance profile of the MIP solving time (s).

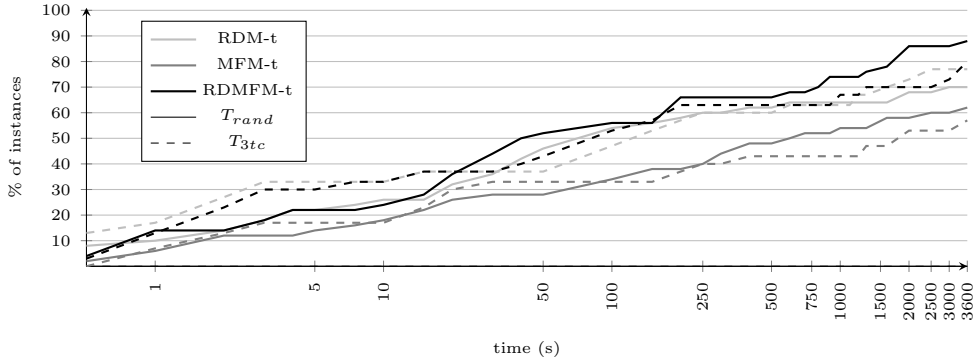


Figure 3.5: COCMST(0.01^{ϵ}): performance profile of the MIP solving time (s).

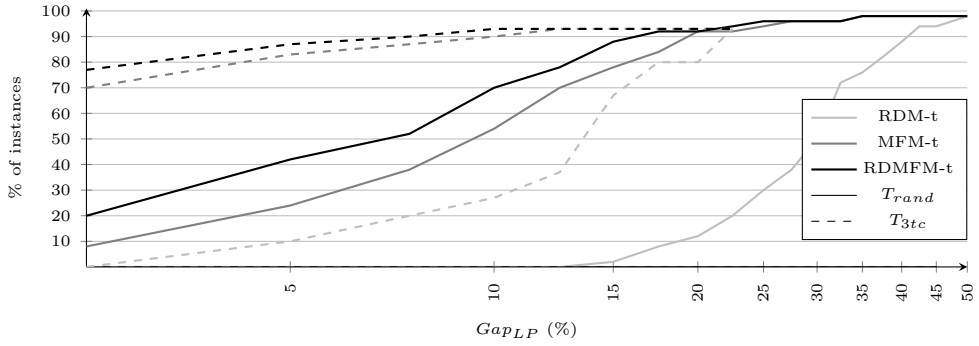


Figure 3.6: COCMST(0.01^{ϵ}): performance profile of Gap_{LP} (%).

3.2.4 Using the COCMST problem to find feasible solutions for the TE-MSTP problem

As it was mentioned at the beginning of this chapter, any feasible solution for the COCMST(Λ) problem is also feasible for the TE-MSTP problem. Moreover, we know that this solution will have a worst-case edge utilization not larger than Λ. In this section, we continue to interpret the results of computational experiments done for the COCMST(Λ) problem ($\Lambda = U^{max*} + \{0.01, 0.05, 0.2\}$), and presented in Appendix B. Nevertheless, we now analyze how the solutions obtained for the COCMST problem evaluate in the objective of TE-MSTP problem. In this analysis, we omit the results of MFM-t, as they are clearly worse than the ones obtained for RDM-t and RDMFM-t. Let $\hat{\mathcal{S}}$ be the optimal solution for the COCMST problem, and $U^{\hat{max}}$ its *de facto* worst-case utilization; $U^{\hat{max}}$ can be calculated, for example, by looking at the minimum slack of constraints (3.1b) and (3.2b). We define $Gap^*_U = \frac{U^{\hat{max}} - U^{max*}}{U^{\hat{max}}}$, and we represent the performance profiles of this measure in Figures 3.7 and 3.8. To keep the comparison consistent, we only consider instances that were solved using both RDM-t and RDMFM-t. The results for both models are very similar (for T_{rand}), if not the same (T_{3tc}), which is unsurprising, as it is unlikely that there are many alternative optimal solutions for the COCMST problems. Note that some Gap^*_U are negative. This happens in some instances whose corresponding TE-MSTP problem had not been solved to optimality; via the COCMST problem, we were able to find better solutions than the ones obtained by solving the TE-MSTP problem. In same cases, by solving the COCMST problem we found solutions that were almost 15% better than the best had obtained, when solving the TE-MSTP problem.

Note, however, that we do not need to solve the COCMST(Λ) problem to optimality to get a solution which is feasible for the TE-MSTP problem and that has a guaranteed worst-case edge utilization of Λ. In our experiments, we also tested setting CPLEX's settings such that it stops the solving procedure after finding a first feasible solution. Figures 3.9 and 3.10 depict the performance profiles for the time it takes CPLEX to find a first feasible solution of instances of T_{rand} and T_{3tc}, respectively. The results are rather mixed, but they seem to indicate that RDMFM is faster in finding feasible solutions in more "difficult" instances.

Next, we compare the "quality" of these solutions, with respect to its worst-case edge utilization. We define Gap^1_U, analogously to Gap^*_U (see above), but considering the first feasible solution obtained by CPLEX. The results are depicted in Figures 3.11 and 3.12. Once again, the performance of both models is very close.

It is noteworthy, however, that the average worst-case edge utilization of these first-found feasible solutions is not much higher than the one observed for the optimal solutions. Moreover, as seen in Table 3.1, as the capacities tighten, the average Gap^1_U is actually smaller than the average Gap^*_U.

Lastly, let us recall the TE-MSTP(Λ) decision problem, introduced in Section 2.1. Note that solving this decision problem for a given Λ, is very similar to finding a feasible solution for the COCMST(Λ) problem. In fact, the only difference is that in the latter we have an objective function. As such, we also define model RDM-0 where constraints

	T_{rand}						T_{3tc}					
	$\epsilon = 0.2$		$\epsilon = 0.05$		$\epsilon = 0.01$		$\epsilon = 0.2$		$\epsilon = 0.05$		$\epsilon = 0.01$	
	RDM	RDMFM	RDM	RDMFM	RDM	RDMFM	RDM	RDMFM	RDM	RDMFM	RDM	RDMFM
Gap_U^*	11.2	11.4	1.5	1.4	-0.4	-0.4	12.1	12.0	2.7	2.7	0.3	0.3
Gap_U^1	13.6	12.8	1.4	1.7	-0.5	-0.6	12.0	12.7	2.8	2.5	0.2	0.2
Gap_U^D	15.7	16.1	2.0	2.1	0.0	0.0	14.0	14.6	3.1	2.6	0.1	0.0

Table 3.1: Comparison between average Gap_U^*, Gap_U^1 and Gap_U^D.

(2.9) are replaced by (3.1b) and model RDMFM-0 where constraints (2.8) are replaced by (3.2b). Furthermore, for implementation purposes, we use a "faux" objective function in RDM-0 and RDMFM-0, where we minimize some constant (*e.g.* $\min 0$).

In Figures 3.13 and 3.14 we show the performance profiles for the solving time of instances of T_{rand} and T_{3tc}, respectively. While there is no clear faster model for instances of T_{rand}, for instances of T_{3tc}, RDM-0 seems to be more efficient in finding feasible solutions. The tests also seem to show that solving the COCMST(Λ) problem and the TE-MSTP(Λ) decision problem takes more or less the same time, with a slight advantage for the latter. We also analyze the quality of the solutions obtained, with respect to their worst-case edge utilization. We define Gap_U^D, analogously to Gap_U^* (see above), but considering the solutions obtained for this decision problem. The performance profiles in Figures 3.15 and 3.16 show that the values of Gap_U^D obtained for instances of both test sets, are similar when we solve RDM-0 and RDMFM-0. Regardless, the average quality of these solutions seems to be not as good as the ones obtained when solving the COCMST problem - with the exception for instances of T_{3tc} and when $\epsilon = 0.01$; this can be observed in Table 3.1.

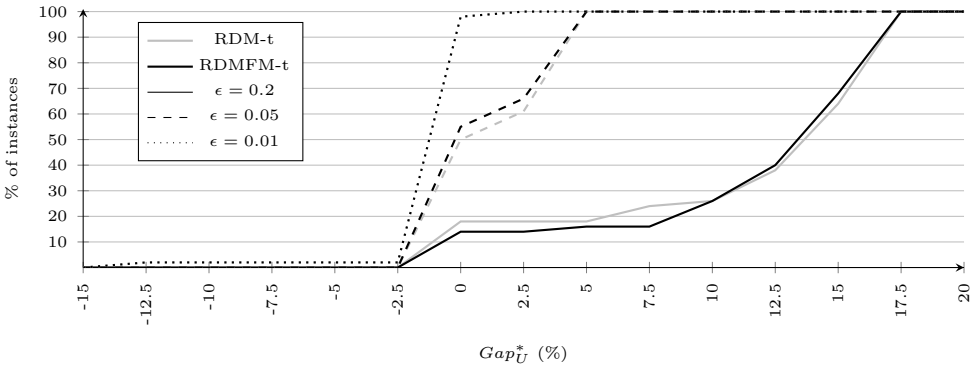


Figure 3.7: COCMST, T_{rand}: performance profile of Gap_U^* (%).

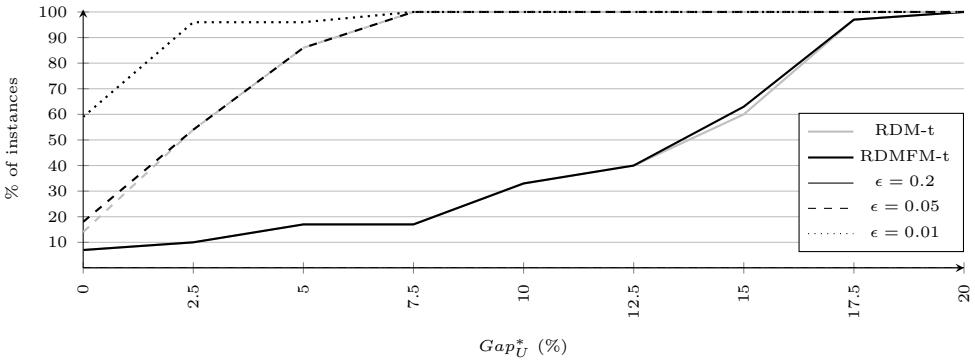


Figure 3.8: COCMST, T_{3tc}: performance profile of Gap_U^* (%).

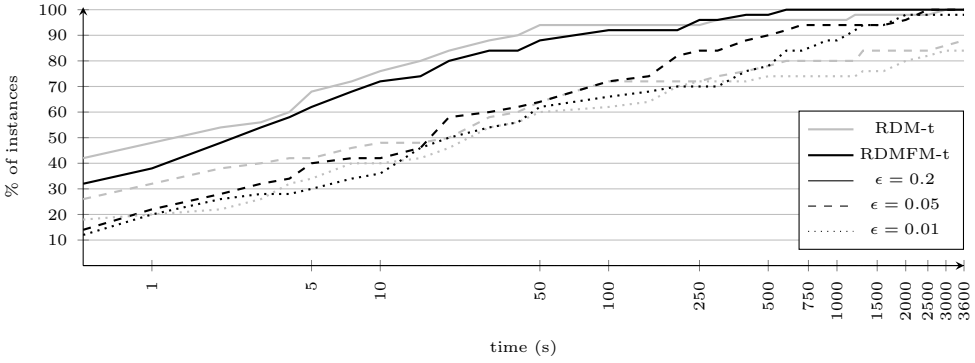


Figure 3.9: COCMST, T_{rand}: performance profile of the time (s) it takes to find a feasible solution.

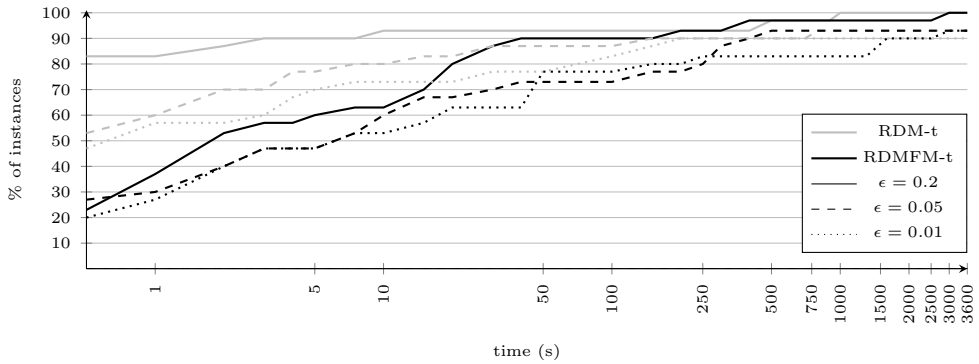


Figure 3.10: COCMST, T_{3tc}: performance profile of the time (s) it takes to find a feasible solution.

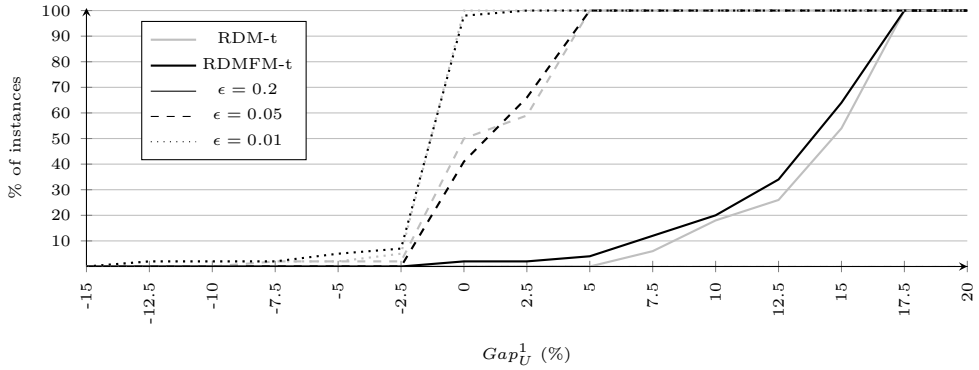


Figure 3.11: COCMST, T_{rand}: performance profile of Gap_U^1 (%).

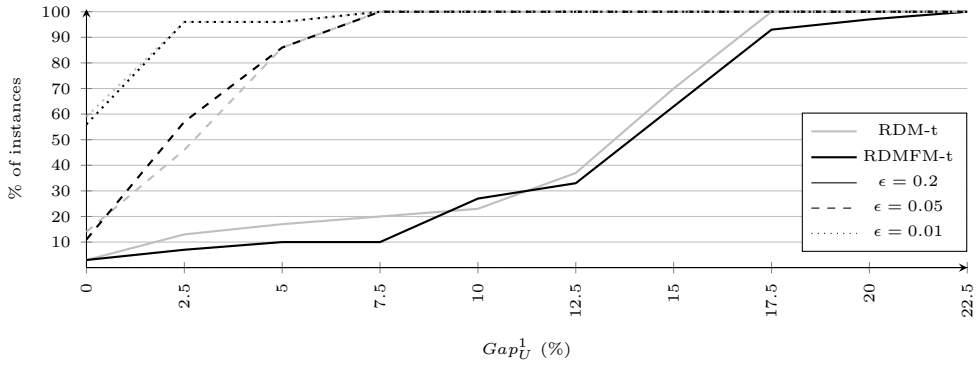


Figure 3.12: COCMST, T_{3tc}: performance profile of Gap_U^1 (%).

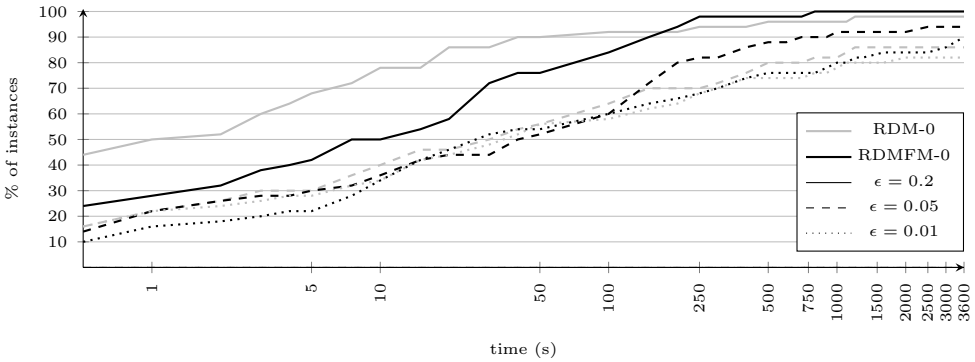


Figure 3.13: TE-MSTP decision, T_{rand}: performance profile of the time (s) it takes to find a feasible solution.

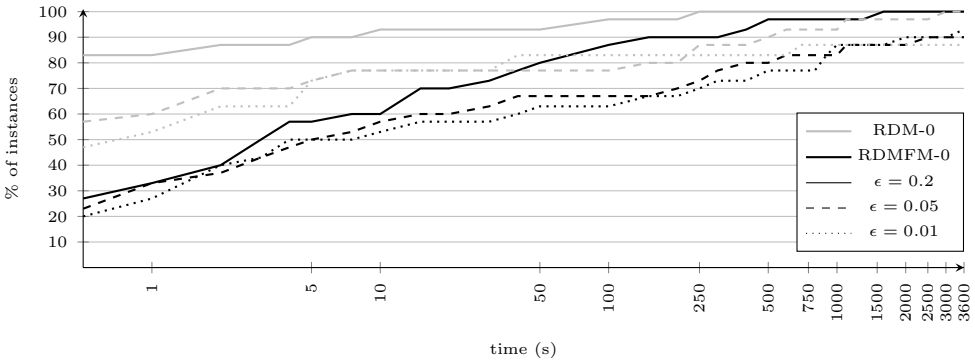


Figure 3.14: TE-MSTP decision, T_{3tc}: performance profile of the time (s) it takes to find a feasible solution.

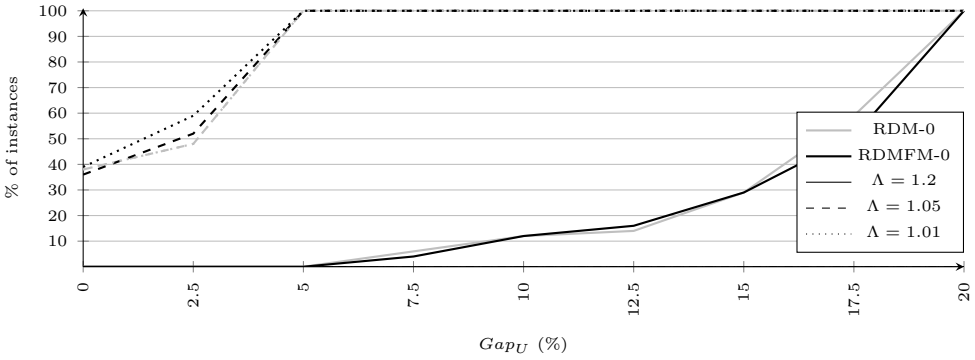


Figure 3.15: TE-MSTP decision, T_{rand}: performance profile of Gap_U^D (%).

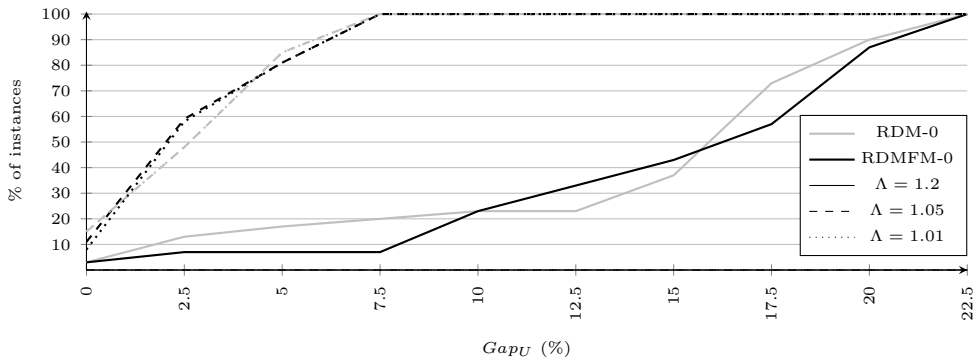


Figure 3.16: TE-MSTP decision, T_{3tc}: performance profile of Gap_U^D (%).

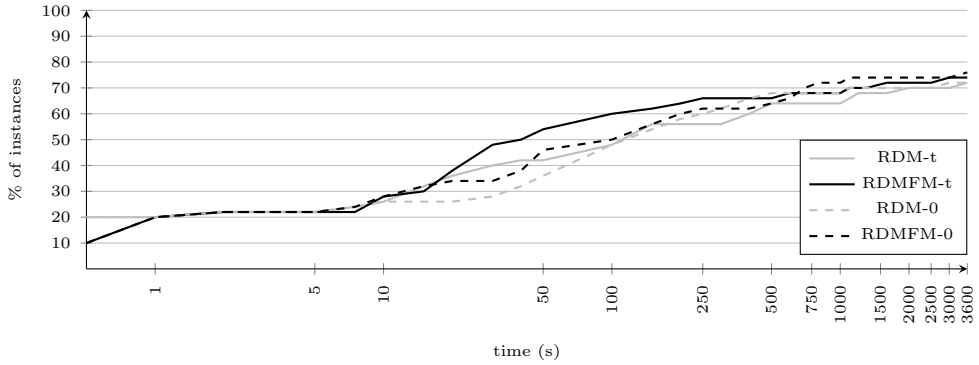


Figure 3.17: COCMST & TE-MSTP decision, T_{rand}: performance profile of the time (s) it takes to prove infeasibility, for $\epsilon = -0.05$.

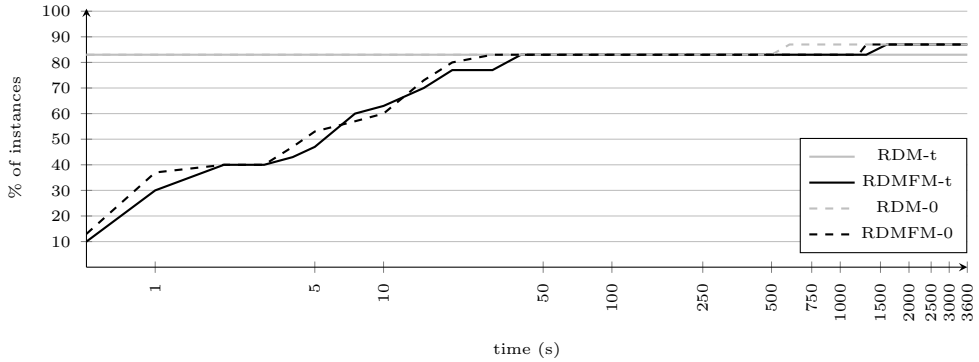


Figure 3.18: COCMST & TE-MSTP decision, T_{3tc}: performance profile of the time (s) it takes to prove infeasibility, for $\epsilon = -0.05$.

3.2.5 Using the COCMST problem to find lower bounds for the TE-MSTP problem

Lastly, we wish to analyze the efficiency of using our models (RDM-t and RDMFM-t for the COCMST problem, and RDM-0 and RDMFM-0 for the TE-MSTP decision problem), to show that an instance cannot have a worst-case edge utilization smaller than a given Λ. This is useful, as if this is the case, Λ is a lower bound to the optimal value of the TE-MSTP problem. In Table C.4, in Appendix C, we present the time it takes for CPLEX implementing our models to prove infeasibility of both COCMST problem and TE-MSTP decision problem, in the case where $\epsilon = -0.05$.

We depict the results for T_{rand} and T_{3tc}, respectively in Figures 3.17 and 3.18. For both test sets, the results when solving the COCMST problem and the TE-MSTP decision problem are similar, with the latter being slightly more efficient. For T_{rand}, the RDMFMs seem to be the better choice of model, whereas for T_{3tc} we observe the converse. Finally, it is interesting to observe that when using the RDMs to attempt to solve instances of T_{3tc}, CPLEX is either able to prove infeasibility immediately, or not do it at all.

3.3 Binary search algorithm

In this section, we propose a binary search algorithm (BSA), that solves the TE-MSTP problem. In this method, we solve a succession of "easier" sub-problems, that allow the BSA to converge to a near-optimal solution of the TE-MSTP problem. These sub-problems must either find feasible solutions, whose edge loads do not exceed the value given by the BSA, or prove that no such solution exists. As it was seen in the previous sections, the COCMST problem and the TE-MSTP decision problem fit the profile of these sub-problems. In Algorithm 3.1, we describe the general framework of the BSA. This framework contains a number of different parameters and/or choices, that will be discussed later on.

In this algorithm, we start by obtaining an upper bound UB (Line 1), and a lower bound LB (Line 2) to U^{max*}, the optimal value for the TE-MSTP problem. Next, while the distance between these bounds is greater than precision ϵ (Line 3), we iteratively try to find feasible solutions (Lines 5-6), such that the edge utilization does not exceed value U^{fix}; value that is set such that it is equidistant from UB and LB (Line 4). If such a solution exists, we update the upper bound UB. Note that the *de facto* worst-case utilization for that solution can be smaller than U^{fix} (Line 8). If the sub-problem is infeasible, we update the lower bound LB to U^{fix} (Line 10). The optimal value U^{max*} will be on the interval of UB and LB (Line 11).

In the next sections, we analyze different possibilities for components and parameters of Algorithm 3.1.

Algorithm 3.1: Binary search algorithm.

```
1  UB ← Get_Upper_Bound
2  LB ← Get_Lower_Bound
3  while UB − LB > ϵ do
4  |  U^fix ← LB + (UB−LB)/2
5  |  m ← Create_Program(U^fix)
6  |  Solve(m)
7  |  if Feasible solution is found then
8  |  |  UB ← Calculate_Max_Utilization(m)
9  |  else
10 |  |  LB ← U^fix
11 U^max* ∈ [LB, UB]
```

3.3.1 Obtaining a first upper bound

The way the TE-MSTP problem is defined, one obvious upper bound for U^{max*} is 1. So, one possibility is to replace Line 1 in Algorithm 3.1 for Lines 1 to 7 in Algorithm 3.2. We begin by finding a feasible solution with worst-case edge utilization 1 (Lines 1-3). If a feasible solution is found, we set as UB the *de facto* worst-case edge utilization of that solution. If no solution is found, the TE-MSTP problem is infeasible.

Algorithm 3.2: Get simple upper bound.

```
1 U^fix ← 1
2 m ← Create_Program(U^fix)
3 Solve(m)
4 if Feasible solution is found then
5 |  UB ← Calculate_Max_Utilization(m)
6 else
7 |  Stop
```

An alternative way to obtain an upper bound is to use an heuristic method that quickly finds a good feasible solution. Namely, the heuristic method proposed by Ho (see Section 1.2) could be implemented, and we expect that this would speed-up the BSA. Nevertheless, note that the heuristic method could also be used to warm-start CPLEX, while solving the models in Section 2.2. So, in view of keeping the performance analysis of the BSA consistent, we do not consider this approach in this thesis.

3.3.2 Obtaining a first lower bound

To obtain a first lower bound, we solve the LP relaxation of the models proposed in Section 2.2 for the TE-MSTP problem. In our experiments, we only tested using RDM

- the model whose LP relaxation is generally the quickest to solve - and RDMFM - the model that tends to produce better lower bounds (see Section 2.4).

3.3.3 Obtaining a feasible solution

In Lines 5 - 6 of Algorithm 3.1, and Lines 2 - 3 of Algorithm 3.2, we attempt to find a feasible solution to the TE-MSTP problem, such that edge utilization is at most a given value, U^{fix}. As aforementioned, one way of achieving such a solution is by solving the COCMST(Λ) problem, for $\Lambda = U^{fix}$. In our computational experiments, we implemented on CPLEX models RDM-t and RDMFM-t (see Section 3.1) for this purpose.

We draw the reader's attention to the flow costs c, in objective functions 3.1a and 3.1a. If we set $c_e = 1, e \in E$, sending flow through every edge will have the same cost. Alternatively, we can manipulate these costs at every iteration of the BSA, such that we increase the odds of the optimization favoring solutions with low worst-case edge utilization. As such, we define $c_e^k, e \in E$, as the cost of sending flow through edge e, at iteration k of the BSA. We set $c_e^0 = 1, e \in E$; for subsequent iterations $k > 0$ and edge $e \in E$: $\bar{c}_e^k := \frac{L_e(\mathcal{S})}{\Gamma \cdot UB - (L_e(\mathcal{S}))}$ and $c_e^k := \Delta \bar{c}_e^k + (1-\Delta) c_e^{k-1}$. $L_e(\mathcal{S})$ is the load on edge $e \in E$, at the last-found feasible solution $\mathcal{S}$, for the TE-MSTP problem. Γ is a parameter that controls the penalization of the heavily loaded edges. Δ is a parameter that stabilizes the update of the objective function. Naturally, if $\Delta = 0$, the costs are kept identical throughout the BSA.

In an attempt to obtain a better first feasible solution, we can also try to force traffic to flow through edges with higher capacity, by setting for every $e \in E$, $c_e^0 = \frac{\Gamma \cdot C^{max} - C_e}{C_e}$; where $C^{max} := \max_{e \in E} C_e$.

As it was scrutinized in Section 3.2.4, any solution of the COCMST problem is feasible for the TE-MSTP problem. This implies that at each iteration of the BSA we do not need to solve the COCMST to optimality. Instead, we can stop the CPLEX's optimization as soon as it finds a feasible solution; unless there is no feasible solution, and in that case the procedure needs to run until CPLEX concludes that the sub-problem is infeasible. As aforementioned, this is similar to solving the TE-MSTP(Λ) decision problem, with $\Lambda = U^{fix}$. For our experiments, we implemented in the BSA both RDM/RDMFM-t for the COCMST problem, and RDM/RDMFM-0 for the TE-MSTP decision problem, described in the previous section.

Lastly, consider the case discussed in the paragraph above, where we stop the optimization when the first feasible solution is found. In the first iterations of the BSA, the corresponding sub-problems are more likely to be feasible. So, it makes sense to focus CPLEX's optimization procedure in the search for feasible solutions. Later, as the bounds start to enclose the optimum value U^{max*}, we start to observe the converse - the sub-problems are more likely to be infeasible. As such, we should ensure that CPLEX's emphasis is on moving the best bound value. We can control this via CPLEX's `MIP emphasis switch`. Accordingly, we define a parameter Ω^{emph}, that controls the moment where we turn `MIP emphasis switch` from finding `Feasible` solu-

tions, to proving `Optimality` (or infeasibility). Assume that $\epsilon = 0.01$ (Line 3 of Algorithm 3.1), and $\Omega^{emph} = 10$. Then, we turn the `MIP emphasis switch` as soon $UB - LB \leq \epsilon \cdot \Omega^{emph} = 0.1$. Moreover, we define $\Omega^{emph} = \infty$ $(-\infty)$ if we keep the `MIP emphasis switch` at the `Feasible` (`Optimality`) option, throughout the entire procedure, and $\Omega^{emph} = 0$ if we keep it at the `Balanced` option.

Likewise, it might be interesting to use different sub-problems, in different moments of the BSA. Namely, searching for feasible solutions, by solving the COCMST problem, might speed up the process of decreasing the upper bound, as solutions with heavily loaded edges are penalized; however, it might be faster for CPLEX to prove infeasibility, when the model has a "faux" objective function like $\min 0$. Analogously, we define parameter Ω^{model} that controls the moment in the BSA where we stop solving the COCMST problem, and start solving the TE-MSTP decision problem instead. We set $\Omega^{mod} = \infty$ $(-\infty)$, if we only solve the COCMST (the TE-MSTP decision) problem throughout the whole procedure.

3.3.4 Local branching

The framework of the BSA lends itself to the use of local branching, proposed by Fischetti and Lodi in [FL03]. This heuristic procedure limits the space of solutions, where the MIP solver (in our case, CPLEX) can perform its search, in an attempt to speed-up the global optimization process. We implement an "adaptive" local branching heuristic at each non-first iteration of the binary search. Let S be a feasible solution for the TE-MSTP problem found at iteration k; UB is its worst-case edge utilization value. Naturally, this solution is not feasible for the sub-problem at iteration $k' > k$ of the BSA, where every edge utilization must be lower than UB. However, instead of starting the optimization at iteration k' from "zero", we can first try to find solutions in the neighborhood of $\mathcal{S}$.

Let $E^t(\mathcal{S})$ the set of edges selected for VLAN $t \in T$, *i.e.* $E^t(\mathcal{S}) = \{e \in E : w_e^t = 1, t \in T\}$. Then, we first try to find a feasible solution whose selected edges in VLAN $t \in T$, differ from $E^t(\mathcal{S})$ in at most Π edges. In this sense, we add the constraint (3.3) to RDM-t, RDMFM-t, RDM-0 and RDMFM-0.

$$\sum_{e \in E^t(\mathcal{S})} w_e^t \geq n - 1 - \Pi, \qquad t \in T \tag{3.3}$$

Formulation 3.3: BSA: local branching constraint.

If no solution is found for Π, we increase it until we either find a feasible solution, or $\Pi > m-(n-1)$, where m is the cardinality of E. For this purpose, we define parameters Π^0 and Π^r. The initial value for Π is given by $\lceil \Pi^0 n \rceil$; Π then increases $\lceil \Pi^r n \rceil$ at every iteration of the local branching procedure. Note that the value of these parameters must be such that $\Pi^0 + \Pi^r \leq 1$.

3.3.5 Parameters configuration

Table 3.2 details all the parameters for Algorithm 3.1, described in the previous sections. The first column identifies the parameter, the second column refers to the section where further explanations about the parameter are provided, and the last one indicates the potential values/options for that parameter.

1^{st} *LB model* refers to the model whose LP relaxation is used to obtain the first lower bound. *MIP model* indicates which model is chosen to obtain feasible designs at the different iterations of the BSA. *MIP stopping criteria* indicates whether we should solve each sub-problem to optimality or stop upon finding the first feasible solution, when the aforementioned choice of sub-problem is the COCMST problem. Parameters Ω^{emph}, Ω^{mod}, Γ, c_e^0 and Δ are explained in Section 3.3.3. The two latter parameters are only relevant if the choice of sub-problem is the COCMST problem. The same applies to the choice of weights of the objective function in the first iteration, c_e^0: we can either have the same weight for every edge, or have different weights, depending on the capacity of the edge. Finally, the role of parameters Π^0 and Π^r is explained in Section 3.3.4.

Parameter	Sec.	Values	Best
1^{st} LB model	3.3.2	{RDM, RDMFM}	RDMFM
MIP model	3.3.3	{RDM(-t/-0), RDMFM(-t/-0)}	RDM(-t/-0)
MIP stopping criteria	3.3.3	{1^{st} feas., Opt. sol.}	1^{st} feas.
Ω^{emph}	3.3.3	$\{-\infty, 0, 2, 5, 10, 50, \infty\}$	10
Ω^{mod}	3.3.3	$\{-\infty, 2, 5, 10, 50, \infty\}$	10
Γ	3.3.3	$\{1.01, 1.1, 2\}$	2
c_e^0	3.3.3	$\{c_e^0 = 1, c_e^0 = \dots \Gamma \dots\}$	$c_e^0 = \dots \Gamma \dots$
Δ	3.3.3	$\{0, 0.2, 0.4, 0.6, 0.8, 1\}$	0.2
Π^0	3.3.4	$\{0.1, 0.3, 0.5, 0.7, 0.9, 1\}$	0.1
Π^r	3.3.4	$\{0.1, 0.3, 0.5, 0.7, 0.9\}$	0.9

Table 3.2: Parameters for Algorithm 3.1.

Naturally, the process of configuring the BSA can be very tedious and long if done manually. As such we used the sequential model-based algorithm configuration (SMAC) tool [HHLB11] to optimize the parameters in Table 3.2, in order to reduce the computational time needed by the BSA to achieve near-optimal solutions, with precision $\epsilon = 0.01$.

We randomly selected 30% of the instances of T_{rand} and T_{3tc} as training instances. We set a time limit of one week, over which SMAC tested 107 different parameter configurations. The best configuration found by the tool is described in the last column of Table 3.2. In it, we choose RDMFM for obtaining the first lower bound. This is as expected, since this model obtains the best LP bounds (see the experiments analyzed in Section 2.4). Nevertheless, in order to obtain feasible designs in each iteration of the BSA the chosen model is RDM. Also, note that we do not solve these MIP to optimality, but rather stop upon finding a first feasible solution. The values for Ω^{emph} and Ω^{mod},

reflect that it is convenient to switch both the CPLEX `MIP emphasis switch` from `Feasible` to `Optimality`, and the problem to be solved from the COCMST problem to the TE-MSTP decision problem, as soon as $UB - LB \leq 0.1$. The values for parameters Γ, c_e^0 and Δ hint that it is best to iteratively adapt the weights in the objective function of the COCMST problem; regardless it is equally important to keep these weights stable throughout the BSA . Finally, the values for the Π parameters imply that local branching is beneficial for the efficiency of the BSA, but only as a two-iteration process: in a first phase, we look for designs very similar to the design found in the previous iteration, and then, if no solution is found during the first phase, we search for designs in the entire solution space.

3.4 Computational experiments for the BSA

In this section, we analyze the efficiency of the BSA in solving instances of the TE-MSTP problem. We tested the BSA using the parameter configuration described in the previous section, and a stopping precision $\epsilon = 0.01$. We compare the performance of the BSA with the one of CPLEX running our most promising model, RDMFM. To make the experiments consistent, we also relax the precision of the stopping criteria of CPLEX to 0.01. The results are reported in Tables D.1 and D.2 of Appendix D.

Figures 3.19 and 3.20 depict the performance profiles for the solving times, for test sets T_{rand} and T_{3tc} respectively. They show that even though the BSA is not particularly faster than CPLEX running RDMDM for instances of T_{rand}, there is some improvement for instances of T_{3tc}. More importantly, we can observe a promising result: for instances that neither the BSA nor CPLEX with RDMFM are able to solve in the time limit, the distance between the corresponding upper and lower bounds tends to be quite narrower for the BSA: 13% versus 41% for T_{rand}, and 1% versus 15% for T_{3tc}. This result hints that the BSA can be more effective than simply have CPLEX running RDMFM, when it comes to optimizing larger and more difficult instances than the ones in T_{rand} and T_{3tc}. In this case, the BSA can be an interesting method to explore, as it seems to be quicker in obtaining reasonable solutions with provable bounds.

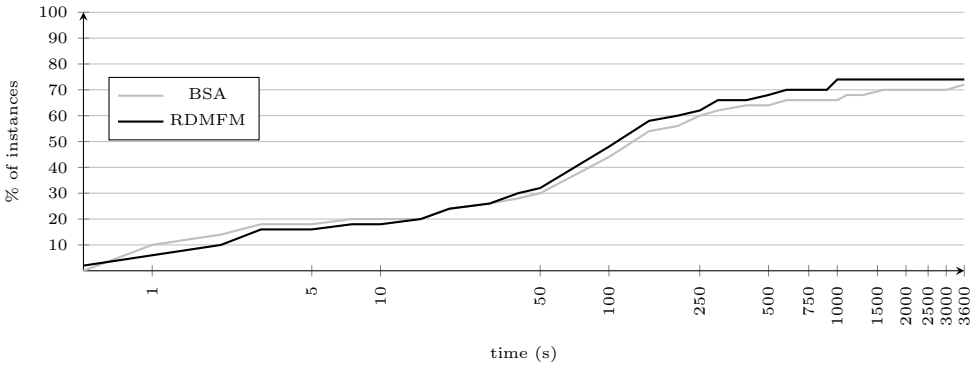


Figure 3.19: TE-MSTP, T_{rand}: performance profile of the MIP solving time (s).

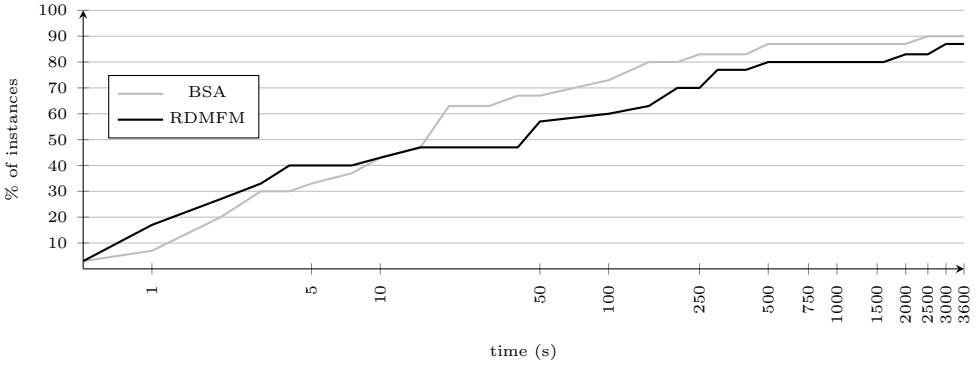


Figure 3.20: TE-MSTP, T_{3tc}: performance profile of the MIP solving time (s).

3.5 Summary and remarks

In this chapter, we studied another network optimization problem for the MSTP. This problem, the COCMST problem is in all identical to the TE-MSTP problem, considered in the previous chapter, with the exception of the objective: in the COCMST problem our goal is to minimize the cost of the total load. In order to solve this problem, we adapted the models that have been proposed for the TE-MSTP problem, resulting in RDM-t, MFM-t and RDMFM-t. We tested the performance of these models in solving the COCMST problem for different values of edge capacity. We observed that RDM-t and RDMFM-t are the formulations that perform better. Moreover, this problem tends to be easier to solve than the TE-MSTP problem - particularly, when the capacity is not too "tight".

The COCMST problem relates to the TE-MSTP problem in the sense that any feasible solution to the former, is also feasible to the latter, with a guaranteed worst-case edge utilization. In this sense, we propose a procedure that obtains near-optimal solutions to the TE-MSTP problem, by iteratively looking for solutions (or proving infeasibility) for the COCMST problem, with edge capacity values given by a BSA. Our tests imply that this method performs well with respect to test set T_{3tc}, finding near-optimal solutions faster than CPLEX implementing RDMFM. Furthermore, for the more "difficult" instances (*i.e.* instances that were not solved by neither method), the BSA tends to achieve a narrower gap between its final upper and lower bound than CPLEX running RDMFM does. This hints that for larger instances, the BSA can be a more efficient method to solve the TE-MSTP problem. In conclusion, the BSA can be a particularly useful method, if the goal of the network provider is not necessarily to obtain the solution with the lowest worst-case link utilization, but one that approaches this solution with a guaranteed distance.

Finally, we remark that the efficiency of the BSA can be further improved, by using heuristics to quickly obtain a good feasible solution and promptly lower the UB. We did not explore this option here, as the same can be applied to CPLEX and RDMFM.

Chapter 4

Piecewise linear unsplittable multicommodity flow problems

In this chapter, we study unsplittable multicommodity flow problems with piecewise linear cost functions. We focus on the case where these functions are convex. However, later, we also discuss the situation when the cost functions are non-convex. We begin by defining the convex unsplittable multicommodity flow (PUMF) problem. Let $G' = (N, A)$ be a directed graph, with a set of nodes N, and a set of arcs A. Consider as well a set of commodities K, where each commodity $k \in K$ has a given origin o_k, a destination d_k, and a demand ρ_k to be routed from o_k to d_k.

Each arc $a \in A$ has an associated cost function $g_a(l_a)$ of the load flowing through the arc l_a. This cost function is continuous, convex and piecewise linear, with the segments being represented by the finite set $S_a = \{1, 2, ..., |S_a|\}$. Each segment $s \in S_a$ has a lower and upper bound on the flow, represented by the breakpoints b_a^{s-1} and b_a^s. If finite, the breakpoint of the last segment of each arc $a \in A$, $b_a^{|S_a|}$, can be interpreted as the capacity of the arc. However, the case where $b_a^{|S_a|} = \infty$ also holds. A segment is also characterized by a slope c_a^s and an intercept f_a^s. Figure 4.1 illustrates this notation. Since in the PUMF problem, the cost functions are convex, these values must be such that $c_a^1 \geq 0$, $c_a^s > c_a^{s-1}$ and $f_a^s \leq 0$, $f_a^s < f_a^{s-1}$. Moreover, as we consider the cost function to be continuous, we assume that $b_a^s c_a^s + f_a^s = b^s c_a^{s+1} + f_a^{s+1}$. We also assume that for every arc $a \in A$, $g_a(0) = 0$, and consequently, $f_a^1 = 0$. The PUMF problem is to find a single path for each commodity, such that the sum of the costs associated to the load of the arcs is minimized.

In the PUMF problem, the routing costs, given by the piecewise linear functions, play a double role: they confine the load within the arcs' capacities, and at the same time, they help to avoid unnecessary detours in the network - since the function is additive, solutions with short paths will be favored. An example of such a cost function was proposed by Fortz and Thorup [FT00, FT04] and is illustrated in Figure 4.2.

In this function, the cost of sending flow is cheap for arcs with low utilization. However, the price quickly rises when the utilization approaches the arc's capacity. Even though it is possible for the utilization to go above 100%, this is so heavily penalized in

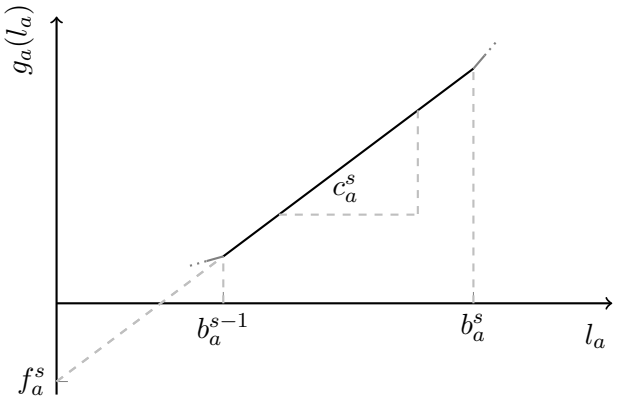


Figure 4.1: Notation for each segment of $g_a(l_a)$.

the cost, that such a solution will likely be avoided.

Objective functions like this have been widely used in problems related with TE in Internet networks. One of the most important and well known of such objectives is the Kleinrock delay function [GK77]:

$$F = \sum_{a \in A} \frac{l_a}{C_a - l_a}, \tag{4.1}$$

where C_a is the capacity of link $a \in A$. This function is illustrated in Figure 4.3. The Kleinrock function helps avoid congestion by penalizing heavily loaded links. Observe that this objective function is convex, and so it can be optimized using convex program-

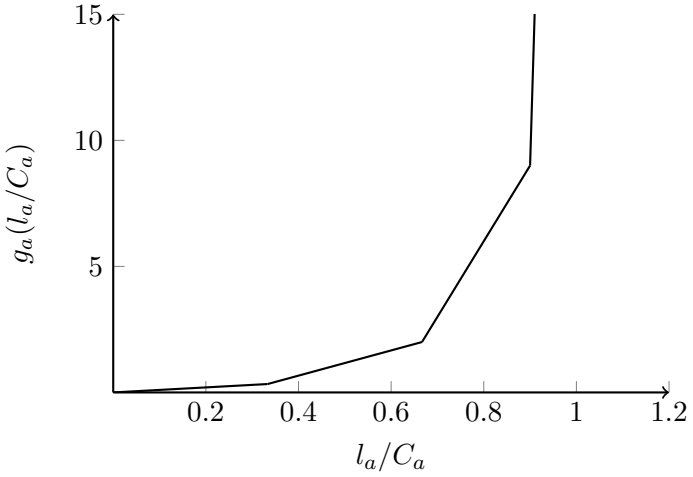


Figure 4.2: Example of a convex piecewise linear cost function.

ming methods [LOP07]. Nevertheless, due to the discrete nature of the PUMF problem, it is convenient to approximate the Kleinrock function with a convex piecewise linear function ([FT00, FT04, PM04]), leading to a MIP problem, that can be solved with the powerful MIP solvers available today (*e.g.* CPLEX). Balon et al. [BSL06] and Gourdin [GK06] discuss various TE objective functions. These authors evaluate how well different objective functions meet TE requirements, and conclude that piecewise linear objectives provide a good trade-off between different measures of quality of service.

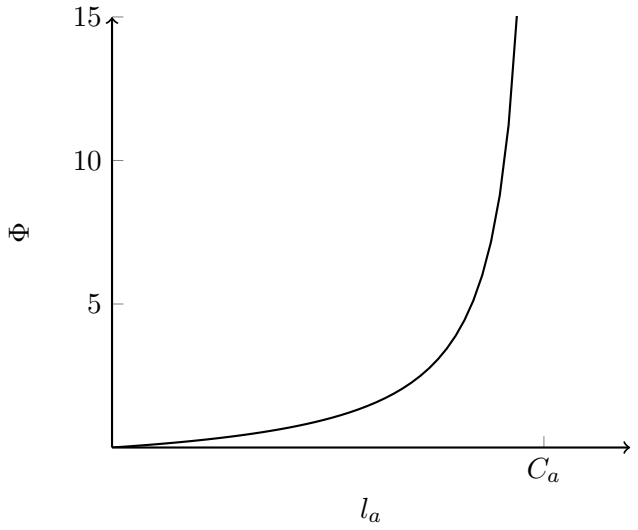


Figure 4.3: Kleinrock function.

Recently, such an objective function was also used by Papadimitriou and Fortz [PF14b, PF14a] in the context of a complex multi-period design and routing problem. Lower bounds resulting from the LP relaxation of the problem are very weak, and part of this weakness is due to the piecewise linear objective function combined with single path routing. Stronger models provided in this thesis could be embedded in the models of [PF14b, PF14a] to improve the lower bounds and make the problems more tractable.

The literature is rich in multicommodity flow problems, but the following two can be identified as being more closely related to the PUMF problem: the origin-destination integer multicommodity flow (ODIMCF) problem, and the non-convex piecewise linear multicommodity network flow (NCPMF) problem. The ODIMCF problem was first introduced by Barnhart *et al.* [BHV00] and has also been referred to as the unsplittable multicommodity flow problem in [AdC03]. This problem shares with the PUMF problem the "unsplitability" of the flows. Nonetheless, it differs in the two following ways: the flow costs are directly proportional to the load on the arcs; and there are explicit capacity constraints on the amount of flow traversing an arc; in the PUMF problem this is somewhat enforced by the piecewise linear cost functions (as explained above).

The NCPMF problem, like the PUMF problem, also imposes costs on the flow via piecewise linear functions. However, unlike the cost functions in the PUMF problem, the piecewise linear functions in the NCPMF problem are non-convex. Later we will consider a version of the PUMF problem where the cost functions are also non-convex. Finally, a second major difference between the two problems is that the flows in the NCPMF are splittable. Croxton *et al.* [CGM03] review three formulations that had been previously used in the literature for generic problems with non-convex piecewise linear costs. These three formulations use integer variables to model the costs, and are shown to be equivalent, with respect to their LP relaxation. In [CGM07], the same authors choose one of these formulations, the multiple choice model [BG89], and strengthen it, so as to solve the NCPMF problem. Due to the common properties of the two problems, these works provide an important basis for our work.

In Section 4.1, we discuss the complexity of the PUMF problem. In Section 4.2, we propose MIP formulations that model the problem. In Section 4.3, we present the results of computational experiments, that help us compare the performance of these formulations in solving instances of the PUMF problem. In Section 4.4 we discuss a B&C algorithm for the PUMF problem. In Section 4.5.2, we propose a new MIP formulation for the model, which is inferred from the interpretation of cuts generated by the Benders' decomposition method of the models proposed in Section 4.5.1. In Section 4.6 we study a variant of the PUMF problem, where the cost functions are non-convex. Finally, in Section 4.7 we discuss conclusions about the research described in this chapter.

4.1 Problem complexity

In this section, we analyse the complexity of the PUMF problem.

Theorem 4. *For $|K| = 1$, the PUMF problem is polynomially solvable.*

Proof. Let us consider an instance of the PUMF problem given by a graph $G = (N, A)$ and a single commodity with origin $o \in N$, destination $d \in N$ and demand ρ. To each arc $a \in A$ we associate a length $\mathcal{W}_a$, that represents the potential cost of having the commodity flow through it, i.e. $\mathcal{W}_a = f_a^{\bar{s}_a} + c_a^{\bar{s}_a}\rho$, with $\bar{s}_a$ defined such that $\bar{s}_a = \{s : b_a^{s-1} \leq \rho \leq b_a^s\}$; or $\mathcal{W}_a = \infty$ if $\rho > b_a^{|S_a|}$. Solving the single-commodity PUMF problem on G is equivalent to identifying the shortest path between o and d in G with respect to lengths $\mathcal{W}_a$.

Let g^* be the optimum value of the single-commodity PUMF problem in G, and p^* the length of the shortest path defined above. We can build a solution to the PUMF problem by sending the flow through the arcs belonging to the shortest path. This solution is, naturally, feasible and with cost p^*. Thus, $g^* \leq p^*$. However, we also know that, given demand ρ, we cannot have the commodity flowing through arc $a \in A$ with a cost inferior to $\mathcal{W}_a$, as previously defined. Since p^* is the shortest path between o and d, we have that $g^* \geq p^*$, and therefore, $g^* = p^*$.

□

Theorem 4 and its proof will be revisited later on, to show Theorem 8.

Theorem 5. *For $|K| > 1$, the PUMF problem is $\mathcal{NP}$-complete.*

Proof. This can be shown by reducing the bin-packing problem to the PUMF problem; a similar proof is found in [Kle96]. To illustrate this, consider an instance of the bin-packing decision problem, with $|A|$ bins with capacity $\bar{b}$, and $|K|$ objects with size $\rho_k \in \mathbb{N}$. Consider as well a directed graph $G = (N, A)$ with only two nodes, o and d, and $|A|$ parallel arcs connecting them. The load cost of every arc $a \in A$ is given by the same piecewise linear function g, that has only two segments, separated by the breakpoint $b^1 = \bar{b}$. The slope on the second segment is very steep, in such a way that $g_a(\bar{b}+1) > \sum_{a' \in A} g_{a'}(\bar{b}), a \in A$. Finally, let K be a set of commodities, each $k \in K$ with a given demand ρ_k, origin in o and destination in d. The problem of determining if it is possible to fit the $|K|$ objects in the $|A|$ bins, is equivalent to determining if all the commodities in K are routable in G, with a cost not bigger than $\sum_{a \in A} g_a(\bar{b})$. As the bin-packing problem is $\mathcal{NP}$-complete [Kar72], so is the PUMF problem. □

4.2 Problem formulation

Consider the notation for the PUMF problem introduced in the beginning of this chapter. In addition, let λ_i^k be such that $\lambda_{o_k}^k = 1$, $\lambda_{d_k}^k = -1$, and $\lambda_i^k = 0$ for every $i \neq \{o_k, d_k\}$. In this section, we propose MIP formulations for the PUMF problem. In Section 4.2.1, we describe two basic models, based on models proposed in the literature for other problems dealing with piecewise linear costs (see the beginning of the chapter for a detailed review). In Section 4.2.2, we focus on the single-commodity case of the PUMF problem, and propose a strengthened formulation, whose LP relaxations always yield integer solutions. Finally, in Section 4.2.3, we extend the strengthened formulation for the multicommodity case.

4.2.1 Basic formulations

We define binary variables x_a^k, with $x_a^k = 1$ if arc $a \in A$ is on the unique path chosen to route commodity $k \in K$, and $x_a^k = 0$ otherwise; and y_a^s, with $y_a^s = 1$, if arc $a \in A$ contains a non-zero flow on segment $s \in S_a$, and $y_a^s = 0$ otherwise. For the sake of simplicity, if arc a contains a non-zero flow on segment $s \in S_a$, we say that arc a is on segment $s \in S_a$. We also define continuous variables l_a^s, that indicate the load going through arc $a \in A$ on segment $s \in S_a$. The PUMF problem can be formulated with the MIP in Formulation 4.2, which we refer to as the Basic Model 1 (BM1)

The typical multicommodity flow balance constraints (4.2b) define the path between the origin and destination node of each commodity. Then, constraint sets (4.2c-4.2e) identify the segment each arc is on. Naturally, only a single segment may be selected per arc (4.2c). The choice of segment is implied by the load flowing through the respective arc. This load is given by constraints (4.2d) and its value is assigned to one of the variables l. To ensure that only the appropriate load variable is non-zero, in (4.2e) we

$$\min \quad \sum_{a\in A}\sum_{s\in S_a}(f_a^s y_a^s + c_a^s l_a^s) \tag{4.2a}$$

$$\text{s.t.} \quad \sum_{a\in\delta^+(i)} x_a^k - \sum_{a\in\delta^-(i)} x_a^k = \lambda_i^k, \qquad i\in N, k\in K \tag{4.2b}$$

$$\sum_{s\in S_a} y_a^s \le 1, \qquad a\in A \tag{4.2c}$$

$$\sum_{s\in S_a} l_a^s = \sum_{k\in K}\rho^k x_a^k, \qquad a\in A \tag{4.2d}$$

$$b_a^{s-1} y_a^s \le l_a^s \le b_a^s y_a^s, \qquad a\in A, s\in S_a \tag{4.2e}$$

$$x_a^k \in \{0,1\}, \qquad a\in A, k\in K \tag{4.2f}$$

$$y_a^s \in \{0,1\}, \qquad a\in A, s\in S_a \tag{4.2g}$$

$$l_a^s \ge 0, \qquad a\in A, s\in S_a \tag{4.2h}$$

Formulation 4.2: PUMF problem: Basic model 1.

either bound l_a^s by the breakpoints of the corresponding segment, if $y_a^s = 1$; or we force it to zero, if $y_a^s = 0$.

Variables x and y are considered as binary (4.2f-4.2g), whilst variables l are considered as continuous (4.2h). Note that if the cost function for every arc is convex, it is not necessary to define explicitly y as binary; instead they can simply be defined as continuous and non-negative. The impact on the effectiveness of the model of defining variables y as binary or continuous is discussed in Section 4.3.

Theorem 6. *If cost functions g_a are convex for all $a \in A$, then the optimal solution of the LP relaxation of BM1 has binary y-variables.*

Proof. Consider a solution of BM1, $\hat{\mathcal{S}}$, with cost $\hat{g}$, and fractional y-variables. Without loss of generality, we assume that this only occurs for one arc - arc a. Then, $\exists s \in S : \hat{y}_a^s \in]0,1[$. We show that there exists a feasible solution for BM1 $\tilde{\mathcal{S}}$, with $\tilde{x}_a^k = \hat{x}_a^k, a \in A, k \in K$, binary y-variables, and cost $\tilde{g} : \tilde{g} \le \hat{g}$.

Let us first assume the case where $\sum_{s\in S_a}\hat{y}_a^s < 1$. As $f_a^1 = 0$, it is possible to increase the value of $\hat{y}_a^1$ without increasing $\hat{g}$. Since $b_a^0 = 0$, this new solution would satisfy (4.2e). As such, let us re-define $\hat{y}_a^1 := 1 - \sum_{s\in S_a\setminus\{1\}}\hat{y}_a^s$ instead. Then, $\sum_{s\in S_a}\hat{y}_a^s = 1$.

Now, let us consider the case where $\sum_{s\in S_a}\hat{y}_a^s = 1$. Then, we show that solution $\tilde{\mathcal{S}}$, with $\tilde{y}_a^{\bar{s}} = 1$ and $\tilde{l}_a^{\bar{s}} = \sum_{s\in S_a}\hat{l}_a^s$ where $\bar{s} = \{s : b_a^{s-1} \le \sum_{s\in S_a}\hat{l}_a^s \le b_a^s\}$, is not more expensive than $\hat{\mathcal{S}}$.

First, let us define $\breve{l}_a^s = \frac{\hat{l}_a^s}{\hat{y}_a^s}$ if $\hat{y}_a^s > 0$ and 0 otherwise. Values $\breve{l}$ are feasible for BM1, as for every $s \in S_a$ such that $\hat{y}_a^s > 0$:

$b_a^{s-1}\hat{y}_a^s \le \hat{l}_a^s \le b_a^s\hat{y}_a^s \Leftrightarrow b_a^{s-1} \le \frac{\hat{l}_a^s}{\hat{y}_a^s} \le b_a^s \Leftrightarrow b_a^{s-1} \le \breve{l}_a^s \le b_a^s$.

Then, we have that:

$\tilde{g} = f_a^{\bar{s}}\tilde{y}_a^{\bar{s}} + c_a^{\bar{s}}\tilde{l}_a^{\bar{s}} = f_a^{\bar{s}}\sum_{s\in S_a}\hat{y}_a^s + c_a^{\bar{s}}\sum_{s\in S_a}\hat{l}_a^s = f_a^{\bar{s}}\sum_{s\in S_a}\hat{y}_a^s + c_a^{\bar{s}}\sum_{s\in S_a}\hat{y}_a^s\breve{l}_a^s$.

Note that this implies that solution $\tilde{\mathcal{S}}$ is a convex combination of feasible solutions with load $\breve{l}_a^s$. As the cost functions are convex, we know that:

$\tilde{g} = f_a^{\bar{s}}\sum_{s\in S_a}\hat{y}_a^s + c_a^{\bar{s}}\sum_{s\in S_a}\hat{y}_a^s\breve{l}_a^s \leq \sum_{s\in S_a}\hat{y}_a^s(f_a^s + c_a^s\breve{l}_a^s) = \sum_{s\in S_a}(f_a^s\hat{y}_a^s + c_a^s\hat{y}_a^s\breve{l}_a^s) = \sum_{s\in S_a}(f_a^s\hat{y}_a^s + c_a^s\hat{l}_a^s) = \hat{g}$. □

BM1 is a multiple choice model, following the terminology in [BG89]; a single segment is chosen per arc, which allows for the direct pricing of the flow, based on the selected segment's intercept and slope. An alternative way of modelling costs, imposed by a piecewise linear functions is suggested in [FT00]. The authors define variables that stand for the routing cost of each arc, to which they impose lower bounds representing the segments of the cost function. Let g_a be the routing cost for arc $a \in A$. We denote the MIP in Formulation 4.3 as the Basic Model 2 (BM2).

$$\min \quad \sum_{a\in A} g_a \tag{4.3a}$$

$$\text{s.t.} \quad \sum_{a\in\delta^+(i)} x_a^k - \sum_{a\in\delta^-(i)} x_a^k = \lambda_i^k, \qquad i \in N, k \in K \tag{4.3b}$$

$$l_a = \sum_{k\in K}\rho^k x_a^k, \qquad a \in A \tag{4.3c}$$

$$g_a \geq f_a^s + c_a^s l_a, \qquad a \in A, s \in S_a \tag{4.3d}$$

$$x_a^k \in \{0,1\}, \qquad a \in A, k \in K \tag{4.3e}$$

$$l_a \geq 0, \qquad a \in A, s \in S_a \tag{4.3f}$$

$$g_a \geq 0, \qquad a \in A \tag{4.3g}$$

Formulation 4.3: PUMF problem: Basic model 2.

Constraints (4.3c) determine the flow traversing each arc, based on the routing defined in (4.3b). In (4.3d), for each segment $s \in S_a$, a lower bound is set for the routing cost g_a of arc $a \in A$. Note that for a given value of l_a, only a single constraint is tight - the one corresponding to the segment arc a is on. Finally, our objective (4.3a) is to minimize the sum of the routing costs for every arc.

Consider the LP relaxations of BM1 and BM2, where we relax the integrality constraints for variables x and y (when applicable). We denote as $g_{B1}(\mathcal{S})$ ($g_{B2}(\mathcal{S})$), the cost of a feasible solution $\mathcal{S}$ of the LP relaxation of BM1 (BM2), and as ${g_{B1}}^*$ (${g_{B2}}^*$) the cost of the optimal solution to these relaxations.

Theorem 7. ${g_{B1}}^* = {g_{B2}}^*$.

Proof. We begin by showing that ${g_{B2}}^* \leq {g_{B1}}^*$. Let $\mathcal{P}_{B1}$ and $\mathcal{P}_{B2}$ be the polyhedra defined by the set of feasible solutions of the LP relaxation of BM1 and BM2, respectively.

Consider the optimal solution of BM1, $\tilde{\mathcal{S}}^* = \{\tilde{x}_a^k, \tilde{l}_a^s, \tilde{y}_a^s\}$. We construct a solution $\hat{\mathcal{S}} = \{\tilde{x}_a^k, \hat{l}_a, \hat{g}_a\}$ by taking:

- $\hat{l}_a := \sum_{s \in S_a} \tilde{l}_a^s, \quad a \in A;$
- $\hat{g}_a := \max_{s \in S_a}(f_a^s + c_a^s \tilde{l}_a^s), \quad a \in A.$

It is to see that $\hat{\mathcal{S}} \in \mathcal{P}_{B2}$. Let us consider the cost of this solution, $g_{B2}(\hat{\mathcal{S}})$. We have that $g_{B2}(\hat{\mathcal{S}}) = \sum_{a \in A} \hat{g}_a = \sum_{a \in A} \max_{s \in S_a}(f_a^s + c_a^s \tilde{l}_a^s)$. Note that $f_a^s \leq 0$, $a \in A$, and that $\tilde{y}_a^s \geq 0$, $a \in A, s \in S_a$. Then, $\sum_{a \in A} \max_{s \in S_a}(f_a^s + c_a^s \tilde{l}_a^s) \leq \sum_{a \in A} \max_{s \in S_a}(f_a^s \tilde{y}_a^s + c_a^s \tilde{l}_a^s)$. As it has been shown in Theorem 6, variables $\tilde{y}$ are binary, even if their integrality is not explicitly enforced. As such, and following constraints (4.2c,4.2e), we know that $f_a^s \tilde{y}_a^s + c_a^s \tilde{l}_a^s$ will only be non-zero for a single $s \in S_a$. Thus, we can conclude that $g_{B2}(\hat{\mathcal{S}}) \leq \sum_{a \in A} \max_{s \in S_a}(f_a^s \tilde{y}_a^s + c_a^s \tilde{l}_a^s) \leq \sum_{a \in A} \sum_{s \in S_a}(f_a^s \tilde{y}_a^s + c_a^s \tilde{l}_a^s) = {g_{B1}}^*$, and consequently ${g_{B2}}^* \leq {g_{B1}}^*$.

For the converse, consider the optimal solution of BM2, $\hat{\mathcal{S}}^* = \{\hat{x}_a^k, \hat{l}_a, \hat{g}_a\}$. Let us define $\tilde{\mathcal{S}} = \{\hat{x}_a^k, \tilde{l}_a^s, \tilde{y}_a^s\}$ by taking:

- $\tilde{l}_a^s := \hat{l}_a$ if $b_a^{s-1} \leq \hat{l}_a \leq b_a^s$; 0 otherwise, $\quad a \in A.$
- $\tilde{y}_a^s := 1$ if $b_a^{s-1} \leq \hat{l}_a \leq b_a^s$; 0 otherwise, $\quad a \in A.$

Naturally, $\tilde{\mathcal{S}}$ is a feasible solution in $\mathcal{P}_{B1}$. The cost of solution $\tilde{\mathcal{S}}$ is $g_{B1}(\tilde{\mathcal{S}}) = \sum_{a \in A} \sum s \in S_a(f_a^s \tilde{y}_a^s + c_a^s \tilde{l}_a^s)$. As discussed above, for each $a \in A$, $f_a^s \tilde{y}_a^s + c_a^s \tilde{l}_a^s$ is non-zero only for a single $s \in S_a$. As such, $g_{B1}(\tilde{\mathcal{S}}) = \sum_{a \in A} \sum_{s \in S_a}(f_a^s \tilde{y}_a^s + c_a^s \tilde{l}_a^s) \leq \sum_{a \in A} \max_{s \in S_a} f_a^s + c_a^s \hat{l}_a \leq \sum_{a \in A} \hat{g}_a = {g_{B2}}^*$. Therefore, we also have that ${g_{B1}}^* \leq {g_{B2}}^*$, and thus, ${g_{B1}}^* = {g_{B2}}^*$. □

4.2.2 Ideal formulation for $|K| = 1$

Let us consider the single-commodity case of the PUMF problem. We simplify the notation previously used, namely for parameters o, d, ρ, λ, and variables x_a. For the sake of completeness, we repeat BM1 and BM2 for this situation (Formulations 4.4 and 4.5).

The LP relaxation of these basic models can provide very weak lower bounds, even for toy instances. As an example to show how weak these bounds are, consider a graph with only two nodes, o and d, connected by three parallel arcs. Assume that the single commodity with origin o, destination d, has demand $\rho = 3$. Finally, assume that the cost function on all three arcs is the same, and is characterized by only two segments, such that the breakpoints are $b^0 = 0$, $b^1 = 1$ and $b^2 = 3$; the slopes are $c^1 = 1$ and $c^2 = 10$; and the intercepts $f^1 = 0$ and $f^2 = -9$. It is easy to see that there are three solutions to the PUMF problem on this graph, all with the same cost of 21: sending the flow through each one of the three available arcs. However, in the LP relaxation of basic models BM1 and BM2, the flow can be equally split among the three arcs, each on the

$$\min \quad \sum_{a\in A}\sum_{s\in S_a}(f_a^s y_a^s + c_a^s l_a^s) \tag{4.2a}$$

$$\text{s.t.} \quad \sum_{a\in\delta^+(i)} x_a - \sum_{a\in\delta^-(i)} x_a = \lambda_i, \qquad i\in N \tag{4.4a}$$

$$\sum_{s\in S_a} y_a^s \le 1, \qquad a\in A \tag{4.2c}$$

$$\sum_{s\in S_a} l_a^s = \rho x_a, \qquad a\in A \tag{4.4b}$$

$$b_a^{s-1} y_a^s \le l_a^s \le b_a^s y_a^s, \qquad a\in A, s\in S_a \tag{4.2e}$$

$$x_a \in \{0,1\}, \qquad a\in A \tag{4.4c}$$

$$y_a^s \in \{0,1\}, \qquad a\in A, s\in S_a \tag{4.2g}$$

$$l_a^s \ge 0, \qquad a\in A, s\in S_a \tag{4.2h}$$

Formulation 4.4: PUMF problem: Basic model 1 for $|K| = 1$.

$$\min \quad \sum_{a\in A} g_a \tag{4.3a}$$

$$\text{s.t.} \quad \sum_{a\in\delta^+(i)} x_a - \sum_{a\in\delta^-(i)} x_a = \lambda_i, \qquad i\in N \tag{4.5a}$$

$$l_a = \rho x_a, \qquad a\in A \tag{4.5b}$$

$$g_a \ge f_a^s + c_a^s l_a, \qquad a\in A, s\in S_a \tag{4.3d}$$

$$x_a \in \{0,1\}, \qquad a\in A \tag{4.5c}$$

$$l_a \ge 0, \qquad a\in A, s\in S_a \tag{4.3f}$$

$$g_a \ge 0, \qquad a\in A \tag{4.3g}$$

Formulation 4.5: PUMF problem: Basic model 2 for $|K| = 1$.

first segment. This results in a LP relaxation optimum value of only 3. By manipulating the structure of the cost functions, this gap can be virtually any value.

Recall Theorem 4, where it is stated that the PUMF problem is polynomially solvable, when considering only one commodity. However, the LP relaxation of the basic models is not integer in this case, as shown in the example above. As such, we want to develop a MIP model, whose LP relaxation always gives the optimal solution for the PUMF problem with $|K| = 1$. To this end, we use variable disaggregation, a common technique to strengthen the LP relaxation of MIPs. Thus, consider the binary variables x_a^s, with $x_a^s = 1$ if arc $a \in A$ is on segment $s \in S_a$ and on the unique path chosen to route the commodity, and $x_a^s = 0$ otherwise. We denote as the Disaggregated Model (DM) to

Formulation 4.6.

$$\min \quad \sum_{a\in A}\sum_{s\in S_a}(f_a^s y_a^s + c_a^s \rho x_a^s) \tag{4.6a}$$

$$\text{s.t.} \quad \sum_{a\in\delta^+(i)}\sum_{s\in S_a} x_a^s - \sum_{a\in\delta^-(i)}\sum_{s\in S_a} x_a^s = \lambda_i, \qquad i \in N \tag{4.6b}$$

$$\sum_{s\in S_a} y_a^s \leq 1, \qquad a \in A \tag{4.6c}$$

$$b_a^{s-1} y_a^s \leq \rho x_a^s \leq b_a^s y_a^s, \qquad a \in A, s \in S_a \tag{4.6d}$$

$$x_a^s \in \{0,1\}, \qquad a \in A, s \in S_a \tag{4.6e}$$

$$y_a^s \in \{0,1\}, \qquad a \in A, s \in S_a \tag{4.6f}$$

Formulation 4.6: PUMF problem: Disaggregated Model for $|K| = 1$.

In this new model the l-variables are no longer necessary as for every $a \in A, s \in S_a$, $l_a^s = \rho x_a^s$, leading to the new objective function (4.6a) and variable bound constraints (4.6d). Constraints (4.6b) define, for each commodity, a path between the origin and destination, now with the disaggregated x-variables. As for BM1, if the cost functions are convex, we can instead define the y-variables as continuous and non-negative.

It is easy to see that DM, as it has been defined so far, is equivalent to the basic models. However, we can use the disaggregated variables to create new inequalities that considerably strengthen the LP relaxation.

First, note that an arc being traversed by a given commodity cannot be on a segment whose upper breakpoint is smaller than the demand flow. Therefore, we can fix the x-variables as follows:

$$x_a^s = 0, \qquad a \in A, s \in S_a : b_a^s < \rho \tag{4.7}$$

Formulation 4.7: PUMF problem: Valid inequalities I.

Furthermore, when combined with inequalities (4.6d), this variable fixing has a strong impact on the values of the y-variables.

A well-known class of valid inequalities, common with this variable disaggregation, are the following:

$$x_a^s \leq y_a^s, \qquad a \in A, s \in S_a \tag{4.8}$$

Formulation 4.8: PUMF problem: Valid inequalities II.

These valid inequalities are an obvious choice in cases where the intercepts f_a^s are non-negative (*e.g.* [CGM07]), as they lift the y-variables. This is not the case for PUMF. A related class of valid inequalities, but now making use of the fact that the intercepts f_a^s are negative, is obtained by tightening coefficients in the first inequality in (4.4b):

$$b_a^{s-1} y_a^s \leq \min(\rho, b_a^{s-1}) x_a^s, \qquad a \in A, s \in S_a \tag{4.9}$$

Formulation 4.9: PUMF problem: Valid inequalities III.

We refer to this strengthening of DM as the Strong Model (SM). We conclude this section by showing that the SM is an ideal formulation for the single-commodity case of the PUMF problem.

Theorem 8. *When $|K| = 1$, the optimal solution of the linear relaxation of SM is integer and solves PUMF.*

Proof. First, let us recall what was shown in Theorem 4: solving the PUMF problem for $|K| = 1$ in $G = (N, A)$ is equivalent to solving a shortest path problem between o and d where the length associated with each arc in $a \in A$ is $\mathcal{W}_a = f_a^{\bar{s}_a} + c_a^{\bar{s}_a}\rho$, with $\bar{s}_a$ such that $\bar{s}_a = \{s : b_a^{s-1} \leq \rho \leq b_a^s\}$. Let p_i be the length the shortest path from o to i. We can dynamically compute the shortest distance between the origin and every node by defining $p_o := 0$, and $p_i := \min\{p_j + \mathcal{W}_a : a = (i, j) \in \delta^+(i)\}$. Let us consider a primal solution of the single-commodity PUMF problem with cost $g^* = p_d$.

Next consider the dual (SDM) of the linear relaxation of SM, presented in Formulation 4.10. We show that there is a solution of SDM that also has cost g^*.

Dual variables α, β, γ, ζ and η correspond, respectively to constraints (4.6b), (4.6c), to the left-most constraints in (4.6d), to constraints (4.9), and to the relaxation of (4.6e), $x_a^s \leq 1$. In turn, dual constraints (4.10b) are linked to the x_a^s-variables when $s = \bar{s}_a$, constraints (4.10c) to the x_a^s-variables when $s > \bar{s}_a$, and (4.10d) to the y-variables. Let us fix the dual variables as follows:

- $\beta_a = 0$, $a \in A$;
- $\zeta_a^s = 0$, $a \in A, s \in S_a$;
- $\eta_a^s = 0$, $a \in A, s \in S_a$;
- $\gamma_a^s = -\frac{f_a^s}{b_a^{s-1}}$, if $s \leq \bar{s}_a$; $-\frac{f_a^s}{\rho}$ if $s > \bar{s}_a$, $a \in A$.

Constraints (4.10d) are satisfied, both when $s \leq \bar{s}_a$ (obvious), and when $s > \bar{s}_a$, as for those cases $\frac{b_a^{s-1}}{\rho} \geq 1$. The remainder of our dual formulation with fixed variables (SDM-f) is as seen in Formulation 4.11.

Note that $f_a^{\bar{s}_a} + c_a^{\bar{s}_a}\rho \leq f_a^s + c_a^s\rho$, $a \in A, s \in S_a : s > \bar{s}_a$. SDM-f is thus the dual of the LP formulation that describes the shortest path problem, as detailed above. It is well-known [Wol98] that these dual and primal formulations have dual gap zero. Therefore, there is a solution in SDM with cost g^*.

$$\max \quad \alpha_o - \alpha_d - \sum_{a \in A} \beta_a - \sum_{a \in A} \sum_{s \in S_a : b_a^s \geq d} \eta_a^s \tag{4.10a}$$

$$\alpha_i - \alpha_j + b_a^{\bar{s}_a - 1} \gamma_a^{\bar{s}_a} - \rho \zeta_a^{\bar{s}_a} - \eta_a^{\bar{s}_a} \leq c_a^{\bar{s}_a} \rho, \qquad a = (i,j) \in A \tag{4.10b}$$

$$\alpha_i - \alpha_j + \rho \gamma_a^s - \rho \zeta_a^s - \eta_a^s \leq c_a^s \rho, \qquad a = (i,j) \in A, s \in S_a : s > \bar{s}_a \tag{4.10c}$$

$$-\beta_a - b_a^{s-1} \gamma_a^s + b_a^s \zeta_a^s \leq f_a^s, \qquad a \in A, s \in S_a \tag{4.10d}$$

$$\beta_a \geq 0, \qquad a \in A \tag{4.10e}$$

$$\gamma_a^s, \zeta_a^s, \eta_a^s \geq 0, \qquad a \in A, s \in S_a \tag{4.10f}$$

Formulation 4.10: PUMF problem: SDM.

$$\max \quad \alpha_o - \alpha_d \tag{4.11a}$$

$$\alpha_i - \alpha_j \leq f_a^s + c_a^s \rho, \qquad a = (i,j) \in A, s \in S_a : s \geq \bar{s}_a \tag{4.11b}$$

Formulation 4.11: PUMF problem: SDM-f.

□

4.2.3 Strong formulation for $|K| \geq 1$

Consider the binary variables x_a^{ks}, with $x_a^{ks} = 1$ if arc $a \in A$ is on segment $s \in S_a$ and on the unique path chosen to route commodity $k \in K$, and $x_a^{ks} = 0$ otherwise. Note that $x_a^{1s} = x_a^s$, defined in the previous section. We redefine the SM for the multicommodity case in Formulation 4.13.

Note that due to (4.12c), the extension of valid inequalities (4.8), (4.13) can still be useful in cutting-off LP solutions; those that for a given arc a have $x_a^{ks} + x_a^{k's'} > 1$, $k' \neq k$, $s' \neq s$. Nevertheless, we do not include (4.13) in SM, as empirical results show that they seldom improve the bounds of the LP relaxation, and often cause out-of-memory issues for large instances.

$$\min \quad \sum_{a\in A}\sum_{s\in S_a}\left(f_a^s y_a^s + c_a^s \sum_{k\in K}\rho^k x_a^{ks}\right) \tag{4.12a}$$

$$\text{s.t.} \quad \sum_{a\in\delta^+(i)}\sum_{s\in S_a} x_a^{ks} - \sum_{a\in\delta^-(i)}\sum_{s\in S_a} x_a^{ks} = \lambda_i^k, \qquad i\in N, k\in K \tag{4.12b}$$

$$\sum_{s\in S_a} y_a^s \le 1, \qquad a\in A \tag{4.12c}$$

$$b_a^{s-1} y_a^s \le \sum_{k\in K}\min(\rho^k, b_a^{s-1}) x_a^{ks}, \qquad a\in A, s\in S_a \tag{4.12d}$$

$$b_a^s y_a^s \ge \sum_{k\in K}\rho^k x_a^{ks}, \qquad a\in A, s\in S_a \tag{4.12e}$$

$$x_a^{ks}\in\{0,1\}, \qquad a\in A, k\in K, s\in S_a : b_a^s \ge \rho^k \tag{4.12f}$$

$$y_a^s\in\{0,1\}, \qquad a\in A, s\in S_a \tag{4.12g}$$

Formulation 4.12: PUMF problem: SM.

$$x_a^{ks} \le y_a^s, \qquad k\in K, a\in A, s\in S_a \tag{4.13}$$

Formulation 4.13: PUMF problem: Extension of (4.8) for the multiple commodities.

4.3 Computational experiments

In this section, we analyze the results of computational experiments that were conducted in order to compare the performance of CPLEX using BM1, BM2 and SM for solving instances of the PUMF problem. The results are described in Appendix E. All the tests were performed on a Intel Core i7 CPU 960 @ 3.20GHz (x8) with 12GB of memory with 64 bits, and running Ubuntu 14.04.2 LTS (GNU/Linux $3.2.0-26-$generic x86_64). The tests were done using the MIP solver ILOG CPLEX 12.6, implemented in Java programming language. We only allow CPLEX to use one thread of the machine's processor. In Section 4.3.1 we describe the test sets that were used for these experiments. In Section 4.3.2 and 4.3.3 we analyze the results for the two different test sets.

4.3.1 Test sets for the PUMF problem

These experiments were done using two test sets, T_1 and T_2, that follow the motivation of the problem.

In test set T_1, we created 55 instances and grouped them into 11 classes, according to number of nodes, arcs and commodities. Table 4.1 describes each class of instances. In each of these instances, the distribution of the arcs on the graph is random. Each arc was assigned a capacity of 50, 75 or 100. The traffic demand between the origin and

Class ID	$\|V\|$	$\|A\|$	$\|K\|$
1	40	936	70
2	40	1092	70
3	40	1092	100
4	60	2478	200
5	60	2832	150
6	60	2832	200
7	80	316	250
8	80	1896	200
9	80	1896	250
10	80	3160	200
11	80	1896	350

Table 4.1: PUMF problem: description of each class of instances.

destination node of each commodity was calculated using the same formula as for test sets T_{rand} and T_{3tc}, used in the computational experiments described in Section 2.4.1:

$$\rho^k = \alpha O_{o_k} D_{d_k} R_{(o_k,d_k)} e^{\frac{-L_2(o_k,d_k)}{2\Delta}} \tag{4.14}$$

The parameter α was set to 0.6, for all the experiments; we found that this value lead to instances where the majority of the arcs used in the optimal solution were not overloaded. The Euclidian distance (L_2) was substituted by the length of the shortest path between each pair of nodes, with respect to the number of links. Δ is the largest distance in the network. The final values were rounded to the nearest integer.

The routing cost on every arc is given by the function shown in Figure 4.2, and described in [FT00]. This function has 6 segments, separated by the following breakpoints: $\{0, \frac{1}{3}C_a, \frac{2}{3}C_a, \frac{9}{10}C_a, C_a, \frac{11}{10}C_a, \infty\}$, where C_a is the capacity of the respective arc. The slopes of each of these 6 segments are respectively 1, 3, 10, 70, 500 and 5000. Note that the intercepts of each segment can be easily calculated in the sense that $f_a^s = b_a^s(c_a^{s-1} - c_a^s) + f_a^{s-1}, a \in A, s \in S_a$.

In test set T_2, we adapted instances of SNDlib [OPTW07] to better fit the PUMF problem. Table 4.2 details the characteristics of these instances. For each of these SNDlib instances, we created four adapted instances for the PUMF problem. For the first instance, we chose as the flow capacity on each arc, either the pre-installed capacity, or, in the cases where the latter was null, the capacity of the first module. So, for example, the capacity assigned to arc (1,6) in instance `atlanta` was 11000, whereas the capacity assigned to arc (1,2) in instance `newyork` was 1000. For the three other adapted instances, we increased or decreased the aforementioned capacities, maintaining the original proportions between the arcs, such that the average arc's utilization was slightly above the breakpoints in the cost function in Figure 4.2: 38%, 71% and 105%. There are 24 instances in total in T_2.

Instance ID	$\|V\|$	$\|A\|$	$\|K\|$
atlanta–D-B-M-N-C-A-N-N	15	44	210
france–D-B-L-N-C-A-N-N	25	90	300
newyork–D-B-E-N-C-A-N-N	16	98	240
pdh–D-B-E-N-C-A-N-N	11	68	24
sun–D-D-M-N-C-A-N-N	27	102	67
ta1–D-B-E-N-C-A-N-N	24	110	396

Table 4.2: PUMF problem: description of each class of instances in T_2.

4.3.2 Results for test set T_1

In this section, we analyze the results of the computational experiments done over instances of test set T_1; these results are detailed in Tables E.1 and E.2, in Appendix E. Figure 4.4 illustrates the performance profile for the Gap_{LP} of the three modes, for instances of T_1. We can observe what had been stated in Theorem 7: the basic models produce the same lower bound. More importantly, we can see how strong the SM really is. The high percentage (93%) of instances that have $Gap_{LP} = 0$ with the SM, clearly contrasts with the results for the basic models.

Next, we analyze the performance of the different models, when used by CPLEX to solve the MIPs of T_1 instances. In preliminary experiments we solved the LPs of the instances of T_1, using different optimizers methods offered by CPLEX. We verified that using the Barrier method for solving the LPs of the basic models, allows for a large speed-up, when compared with the optimizer provided by CPLEX's automatic selection. As for SM, CPLEX's default selection seems to provide the best results. We also tested the impact in the solving time of the MIPs, of defining variables y in BM1 and SM, as binary or continuous. The results of these experiments are depicted in Figure 4.5. We can observe that defining variables y as continuous is highly advantageous, for both BM1 and SM, allowing CPLEX to solve more instances in the time limit of 1 hour. Furthermore, defining y as binary, in the case of SM, led to memory issues. To have a better understanding of this behaviour, further computational experiments involving test set T_1 were carried out on CPLEX using the configurations detailed in Table 4.3.

Model	LP optimizer	y-variables
SM	Default	Continuous
BM1	Barrier	Continuous
BM2	Barrier	

Table 4.3: Configuration implemented for instances of T_1.

We are now in a better condition for comparing the MIP solving time, for the three different formulations. Figure 4.6 illustrates these results. They reveal that SM clearly out-performs BM1 and BM2 when solved with CPLEX. Even though, by using BM1, we are able to solve 95% of the instances in the time limit, by using SM we solved every

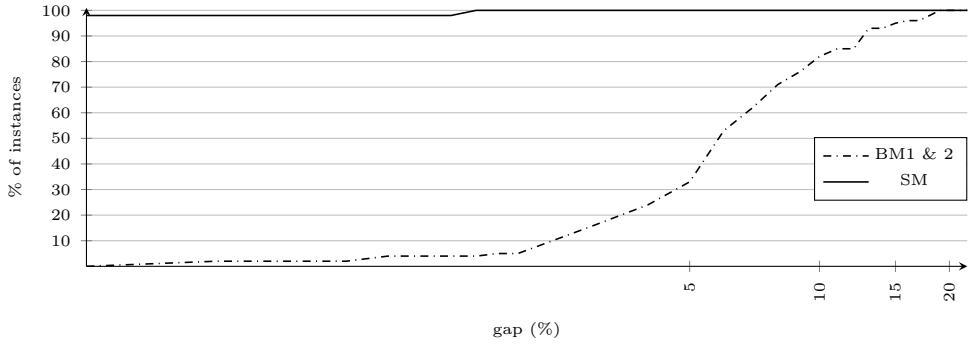


Figure 4.4: PUMF, T_1: performance profile of Gap_{LP} (%).

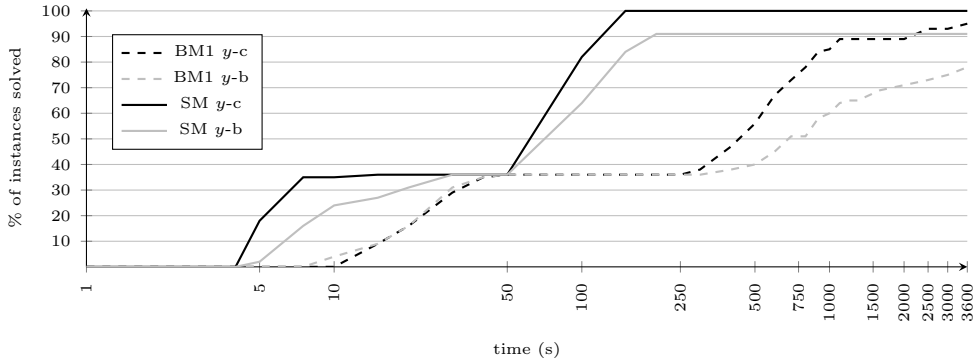


Figure 4.5: PUMF, T_1: performance profile of MIP solving times (s) with y binary (y-b) or continuous (y-c).

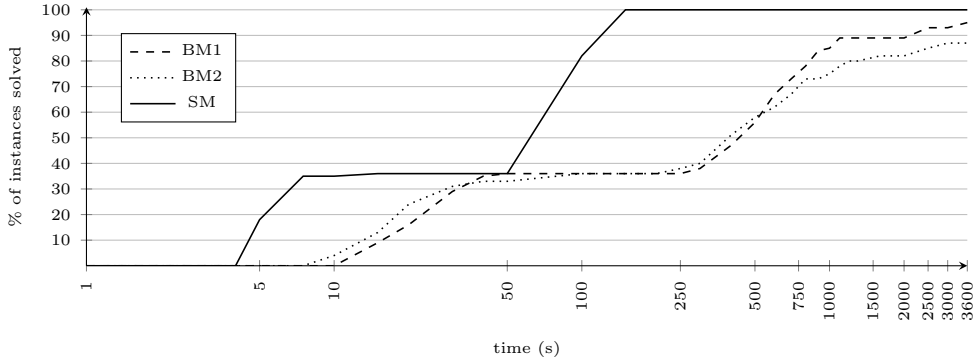


Figure 4.6: PUMF, T_1: performance profile of MIP solving times (s).

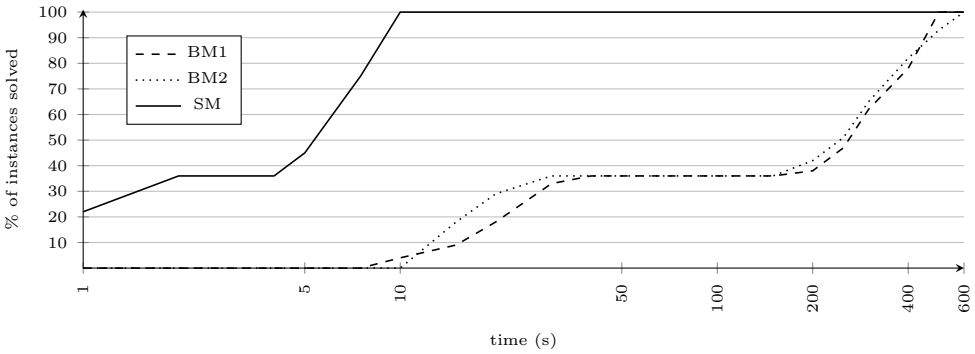


Figure 4.7: PUMF, T_1: performance profile of LP solving times (s).

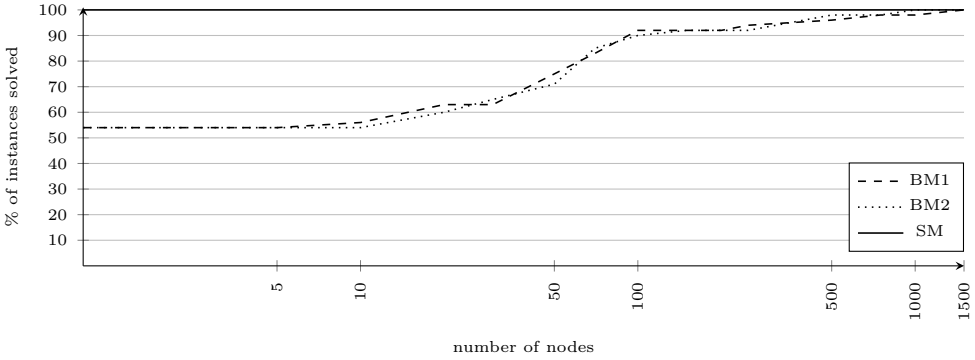


Figure 4.8: PUMF, T_1: performance profile of the B&B tree nodes.

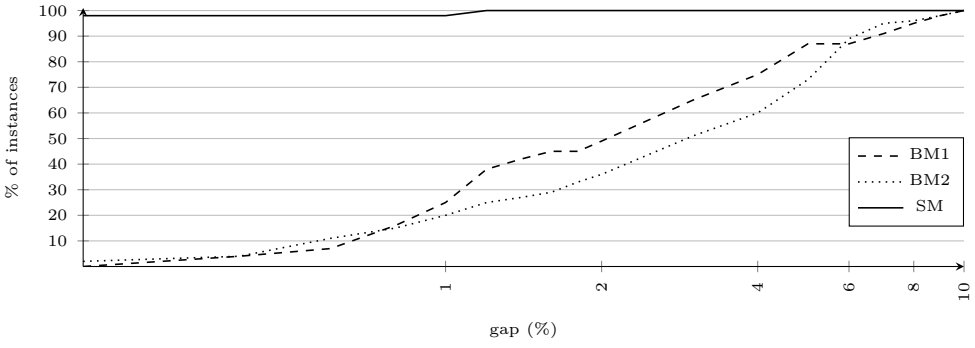


Figure 4.9: PUMF, T_1: performance profile of the Gap_0 (%).

instance, in only under 150 seconds. Finally, we can also observe that BM2 is slower than the other models, only solving 87% of the instances within the time limit.

It is easy to understand that the efficiency of SM is closely linked to its stronger lower bounds. Another argument is the fast solving time of the LPs, observable in Figure 4.7. The strength of SM also accounts for the fact that all instances are solved in the root node of the B&B tree (see Figure 4.8).

For the basic models, it is interesting to note that despite having the same LP bounds (see Theorem 7 and Figure 4.4), comparable LP solving times (see Figure 4.7) and very similarly sized B&B trees (see Figure 4.8), the solving time of the MIPs are distinct. This might be explained by better lower bounds at the end of the root node, as a result of CPLEX's automatically-generated cuts. The latter can be observed in the performance profile of the Gap_0, depicted in Figure 4.9.

4.3.3 Results for test set T_2

In this subsection, we analyze the results of the computational experiments done on instances of test set T_2, described in Table E.3 of Appendix E. Figure 4.10 shows the performance profile for the Gap_{LP}. Once again, SM is able to produce much better lower bounds than both basic models. However, the gaps are slightly higher than they were for instances of T_1. Only 54% of the instances have Gap_{LP} 0, and the values go as high as 3%.

Next, we analyze the performance of CPLEX in solving T_2 instances' MIPs, under our three models. Once again, we analyzed which LP optimizer was better suited for each model. Contrary to what was the case for instances of T_1, for solving the LPs of T_2 instances, CPLEX's default LP optimizer is the fastest for all models. We also analyzed the impact of defining y-variables as continuous or binary. The results can be observed in Figure 4.11. They are quite different than for instances of T_1: for instances of T_2, it seems to be more convenient to define variables y as binary. This is especially observable for SM. We believe that this disparity is a result of the different structures of the networks, in test sets T_1 and T_2, and how they impact the performance of CPLEX's pre-processing. For most instances of T_1, CPLEX is able to reduce the MIP further, when the y-variables are defined as continuous; whereas for instances of T_2, the tendency seems to be the opposite. On further computational experiments, we implement the models on CPLEX following the configurations in Table 4.4.

Model	LP optimizer	y-variables
SM	Default	Binary
BM1	Default	Binary
BM2	Default	

Table 4.4: Configuration implemented for instances of T_2

We can now compare the solving time of the MIPs of the three formulations. The performance profiles illustrated in Figure 4.12 show that for solving instances of T_2, BM2

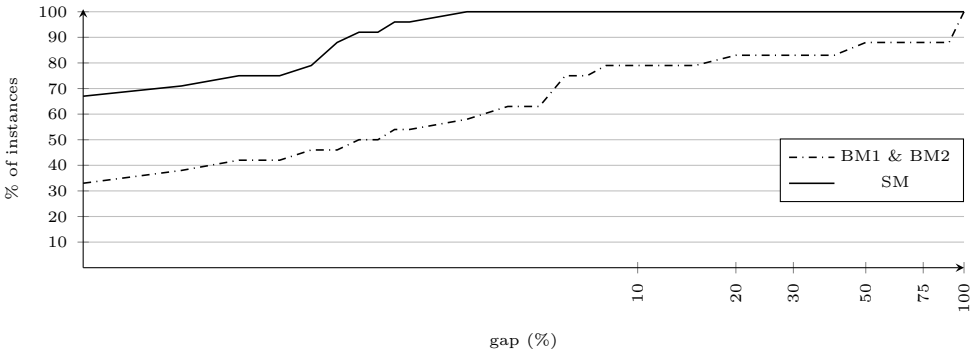


Figure 4.10: PUMF, T_2: performance profile of Gap_{LP} (%).

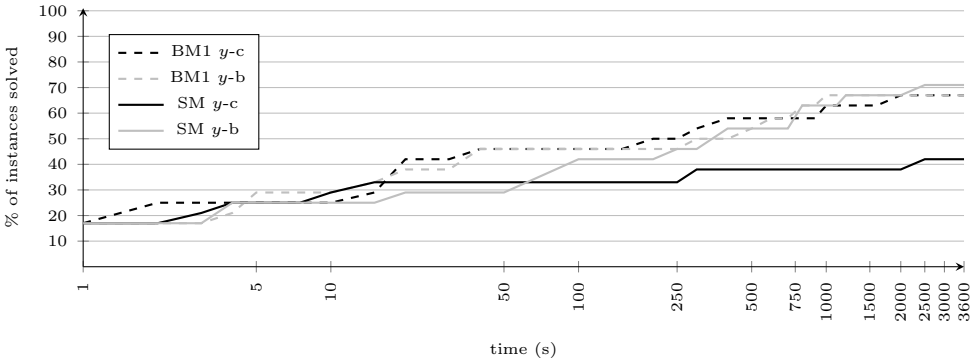


Figure 4.11: PUMF, T_2: performance profile of MIP solving times (s) with y binary (y-b) or continuous (y-c).

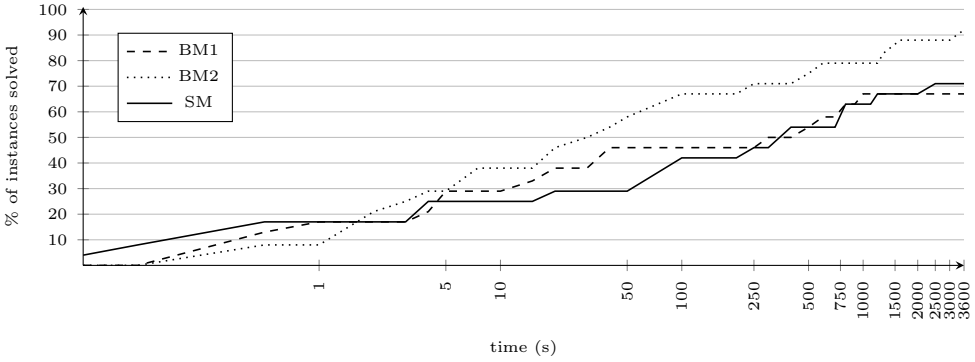


Figure 4.12: PUMF, T_2: performance profile of MIP solving times (s).

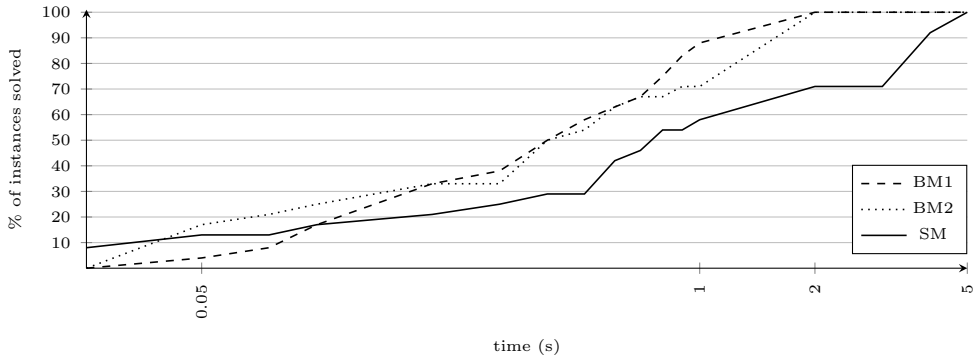


Figure 4.13: PUMF, T_2: performance profile of LP solving times (s).

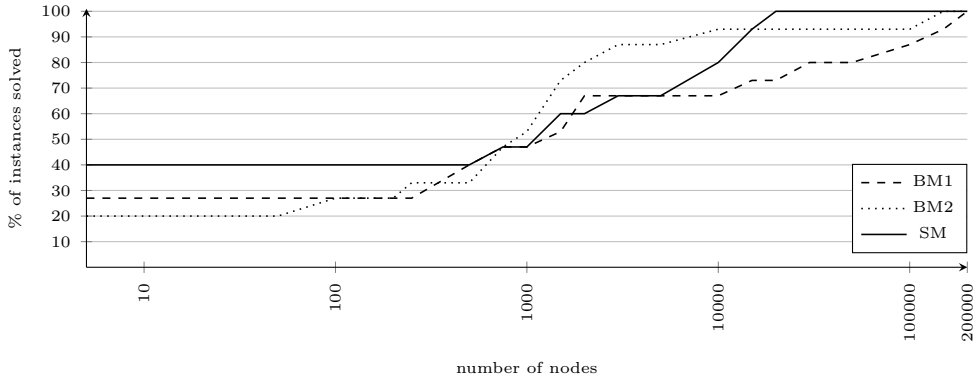


Figure 4.14: PUMF, T_2: performance profile of the B&B tree nodes.

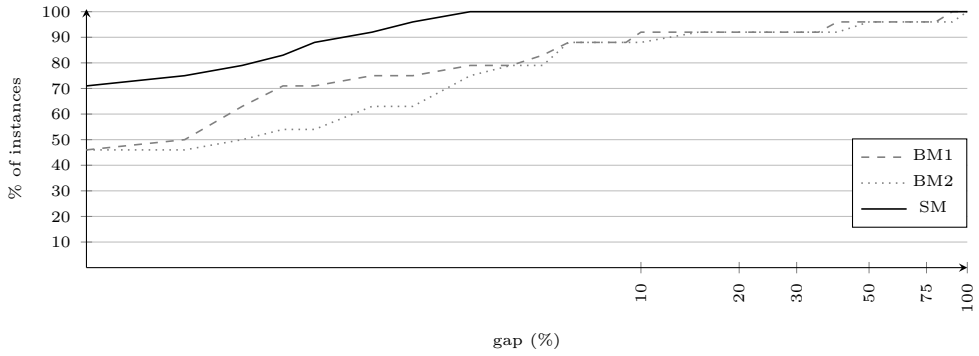


Figure 4.15: PUMF, T_2: performance profile of the Gap_0 (%).

is the most efficient model. With it, CPLEX can solve 92% of the instances in the 1 hour time limit. Moreover, it reveals that SM does not perform as well as it did for instances of T_1. When using SM, CPLEX only solves 71% of the instances within the time limit and, in general, the solving times are comparable to those obtained with BM1. However, it should be noted that the average gap between the best upper and lower bound found at the end of the run, was only 1.3% for instances that were unsolved using SM.

The below-par performance of SM with respect to the solving time of the MIPs, especially considering the strength revealed in Figure 4.10, can be explained by the following two factors: *i)* the LPs of SM are very slow to be solved (see Figure 4.13), and *ii)* the difference between the lower bounds for the basic and strong models at the end of the root node is not as pronounced as in the case of the lower bounds provided by the LP relaxations (see Figure 4.15). The latter implies that CPLEX is able to produce good cuts for the basic models. The faster MIP solving times for BM2, when compared with BM1, are probably explained by the smaller B&B trees (see Figure 4.14), as BM1 out-performs BM2 in both the LPs solving time (see Figure 4.13) and gap at the end of the root node (see Figure 4.15).

4.4 B&C algorithm

The computational experiments analyzed in the previous section revealed that despite typically yielding very tight LP bounds, the SM can struggle to solve some instances to optimality, due to the large size of its LPs. In this section, we propose a B&C algorithm, which we hope will be able to benefit from SM's strength, while solving a more compact model. For that purpose, we propose a Benders' decompositions, in Sections 4.4.1, which we integrate in the proposed B&C algorithm. An introduction to the Benders' decomposition method can be found back in Section 1.6. In Section 4.4.2, we comment on the efficiency of this algorithm.

4.4.1 Benders' decomposition

The computational experiments with instances of test set T_2, analyzed in the previous section, showed that, for some instances, the LPs of SM can be lengthy to solve, especially when compared with the LPs of BM1. As the number of constraints does not increase with this strengthening of BM1, it is safe to assume that this is due to the disaggregation of variables x. As such, we could be tempted to project out these variables, while keeping the y-variables in the master problem. Let us consider a solution $\bar{y}$, that satisfies constraints (4.12c). The Benders' slave problem would then be the one in Formulation 4.15. However, it is is easy to understand that due to the relaxation in constraints (4.15e), a feasible solution to the problem in 4.15 can have multipath routing.

In this sense, we propose an alternative Benders' decomposition of SM. Let $\bar{x}$ be a feasible routing of the commodities (*i.e.* satisfying (4.12b,4.12f)). The corresponding Benders' slave problem $\text{SM}_{LP}(\bar{x})$ is seen in Formulation 4.16. Note that we can decompose this sub-problem by arc: $\text{SM}_{LP}(\bar{a}, \bar{x})$. Formulation 4.17 describes the dual

of $\text{SM}_{LP}(\bar{a}, \bar{x})$, $\text{SM}_D(\bar{a}, \bar{x})$, where α, γ and ζ variables are associated respectively with constraints (4.16b), (4.16c) and (4.16d), and constraints (4.17b) are linked to the y-variables.

The problem described in Formulation 4.17, can be both bounded or unbounded, depending on whether or not the sub-problem in Formulation 4.16 is feasible. As mentioned in Section 1.6, it can be very computationally hard to find extreme rays, in the case of an unbounded SM_D. As such, we set an upper bound on the objective function (4.17a) to a normalization positive value (*e.g.* 1), ensuring that the dual slave problem is always bounded. Note that if slave problem SM_{LP} is instead feasible, its optimal solution will be necessarily non-positive, as the y variables are non-negative (4.17e), and by definition, the intercepts f are non-positive. As such, the bound set on the objective function remains valid; it simply becomes irrelevant. From this problem we can infer the Benders' cut $\beta_{\bar{a},\bar{x}}(x) = -\bar{\alpha} - \sum_{s \in S_{\bar{a}}} \sum_{k \in K} [\min(\rho^k, b_{\bar{a}}^{s-1}) \bar{\gamma}^s - \rho^k \bar{\zeta}^s] \cdot x_{\bar{a}}^{ks}$, that we iteratively add to the master problem SM_M (see Formulation 4.18), following the procedure described in Algorithm 1.1, in Section 1.6. Note that for implementation purposes we set a lower bound of each variable Ψ_a in (4.18f), as otherwise the master would be unbounded in the first iterations of the Benders' algorithm.

$$\min \quad \sum_{a \in A} \sum_{s \in S_a} (c_a^s \sum_{k \in K} \rho^k x_a^{ks}) \tag{4.15a}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(i)} \sum_{s \in S_a} x_a^{ks} - \sum_{a \in \delta^-(i)} \sum_{s \in S_a} x_a^{ks} = \lambda_i^k, \qquad i \in N, k \in K \tag{4.15b}$$

$$- \sum_{k \in K} \min(\rho^k, b_a^{s-1}) x_a^{ks} \geq -b_a^{s-1} \bar{y}_a^s, \qquad a \in A, s \in S_a \tag{4.15c}$$

$$b_a^s \bar{y}_a^s \geq \sum_{k \in K} \rho^k x_a^{ks}, \qquad a \in A, s \in S_a \tag{4.15d}$$

$$x_a^{ks} \in [0, 1], \qquad a \in A, k \in K, s \in S_a : b_a^s \geq \rho^k \tag{4.15e}$$

Formulation 4.15: PUMF problem: Benders' slave problem, given $\bar{y}$.

$$\min \quad \sum_{a\in A}\sum_{s\in S_a} f_a^s y_a^s \tag{4.16a}$$

$$\text{s.t.} \quad -\sum_{s\in S_a} y_a^s \geq -1, \qquad a\in A \tag{4.16b}$$

$$-b_a^{s-1} y_a^s \geq -\sum_{k\in K} \min(\rho^k, b_a^{s-1})\bar{x}_a^{ks}, \qquad a\in A, s\in S_a \tag{4.16c}$$

$$b_a^s y_a^s \geq \sum_{k\in K} \rho^k \bar{x}_a^{ks}, \qquad a\in A, s\in S_a \tag{4.16d}$$

$$y_a^s \geq 0, \qquad a\in A, s\in S_a \tag{4.16e}$$

Formulation 4.16: PUMF problem: $\mathrm{SM}_{LP}(\bar{x})$.

$$\max \quad -\alpha - \sum_{s\in S_{\bar{a}}}\sum_{k\in K} \bar{x}_{\bar{a}}^{ks}[\min(\rho^k, b_{\bar{a}}^{s-1})\gamma^s - \rho^k\zeta^s] \tag{4.17a}$$

$$\text{s.t.} \quad -\alpha - b_{\bar{a}}^{s-1}\gamma^s + b_{\bar{a}}^s\zeta^s \leq f_{\bar{a}}^s, \qquad s\in S_{\bar{a}} \tag{4.17b}$$

$$\alpha \geq 0, \tag{4.17c}$$

$$\beta^s \geq 0, \qquad s\in S_{\bar{a}} \tag{4.17d}$$

$$\gamma^s \geq 0, \qquad s\in S_{\bar{a}} \tag{4.17e}$$

Formulation 4.17: PUMF problem: $\mathrm{SM}_D(\bar{a}, \bar{x})$.

$$\min \quad \sum_{a\in A}\sum_{s\in S_a}\left(\Psi_a + c_a^s \sum_{k\in K}\rho^k x_a^{ks}\right) \tag{4.18a}$$

$$\text{s.t.} \quad \sum_{a\in\delta^+(i)}\sum_{s\in S_a} x_a^{ks} - \sum_{a\in\delta^-(i)}\sum_{s\in S_a} x_a^{ks} = \lambda_i^k, \qquad i\in N, k\in K \tag{4.18b}$$

$$\Psi_a \geq \beta_{a,\bar{x}^{k'}}(x), \qquad a\in A, k'=1...k \tag{4.18c}$$

$$\beta_{a,\bar{x}^{k'}}(x) \leq 0, \qquad a\in A, k'=1...k \tag{4.18d}$$

$$x_a^{ks} \in \{0,1\}, \qquad a\in A, k\in K, s\in S_a : b_a^s \geq \rho^k \tag{4.18e}$$

$$\Psi_a \geq f_a^{|S_a|}, \qquad a\in A \tag{4.18f}$$

Formulation 4.18: PUMF problem: SM_M^k.

4.4.2 Computational experiments for the B&C algorithm

The implementation of this B&C algorithm follows the implementation of the B&C algorithm proposed for the TE-MSTP problem, and described back in Section 2.5.2. As in that case, we also use in this B&C algorithm the in-out cut loop proposed in [FLS15], and presented in Section 1.6. This procedure requires one to provide a feasible solution, to serve as a stabilizing point. In Algorithm 4.1 we propose a greedy heuristic procedure, that quickly provides a feasible solution to the PUMF problem. In it, we consider a commodity at a time, by order of their demand. For each commodity, we can calculate the cost of sending it through every arc, by looking at the routing of the flow of the commodities that have already been considered. Thus, given that cost, we calculate the shortest path between the origin and destination of said commodity using Dijkstra's algorithm, and assign the corresponding values for the x variables.

We tested the efficiency of this algorithm in solving instances of test sets T_1 and T_2. The results revealed that this B&C algorithm is not a valid alternative to solving the PUMF problem. For 33 instances of T_1 and 10 of T_2 the memory required by the algorithm exceeded the available memory of the machine (see specifications in the beginning of Section 4.3). For the other instances, the algorithm often exceeded the time limit of one hour, and was always greater than the MIP time for solving the SM by itself.

Algorithm 4.1: Greedy heuristic for the PUMF problem.

$\mathcal{K} \leftarrow \{1, ..., |K|\}$
$\mathcal{F}_a \leftarrow 0, a \in A$
while $|\mathcal{K}| > 0$ **do**
  $k \leftarrow \arg\max_{k \in \mathcal{K}} \rho_k$
  $\mathcal{K} \leftarrow \mathcal{K} \backslash \{k\}$
  $\bar{s}_a \leftarrow s \in S_a : b_a^{s-1} \leq \mathcal{F}_a + \rho_k \leq b_a^s$
  $\mathcal{W}_a \leftarrow f_a^{\bar{s}_a} + c_a^{\bar{s}_a}(\mathcal{F}_a + \rho_k), a \in A$
  $\mathcal{W}_a \leftarrow \infty, a \notin A$
  Infer shortest path between o_k and d_k, with respect to $\mathcal{W}$
  Assign values for x^k accordingly
  Update loads $\mathcal{F}$

4.5 Strengthened aggregated formulation

In this section, we present a new MIP formulation for the PUMF problem. This formulation is a strengthening of BM1, presented in Section 4.2.1. It is obtained by adding a set of valid inequalities that are inferred by generalizing Benders' cuts obtained after a Benders' decomposition, which is proposed in Section 4.5.1 (this decomposition is different than the one described in the previous section). The model is presented in Section 4.5.2.

4.5.1 Benders' decomposition II

In this section, we propose an alternative Benders' decomposition for the PUMF problem, where we project the strong valid inequalities of this formulation onto the space of variables of BM1. As such, let $\bar{\mathcal{S}} = \{\overline{x}_a^k, \overline{l}_a^s, \overline{y}_a^s\}$ be a solution to the LP relaxation of BM1. If we define $\check{x}_a^{ks} = \bar{x}_a^k \overline{y}_a^s$, then $\check{\mathcal{S}} = \{\check{x}_a^{ks}, \overline{y}_a^s\}$ is a projection of $\bar{\mathcal{S}}$ to the space of variables in SM. The results of the computational experiments analyzed in Section 4.3, imply that for most instances, $\bar{\mathcal{S}}$ does not satisfy all the valid inequalities proposed for SM. Formulation 4.19 models a feasibility problem that checks whether or not solution $\bar{\mathcal{S}}$ is feasible for SM. Note that we can decompose this problem by arc. If $\bar{\mathcal{S}}$ is such that this (sub-)problem is infeasible, we want to infer an inequality that we can add to BM1 and accordingly cuts-off that point.

$$\max \quad 0 \tag{4.19a}$$

s.t.

$$\sum_{s \in S_a} x_a^{ks} = \overline{x}_a^k, \qquad a \in A, k \in K \tag{4.19b}$$

$$-\sum_{k \in K} \min(\rho^k, b_a^{s-1}) x_a^{ks} \leq -b_a^{s-1} \overline{y}_a^s, \qquad a \in A, s \in S_a \tag{4.19c}$$

$$\sum_{k \in K} \rho^k x_a^{ks} = \overline{l}_a^s, \qquad a \in A, s \in S_a \tag{4.19d}$$

$$x_a^{ks} \geq 0, \qquad a \in A, k \in K, s \in S_a : b_a^s \geq \rho^k \tag{4.19e}$$

Formulation 4.19: PUMF problem: Feasibility of the LP solution of BM1, with regards to the valid inequalities of SM.

$$\min \quad \sum_{k \in K} \check{x}_a^k \alpha^k - \sum_{s \in S_a} b_a^{s-1} \overline{y}_a^s \gamma^s + \sum_{s \in S_a} \overline{l}_a^s \zeta^s \tag{4.20a}$$

$$\alpha^k - \min(\rho^k, b_a^{s-1}) \gamma^s + \rho^k \zeta^s \geq 0, \qquad k \in K, s \in S_a : b_a^s \geq \rho^k \tag{4.20b}$$

$$\gamma^s \geq 0, \qquad s \in S_a \tag{4.20c}$$

Formulation 4.20: PUMF problem: Inferring a cut to strengthen BM1.

We can separate these cuts by solving the dual of Formulation 4.19, Formulation 4.20 for every arc $a \in A$. Variables α correspond to constraints (4.19b), γ to (4.19c) and ζ to (4.19d), whereas dual constraints (4.20b) are linked to the disaggregated x variables. If the optimum value of Formulation 4.20 is 0 for a given $\bar{\mathcal{S}}$ and arc $a \in A$, then $\bar{\mathcal{S}}$ satisfies all the constraints in Formulation 4.19 for that same arc. Conversely, if the problem is unbounded, then by the duality relationship, the problem in Formulation

4.19 is infeasible. As such, we derive a Benders' cut, $\sum_{k\in K} \vec{\alpha}^k x_a^k - \sum_{s\in S_a} b_a^{s-1}\vec{\gamma}^s y_a^s + \sum_{s\in S_a} \vec{\zeta}^s l_a^s \geq 0$, where $\vec{\alpha}$, are $\vec{\gamma}$ and $\vec{\zeta}$, are the extreme rays. As mentioned in Section 1.6, it can be very computationally hard to find extreme rays. As such, we fix the objective function in Formulation 4.20 to a normalization negative value (*e.g.* -1), and "transform" the dual problem into a bounded one. In the first Benders' decomposition, proposed in Section 4.4.1, we use a similar strategy for bounding the dual slave problem. However, as it is known that the choice of normalization can greatly impact the performance of the Benders' decomposition (see [FSZ10]), in the future it would be interesting to explore other strategies.

4.5.2 Valid inequalities for BM1

Note that a Benders' decomposition method based on the decomposition proposed in the previous section, can be used integrated in a B&C algorithm, in the same way as proposed in Section 4.4. Yet, the results of preliminary tests made with such an implementation were not very promising. These tests allowed us, however, to identify classes of strong valid inequalities for BM1. In this section, we present these inequalities.

The first valid inequality (4.21) states that if commodity $k \in K$ is sent through arc $a \in A$, then the load on that arc has to be at least ρ^k, and that arc a must be in a segment $s \in S_a$ such that $b_a^s \geq \rho^k$.

$$x_a^k \leq \frac{1}{\rho^k} \sum_{s\in S_a : b_a^s \geq \rho^k} l_a^s, \qquad a \in A, k \in K \tag{4.21}$$

Formulation 4.21: PUMF problem: Valid inequality #1 for the BM1.

We now give some intuition on the impact of these inequalities. Let us consider, for every $a \in A, k \in K$ inequalities $x_a^k \leq \frac{1}{\rho^k}\sum_{s\in S_a} l_a^s$; they are a result of weakening equalities (4.2d) in BM1, in order to consider only a single term on the left-hand side. Evidently, this inequality is redundant in BM1. Observe that valid inequalities (4.21) are a lifted version of these "weakened" inequalities. This type of transformation, where an equality is weakened by removing terms from one side, thus allowing for the exclusion of some cases on the other side, has been used in models for other network flow problems, usually leading to new formulations with strengthened LP bounds. Let us also assume that for a given LP solution, arc a is only used by one commodity, k; then, only a single variable x will be non-zero on the right-hand side of (4.2d). In that case, the corresponding "weakened" inequality is tight, in the sense that is satisfied as an equality. However, in such a case it is easy to understand the effect of valid inequality (4.21). Whereas with the "weakened" inequality the sum of all the l variables should be equal to $\rho^k x_a^k$, with (4.21) this value has to be attributed to the sum of a more restricted set of l variables - the ones for which the upper bound of the associated segment is not lower than the ρ_k. In conclusion, the value of the LP bound cannot be worse - and in some

cases, it will hopefully improve.

For the second set of valid inequalities (4.22), we need to make the assumption that if an arc $a \in A$ has no load, then $y_a^s = 0, s \in S_a$. Note that in BM1 and SM this is not imposed, and as $f_a^1 = 0, a \in A$, it would be possible to have a solution with $x_a^k = 0, k \in K$ and $y_a^1 = 1$, with the same cost as one with $x_a^k = 0, k \in K$ and $y_a^s = 0, s \in S_a$. This distinction has no implication in the outcome of the problem, nor the performance of the models. In that sense, inequalities (4.22) imply that for an arc to have non-zero load, and thus be in one of the segments in S_a, that arc must be on the routing path of at least one commodity.

$$\sum_{s \in S_a} y_a^s \leq \sum_{k \in K} x_a^k, \qquad a \in A \tag{4.22}$$

Formulation 4.22: PUMF problem: Valid inequality #2 for the BM1.

The structure of valid inequalities (4.22) is similar to the one of $\sum_{s \in S_a} b_a^{s-1} y_a^s \leq \sum_{k \in K} \rho^k x_a^k, a \in A$, obtained by summing up the left-hand inequalities (4.2e) for all $s \in S_a$, and replacing $\sum_{s \in S_a} l_a^s$ for $\sum_{k \in K} \rho^k x_a^k$, following equalities (4.2d). Consider, once again, an LP solution where only one commodity k is routed through arc a. The aforementioned aggregated inequality will only force x_a^k to be $\frac{b_a^{\bar{s}-1}}{\rho^k}$, where $\bar{s} : y_a^{\bar{s}} = 1$. When this ratio is less than 1, it is easy to see that (4.22) dominates the former inequality. In fact, in that particular case, inequality (4.22) has a similar effect to the one enforced by the aggregation of valid inequalities (4.12d) in SM, over all $s \in S_a$.

Finally, for the last set of valid inequalities, (4.23), let us define $\bar{s}_a^k = \{s : b_a^{s-1} \leq \rho^k \leq b_a^s\}$, and $\Xi_a^k = \frac{\rho^k}{b_a^{\bar{s}_a^k - 1}} - 1$. We now show that inequalities (4.23) are valid for the PUMF problem.

$$\rho^k y_a^{\bar{s}_a^k} + \Xi_a^k \sum_{s \in S_a : s < \bar{s}_a^k} l_a^s \leq l_a^{\bar{s}_a^k} + \Xi_a^k \sum_{k' \in K : k' \neq k} \rho^{k'} x_a^{k'}, \; a \in A, k \in K \tag{4.23}$$

Formulation 4.23: PUMF problem: Valid inequality #3 for the BM1.

Proof. We need to consider four cases: i) arc $a \in A$ is on the path used to route commodity $k \in K$, and arc a is on segment $\bar{s}_a^k$; ii) arc $a \in A$ is on the path used to route commodity $k \in K$, but arc a is not on segment $\bar{s}_a^k$; iii) arc $a \in A$ is not on the path used to route commodity $k \in K$, and arc a is on segment $\bar{s}_a^k$; iv) arc $a \in A$ is not on the path used to route commodity $k \in K$, and arc a is not on segment $\bar{s}_a^k$.

- Case i): We have that $y_a^{\bar{s}_a^k} = 1$, and thus $y_a^s = 0, s \in S_a : s < \bar{s}_a^k$, and consequently $l_a^s = 0, s \in S_a : s < \bar{s}_a^k$. The load on that arc $l_a^{\bar{s}_a^k}$, must be at least ρ^k;

- Case ii): If arc a is on the path used to route commodity $k \in K$, but arc a is not on segment $\bar{s}_a^k$, then arc a must be on a segment $s \in S_a : s > \bar{s}_a^k$. This implies that the left-hand side of (4.23) is null. As $\Xi_a^k \geq 0, a \in A, k \in K$, (4.23) is valid for this case;

- Case iii): We can re-write the right-hand side of (4.23) as $l_a^{\bar{s}_a^k} + (\frac{\rho^k}{b_a^{\bar{s}_a^k - 1}} - 1) l_a^{\bar{s}_a^k}$. This can be reduced to $\frac{\rho^k}{b_a^{\bar{s}_a^k - 1}} l_a^{\bar{s}_a^k}$. From (4.2e) we know that $\frac{l_a^{\bar{s}_a^k}}{b_a^{\bar{s}_a^k - 1}} \geq 1$; so $\frac{\rho^k}{b_a^{\bar{s}_a^k - 1}} l_a^{\bar{s}_a^k} \geq \rho^k y_a^{\bar{s}_a^k}$;

- Case iv): Cuts (4.23) are re-written as $\sum_{s \in S_a : s < \bar{s}_a^k} l_a^s \leq \sum_{k' \in K : k' \neq k} \rho^{k'} x_a^{k'}$, which is clearly valid.

□

We denote as the strengthened aggregated model (SAM), the combination of BM1 and valid inequalities (4.21-4.23). For this model we also redefine $S_a, a \in A$ as the set of segments $s \in S_a$, whose flow upper bound b_a^s is not smaller than the lowest demand.

At the time the author has concluded writing this thesis, only a few preliminary experiments had been done to properly test the performance of SAM. These tests were conducted using test set T_2, as previous results revealed that SM did not perform as well for these particular instances (see Section 4.3). The results of these preliminary tests seem to imply that SAM is not more efficient than SM. Note that there are $|A||K|$ many constraints of type (4.21) and (4.23). Despite being polynomially many, this still translates in a sizable increase when comparing to BM1, especially for instances like the ones in T_2 where there are a lot of commodities. Therefore, it makes sense to generate these constraints on the fly as cutting planes, rather than considering them all *a priori*. As such, it is possible that the performance of SAM can improve, following the implementation of a better, faster, cut separation procedure.

An important result to take from the preliminary tests regards the strength of SAM. For every instance of both T_1 and T_2, the lower bounds provided by the LP relaxations of SAM and SM were the same. An interesting question that remains open is whether or not these two models are in fact equivalent.

4.6 Non-convex case

In the previous sections, we have considered the case where the piecewise linear function was convex. However, we can also define a unsplittable multicommodity flow problem with non-convex piecewise linear cost functions (NPUMF). This problem is a direct extension to the NCPMF problem (discussed in [CGM03], [CGM07] and [GG14]), with the added constraint of having single-path routing for each commodity. In order to characterize the non-convex piecewise linear cost functions, we resort to the notation described in the beginning of this chapter, and illustrated in Figure 4.1. However, some

new assumptions about these functions are made: the slopes c_a^s are non-negative as well, but no monotonicity is imposed and the intercepts f_a^s are not necessarily non-positive. The functions are not necessarily continuous, but we assume them to be lower semi-continuous (*i.e.*, $g_a(l_a) \leq \liminf_{l'_a \to l_a} g_a(l'_a)$, for any sequence l'_a that approaches l_a). Figure 4.16 illustrates two examples of non-convex piecewise linear functions: one concave (black) and another non-concave (gray).

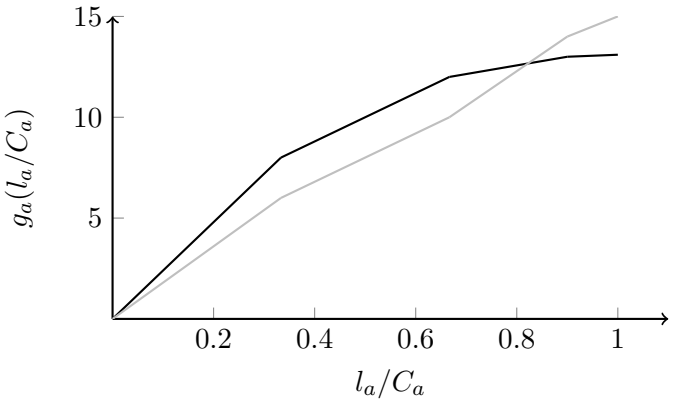


Figure 4.16: Two examples of non-convex piecewise linear cost functions.

In this section, we analyze the performance of our proposed models for instances of the NPUMF problem. First, observe that while BM2 cannot be used to model the NPUMF problem, both BM1 and DM are valid formulations for the problem. SM is also a valid formulation for the NPUMF problem; however, as discussed in Section 4.2, for cases where $f_a^s \geq 0$, a better strengthening can be obtained from valid inequalities (4.13), rather than from inequalities (4.12d). As such, for clarity, we include SM readapted for this case, Formulation 4.24; we name it as the Strong Model for the Non-Convex Case (SM-n).

Note that since for the NPUMF problem the cost functions are non-convex, the y-variables in BM1 and SM-n need to be considered as binary.

In our analysis, we also explore the impact of imposing unsplittable flows (NPUMF problem), versus allowing each commodity to be sent through multiple paths (NCPMF problem). In this sense, we need to adapt our models to solve the NCPMF problem. To this purpose, we define the Relaxed Basic Model 1 (BM1-r), where the integrality of variables x (4.2f) is relaxed. We can also consider the same relaxation in constraint (4.24e) of SM-n; we name this formulation as the Relaxed Strong Model (SM-r). Observe that if the flows can be split, we cannot apply valid inequalities (4.24g), and as such, we exclude them from SM-r. Both BM1-r and SM-r have been proposed in [CGM07] to model the NCPMF problem.

In our computational experiments we tested these models, using instances that had been previously developed for the NCPMF problem [GG14]. For test set T_n, we selected 40 instances, that describe 10 different networks, whose characteristics are detailed in

Network ID	$\|V\|$	$\|A\|$	$\|K\|$
1	20	100	50
2	20	100	50
3	25	100	100
4	25	150	100
5	25	150	100
6	40	300	100
7	40	300	400
8	40	400	400
9	50	400	400
10	50	600	400

Table 4.5: Description of each class of instances for the NPUMF.

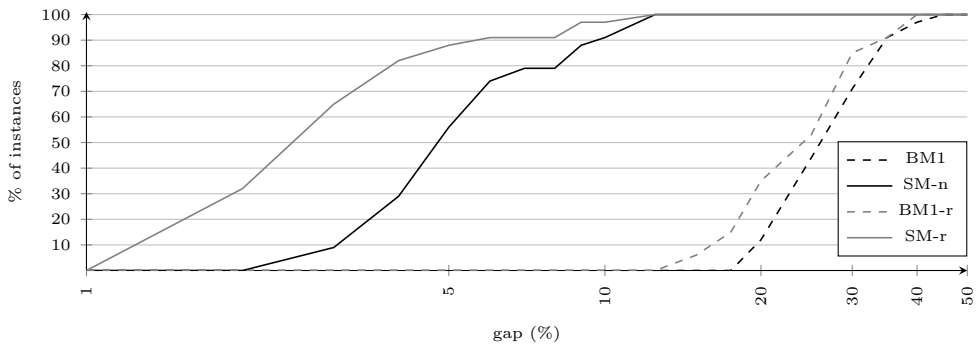


Figure 4.17: NPUMF, T_n: performance profile of Gap_{LP} (%).

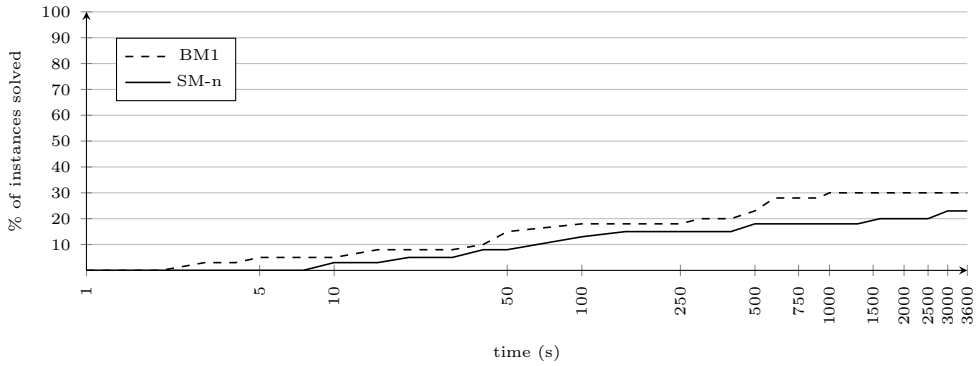


Figure 4.18: NPUMF, T_n: performance profile of MIP solving times (s).

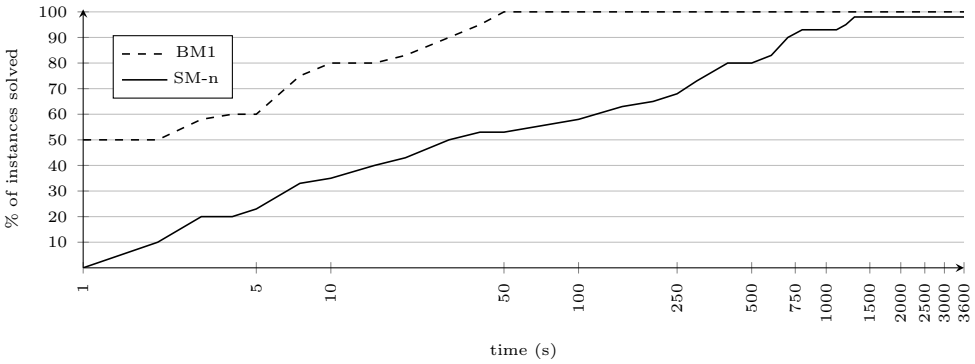


Figure 4.19: NPUMF, T_n: performance profile of LP solving times (s).

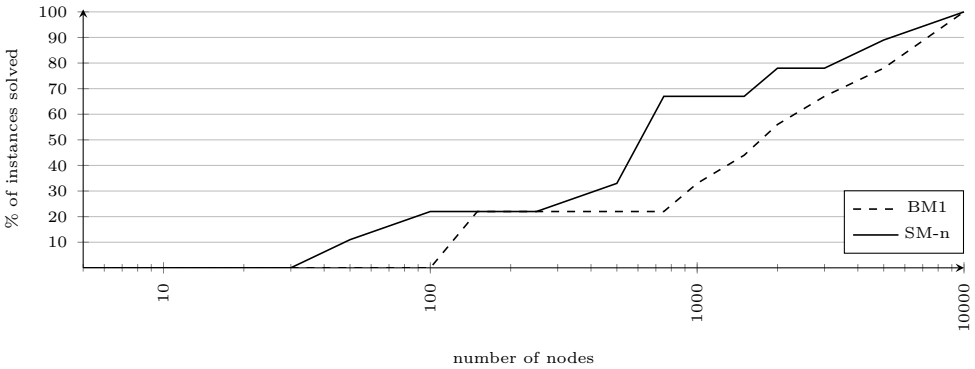


Figure 4.20: NPUMF, T_n: performance profile of B&B tree nodes.

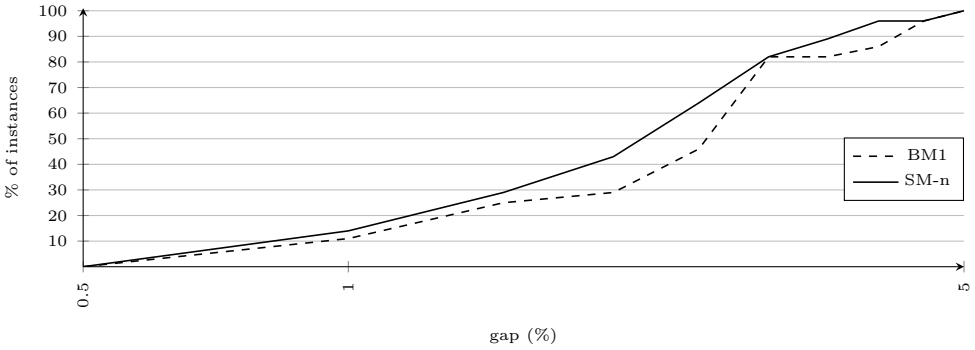


Figure 4.21: NPUMF, T_n: performance profile of Gap_0 (%).

$$\min \quad \sum_{a\in A}\sum_{s\in S_a}\left(f_a^s y_a^s + c_a^s \sum_{k\in K}\rho^k x_a^{ks}\right) \tag{4.24a}$$

$$\text{s.t.} \quad \sum_{a\in A}\sum_{s\in S_a} x_a^{ks} - \sum_{a\in A}\sum_{s\in S_a} x_a^{ks} = \lambda_i^k, \qquad i\in V, k\in K, \tag{4.24b}$$

$$\sum_{s\in S_a} y_a^s \leq 1, \qquad a\in A, \tag{4.24c}$$

$$b_a^{s-1} y_a^s \leq \sum_{k\in K}\rho^k x_a^{ks} \leq b_a^s y_a^s, \qquad a\in A, s\in S_a, \tag{4.24d}$$

$$x_a^{ks}\in\{0,1\}, \qquad a\in A, k\in K, s\in S_a, \tag{4.24e}$$

$$y_a^s\in\{0,1\}, \qquad a\in A, s\in S_a, \tag{4.24f}$$

$$x_a^{ks} = 0, \qquad a\in A, k\in K, s\in S_a : b_a^s < \rho^k, \tag{4.24g}$$

$$x_a^{ks} \leq y_a^s, \qquad k\in K, a\in A, s\in S_a. \tag{4.24h}$$

Formulation 4.24: PUMF problem: SM-n.

Table 4.5. Despite having the same number of nodes, arcs and commodities, network 1 (network 4) has tighter arc capacities $b_a^{|S_a|}$, than network 2 (network 5). For each network, there are 4 different instances based on the characteristics of the non-convex piecewise linear cost functions. There are 2 instances with concave functions, and 2 instances with non-concave functions. For each type of cost functions, there is an instance with up to 4 segments, and another with up to 8 segments. Every segment has non-negative intercept f, even when the cost functions are non-concave. Finally, note that in the original instances, each commodity had one source and multiple targets. To adapt the instances to the NPUMF problem, we separated each commodity to many, such that each has a single origin and a single destination. The results of our computational experiments for the NPUMF problem are detailed in Table F.1, in Appendix F.

In Figure 4.17, we present the performance profile of the Gap_{LP}, for the NPUMF and the NCPMF problems. For this graph, we only considered the 34 instances, for which we were able to obtain an upper bound for both the NPUMF and NCPMF problems, and lower bounds for all four formulations (BM1, SM-n, BM1-r and SM-r) in the time limit of 1 hour. As before, it reveals that SM-n produces much better lower bounds than BM1. It also shows that, as expected, the gaps for the unsplittable flows case are higher than for the splittable flows one. However, it is noteworthy that the maximum observed gaps for SM-n and SM-r are similar. Moreover, one should take into account that it is easier for CPLEX to find good (or optimal) solutions for the NCPMF problem, than for the NPUMF problem. As such, it is likely that in many cases, the quality of upper bounds used to calculate the Gap_{LP} of SM-n are not very good, and the values for the Gap_{LP} are, ultimately, greatly overestimated.

Next, we analyze the behaviour of CPLEX for instances of test set T_n, while using

BM1 and SM-n. As it was done for previous experiments, we tested which CPLEX's LP optimizer method was faster to solve the instances of T_n. These preliminary experiments showed that while for BM1, CPLEX's default choice works best, for SM-n the Barrier method generally improves the LP solving time. In Figure 4.18, we depict the performance profile of the solving time of the MIPs of T_n. Similarly to what happens for instances of T_2 (see Subsection 4.3.3), BM1 is faster in solving instances of T_n than SM-n. However, in this case, the number of instances solved in the time limit is very low (30%). This is probably explained by the long solving times of the LPs of SM-n (see Figure 4.19). While all the LPs of BM1 are solved in under 50 seconds, not all the LPs of SM-n were solved in the time limit of 1 hour. Moreover, in Figure 4.21 we can observe that for BM1, CPLEX is able to automatically generate strong cuts, that highly improve the lower bounds at the root nodes, approaching them to the ones of SM-n. Figure 4.20 seems to indicate that the B&B trees for SM-n are smaller than for BM1. Notwithstanding, in our performance profiles of the B&B trees nodes (Figures 4.8, 4.14 and 4.20), we only consider instances that were solved in the time limit by all models, in order to keep the results consistent. Seeing that only 9 out of the 40 instances of test set T_n are in this condition, the results depicted in Figure 4.20 might not be significant.

4.7 Summary and remarks

In this chapter, we presented a special case of the multicommodity flow problem, for which the flow of each commodity is unsplittable, and the routing costs on the arcs are given by piecewise linear functions. We focus on the case where the cost functions are convex - the PUMF problem. We show that the PUMF problem is $\mathcal{NP}$-hard for the general case, but polynomial when there is only a single commodity. We begin by proposing two basic MIP formulations for this problem, BM1 and BM2. By disaggregating the flow variables in BM1, we are able to develop valid inequalities that we include in a new MIP formulation: SM. We show that the solution of the LP relaxation of SM is always integer, in the single-commodity case.

We present results of extensive computational experiments, done over instances of two different test sets: in T_1 we randomly generate our own instances, while in T_2 we adapt instances of the SNDlib. These experiments reveal that the LP relaxation of SM is able to provide very good lower bounds, far better than the ones provided by the LP relaxation of the basic models. In fact, for 93% of the instances in T_1 and for 58% of the instances of T_2, the LP gap of SM is null. In [LMPM14], it was shown that for large topologies, splittable multicommodity flow problems with piecewise linear costs (as well as for others types of objective) essentially become unsplittable flow problems, since at optimality each demand tends to use a single path, despite the existence of multiple paths. These results might explain the very good bounds that we were able to obtain for large networks, as it implies that the optimal solutions of the LP relation of the PUMF problem are integer.

Despite being the fastest model to solve instances of T_1, SM did not perform as well for instances of T_2 as BM1 and BM2. This is explained by the large LPs, that can take

a long time for CPLEX to solve. As such, we propose two Benders' decompositions of SM, in an attempt to profit from the models strength, while reducing the size of the LPs. The first Benders' decomposition is integrated in a B&C algorithm; our computational experiments show however, that this method is not efficient in solving the PUMF problem. In the second Benders' decomposition, we project the strong valid inequalities of SM onto BM1. Through the intrepretation of the resulting Benders' cuts, we were able to identify a well-defined set of valid inequalities for the latter. We denote as SAM, the combination of BM1 and these strong valid inequalities. Preliminary tests seem to imply that the bounds obtained by the LP relaxation of SAM are the same as the ones obtained by the LP relaxation of SM. In the future, we would like to investigate whether the models are provably equivalent. Despite having less variables than SM, preliminary tests did not yield better MIP solving times for SAM when compared to the other models. This might be explained by the large number of additional constraints. Nonetheless, it is possible that a better and faster cut separation procedure, might result in an overall more efficient B&C algorithm.

We also consider the case where the cost functions are non-convex. This problem, denoted NPUMF, is an extension of the NCPMF problem, where the flows can be split and sent through multiple paths [CGM03], [CGM07] and [GG14]. We adapt our SM to better adapt to characteristics of the NPUMF problem, and we name it SM-n. We test the performance of BM1 and SM-n in solving instances proposed in [GG14]. Our results reveal that, despite the added difficult of the NPUMF problem, in relation to the NCPMF problem, the LP gaps are only slightly worse. This might be explained by valid inequalities (4.24g) that are not valid for the NCPMF problem.

Both the PUMF and the NPUMF problems are generic problems, that might not be applicable to real life problems by itself. However, they have fundamental characteristics that can be identified in other more complex problems, in *e.g.* telecommunications and transportation. As such, the study described in this chapter is essential, in order to have a better understanding of how to model piecewise linear functions, in the context of single-path routing. In the next chapter, we make use of convex piecewise linear flow cost functions, in an optimization problem where the objective is to avoid that the edges of a network implementing the MSTP are heavily loaded.

Chapter 5

MSTP: minimization of piecewise linear flow cost functions

In Chapter 2, we described the TE-MSTP problem, whose aim is to find designs for networks implementing the MSTP, with low link utilization. In this sense, we considered the objective of this optimization problem to be the minimization of the worst-case link utilization. In Chapter 3, we presented the COCMST problem, whose objective is also to obtain optimal network designs for the MSTP, but with a different objective: the minimization of the total load. The results of computational experiments revealed some tendency for this problem to be easier than the TE-MSTP problem. Regardless, it is easy to understand that the best designs for the COCMST problem can differ significantly from the best designs of the TE-MSTP problem. Whereas for the TE-MSTP problem, the tendency is for the VLANs and their respective flows to be spread among all the links in the network; in the COCMST problem the tendency is to assign shorter paths for the "heavier" demands, and leave the longer paths to the "smaller" ones.

In this chapter, we consider a novel network design problem for the MSTP, that combines characteristics of the TE-MSTP problem and the PUMF problem, studied in the previous chapter. As such, let us consider an undirected graph $G = (N, E)$, with edge capacities C_e; a set of VLANs T; a set of commodities K^t for every $t \in T$, where for each $k \in K^t$ has origin origin $o_k \in N$, destination $d_k \in N$, and demand ρ_k; and a piecewise linear convex load cost function g_e for each edge $e \in E$, where each segment $s \in S_e = \{1, 2, ..., |S_e|\}$, is defined by two breakpoints b_e^{s-1} and b_e^s ($b_e^{|S_e|} = C_e$), a slope c_e^s and an intercept f_e^s. We denote as the TE for MSTP problem with piecewise linear costs (TE-MSTP-p) to the problem of finding a design for all VLANs $t \in T$, such that:

- the topology of each VLAN is a spanning tree;
- all given traffic demands in a VLAN are routed;
- the sum of the cost given by the piecewise linear load cost function of each edge, is minimized.

We believe that this problem might be a valid alternative to the TE-MSTP problem, in the sense that it also favours solutions with low link utilization. However, we can expect some difference between the solutions of both problems. While for the TE-MSTP problem the sole focus is in the minimization of the "most used" link utilization, for the TE-MSTP-p problem the objective is to have an overall low link utilization.

It is easy to see that this problem too is $\mathcal{NP}$-hard (reduction from the OCST problem). In Section 5.1 we present two MIP formulations, based on models proposed in the previous chapters for the TE-MSTP and PUMF problems. In Section 5.2, computational experiments are analyzed, in order to compare the performance of both formulations, and in order to compare the designs obtained by the TE-MSTP and the TE-MSTP-p problems. In Section 5.3, we propose a B&C algorithm for this problem. Finally, in Section 5.4 we draw conclusions.

5.1 Problem formulation

In this section, we propose two MIP formulations for the TE-MSTP-p problem. These models are combinations of the models that we proposed for the TE-MSTP problem (see Section 2.2) and PUMF problem (see Section 4.2). Note that many combinations are possible. However we focus on two models that we believe have more potential to perform well.

Let us consider the three main components of the TE-MSTP-p problem:

Sub-problem 1: Designing spanning trees;

Sub-problem 2: Routing the traffic demands;

Sub-problem 3: Pricing the edge loads.

Sub-problem 1 and 2 are common to the TE-MSTP problem, and were studied back in Section 2.2. One proposal included adapting Kipp Martin's formulation for the MST problem to design multiple spanning trees, while using multicommodity flow variables to route the traffic demands. This combination resulted in the RDMFM, and both the polyhedral studies in Section 2.3 and the analysis of computational experiments in Section 2.4 implied that it is the most promising. For the sake of completeness, we repeat these models in Formulation 5.1 and 5.2. In order to link these two models, we use the linking constraints in Formulation 5.3.

In order to solve the third sub-problem, we look into the models proposed in the previous chapter, for the PUMF problem (see Section 4.2). Let us recall BM2, that proved to be efficient in solving instances of test set T_2 (see Section 4.3.3). Sub-problem 3 can be modeled, adapting BM2 to the multiple VLANs case, as seen in Formulation 5.4. We combine these models into a single one that solves the TE-MSTP-p, the Basic RDMFM (B-RDMFM); the objective function in this model is (5.5).

We have also proposed a strengthened formulation for the PUMF problem, that obtained very tight LP bounds in the computational experiments (see Section 4.3). In

$$\sum_{a \in \delta^-(j)} z_a^{ut} = 1, \qquad u, j \in N : u \neq j, t \in T \tag{5.1a}$$
$$z_{ij}^{ut} + z_{ji}^{ut} = w_e^t, \qquad u \in N, e = \{i, j\} \in E, t \in T \tag{5.1b}$$
$$z_{ij}^{ut} \in \{0, 1\}, \qquad u \in N, (i, j) \in A : j \neq u, t \in T \tag{5.1c}$$
$$w_e^t \in \{0, 1\}, \qquad e \in E, t \in T \tag{5.1d}$$

Formulation 5.1: TE-MSTP-p SP1: rooted directed formulation.

$$\sum_{a \in \delta^+(i)} x_a^{tk} - \sum_{a \in \delta^-(i)} x_a^{tk} = \lambda_i^k, \qquad i \in N, t \in T, k \in K^t \tag{5.2a}$$
$$x_a^{tk} \in \{0, 1\}, \qquad a \in A, t \in T, k \in K^t \tag{5.2b}$$

Formulation 5.2: TE-MSTP-p SP2: multicommodity flow formulation.

$$x_{ij}^{tk} \leq z_{ij}^{ut}, \qquad (i, j) \in A, u \in N, t \in T, k \in K^t : o_k = u \tag{5.3a}$$
$$x_{ij}^{tk} \leq z_{ji}^{vt}, \qquad (i, j) \in A, v \in N, t \in T, k \in K^t : d_k = v \tag{5.3b}$$

Formulation 5.3: TE-MSTP-p problem: linking constraints between SP1 and SP2.

$$g_e \geq f_e^s + c_e^s \sum_{t \in T} \sum_{k \in K^t} \rho^k (x_{ij}^{tk} + x_{ji}^{tk}), \qquad e = \{i, j\} \in E, s \in S_e \tag{5.4a}$$
$$g_e \geq 0, \qquad e \in E \tag{5.4b}$$

Formulation 5.4: TE-MSTP-p SP3: "basic" formulation.

$$\min \quad \sum_{e \in E} g_e \tag{5.5}$$

Formulation 5.5: TE-MSTP-p problem: "basic" objective function.

this formulation we disaggregate the x variables, such that it also contains the s index. We can substitute $x_a^{tk}, a \in A, t \in T, k \in K^t$ in Formulations 5.2 and 5.3 by $\sum_{s \in S_a} x_a^{tks}$, where $S_a = S_e$ if $e = \{i, j\} \in E$ and $a = (i, j)$ or $a = (j, i)$; this is shown in Formulations 5.6 and 5.7, respectively. Finally, Sub-problem 3 is modelled as seen in Formulation 5.8, by adapting the strong valid inequalities of SM to multiple VLANs. We can also consider

the set of valid inequalities (5.9), that link Sub-problems 1 and 3, by connecting the w and the y variables; empirical evidence reveals that they tend to improve the bound given by the LP relaxation. We combine these models to solve the TE-MSTP-p problem; the resulting formulation is denominated as Strengthened RDMFM (S-RDMFM). Table 5.1 summarizes the composition of both B-RDMFM and S-RDMFM.

$$\sum_{a\in\delta^+(i)}\sum_{s\in S_a} x_a^{tks} - \sum_{a\in\delta^-(i)}\sum_{s\in S_a} x_a^{tks} = \lambda_i^k, \qquad i\in N, t\in T, k\in K^t \tag{5.6a}$$

$$x_a^{tks}\in\{0,1\}, \qquad a\in A, s\in S_a, t\in T, k\in K^t \tag{5.6b}$$

Formulation 5.6: TE-MSTP-p SP2: disaggregated multicommodity flow formulation.

$$\sum_{s\in S_a} x_a^{tks} \le z_{ij}^{ut}, \qquad (i,j)\in A, u\in N, t\in T, k\in K^t : o_k = u \tag{5.7a}$$

$$\sum_{s\in S_a} x_a^{tks} \le z_{ji}^{vt}, \qquad (i,j)\in A, v\in N, t\in T, k\in K^t : d_k = v \tag{5.7b}$$

Formulation 5.7: TE-MSTP-p problem: disaggregated linking constraints between SP1 and SP2.

$$\sum_{s\in S_e} y_e^s \le 1, \qquad e\in E \tag{5.8a}$$

$$b_e^{s-1} y_e^s \le \sum_{t\in T}\sum_{k\in K^t} \min(\rho^k, b_e^{s-1})(x_{ij}^{tks} + x_{ji}^{tks}), \qquad e=\{i,j\}\in E, s\in S_e \tag{5.8b}$$

$$b_e^s y_e^s \ge \sum_{t\in T}\sum_{k\in K^t} \rho^k (x_{ij}^{tks} + x_{ji}^{tks}), \qquad e=\{i,j\}\in E, s\in S_e \tag{5.8c}$$

$$x_a^{tks}\in\{0,1\}, \qquad a\in A, t\in T, k\in K^t, s\in S_a : b_a^s \ge \rho^k \tag{5.8d}$$

$$y_e^s\in\{0,1\}, \qquad e\in E, s\in S_e \tag{5.8e}$$

Formulation 5.8: TE-MSTP-p SP3: multiple choice formulation.

$$\sum_{t \in T} w_e^t \geq \sum_{s \in S_e} y_e^s, \qquad e \in E \tag{5.9}$$

Formulation 5.9: TE-MSTP-p problem: disaggregated linking constraints between SP1 and SP3.

$$\min \sum_{e=\{i,j\} \in E} \sum_{s \in S_e} \left(f_e^s y_e^s + c_e^s \sum_{k \in K} \rho^k (x_{ij}^{ks} + x_{ji}^{ks}) \right) \tag{5.10}$$

Formulation 5.10: TE-MSTP-p problem: disaggregated objective function.

Model	Variables	Obj	SP1	SP2	Link^{1,2}	SP3	Link^{1,3}
B-RDMF	$\{z_a^{ut}, w_e^t, x_a^{tk}, g_e\}$	(5.5)	(5.1a-5.1c)	(5.2a-5.2b)	(5.3a-5.3b)	(5.4a-5.4b)	-
S-RDMF	$\{z_a^{ut}, w_e^t, x_a^{tks}, y_e^s\}$	(5.10)	(5.1a-5.1c)	(5.6a-5.6b)	(5.7a-5.7b)	(5.8a-5.8e)	(5.9)

Table 5.1: TE-MSTP-p problem: composition of each complete formulation.

Let us consider the LP relaxation of B-RDMFM and S-RDMFM, and let g_B^* and g_S^* be respectively the cost of the optimal solutions of those relaxations. It is obvious from the studies in the previous chapter that $g_S^* \geq g_B^*$.

5.2 Computational experiments

In this section, we discuss the results of the computational experiments described in Tables G.1 to G.3, in Appendix G. These tests were made in order to evaluate the performance of B-RDMFM and S-RDMFM in solving the TE-MSTP-p problem. All the tests were performed on a Intel Core i7 CPU 960 @ 3.20GHz (x8) with 12GB of memory with 64 bits, and running Ubuntu 14.04.2 LTS (GNU/Linux $3.2.0-26-$generic x86_64). The tests were done using the MIP solver ILOG CPLEX 12.6, implemented in Java programming language. We only allow CPLEX to use one thread of the machine's processor.

For these tests, we adapted instances of test sets T_{rand} and T_{3tc}; more information regarding these instances can be found in Section 2.4.1. For every instance in these test sets, we assigned to each edge a convex piecewise linear cost function. These functions have 7 segments, with breakpoints $\{0, \frac{1}{10}C_e, \frac{3}{10}C_e, \frac{1}{2}C_e, \frac{7}{10}C_e, C_e, \infty\}$, where C_e is the capacity of the respective edge. The slopes for each segment are respectively 1, 3, 10, 70, 500, 5000.

5.2.1 Analysis of the results of test set T_{rand}

In this section, we analyze the results of our models to solve instances of T_{rand}, described in Tables G.1 and G.2, in Appendix G. Figure 5.1 depicts the performance profile of the Gap_{LP}. In it, we can observe that the lower bounds obtained by the LP relaxation of S-RDMFM are better than the ones obtained by the LP relaxation of B-RDMFM. Also, note that CPLEX is unable to improve this lower bound in the root node, via its automatically generated cuts. Nevertheless, this is not sufficient to make the stronger model more efficient. The MIP solving time performance profile in Figure 5.2 clearly identifies B-RDMFM as the faster model, when implemented in CPLEX: while with S-RDMFM we are only able to solve 62% of the instances in the time limit of one hour, with the former we are able to solve 76%. This is probably explained by the solving times of the LP, which are shorter for B-RDMFM (see Figure 5.3). Regarding the number of B&B nodes, whose performance profile can be seen in Figure 5.4, no model seems to clearly out-perform the other.

In the performance profiles of Figures 5.1 to 5.3, we also include the results for the RDMFM, the most efficient model for the TE-MSTP problem (see Section 2.4 for a complete analysis of the results). Note, that these results might differ from the ones presented in Section 2.4, as we re-ran the tests, only allowing CPLEX to use one thread of the machine's processor. Figure 5.2 shows that for "harder" instances, the MIP of RDMFM are faster to solve than the MIP inferred by the models for the TE-MSTP-p. This is clearly a result of the greater efficient in solving RDMFM's LPs, as the LP gaps actually tend to be larger for this model than for B-RDMFM and S-RDMFM. In Tables G.1 and G.2 we also compare the worst-case edge utilization observed in the optimal solutions of the TE-MSTP-p problem, with the minimal worst-case edge utilization calculated via solving the TE-MSTP problem. We see that the results are rather mixed, with the average relative difference being of 12%.

5.2.2 Analysis of the results of test set T_{3tc}

In this section, we analyze the results of our models to solve instances of T_{3tc}. Figure 5.5 depicts the performance profile of the Gap_{LP}. It shows that for most instances, the LP bounds obtained by the relaxation of B-RDMFM are the same as the ones obtained for S-RDMFM. As such, and given that i) the LPs of B-RDMFM are faster to solve than the ones of S-RDMFM (see Figure 5.7) and ii) the B&B trees of B-RDMFM are smaller than the ones of S-RDMFM (see Figure 5.8), it is natural that B-RDMFM is the most efficient model in obtaining optimal solutions to the TE-MSTP-p problem, as can be observed in Figure 5.6.

When comparing the results of the TE-MSTP-p problem and the TE-MSTP problem, we observe that the efficiency of solving the LPs of B-RDMFM, for instances of T_{3tc}, is comparable to the one of solving the LPs of RDMFM (see Figure 5.7). However, the MIPs of RDMFM are much faster to solve than the ones of B-RDMFM (see Figure 5.6). Regarding the Gap_{LP}, we can observe that even though there are many more instances that have gap 0 with RDMFM than with B-RDMFM and S-RDMFM, the worst observed

Gap_{LP} is larger for the former.

Finally, in Table G.3, we present the relative difference between the worst-case edge utilization of the optimal solutions of the TE-MSTP-p problem, and the minimal one, calculated by solving the TE-MSTP problem. The average relative difference is of 9%, slightly lower than for instances of T_{rand}.

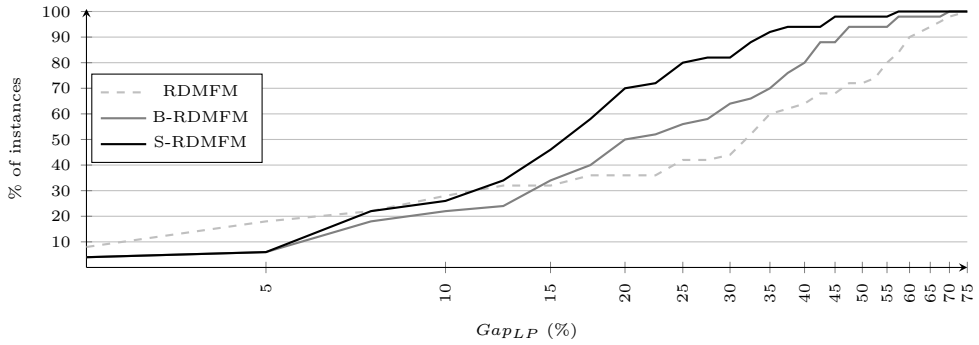


Figure 5.1: TE-MSTP-p, T_{rand}: performance profile of the Gap_{LP} (%).

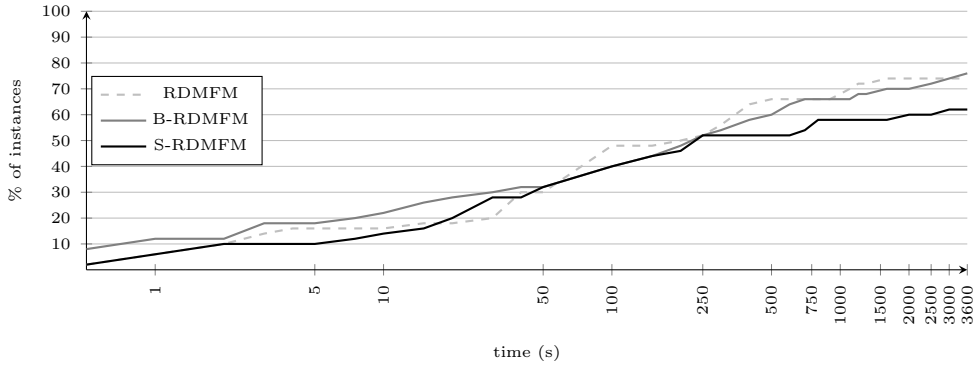


Figure 5.2: TE-MSTP-p, T_{rand}: performance profile of the MIP solving time (s).

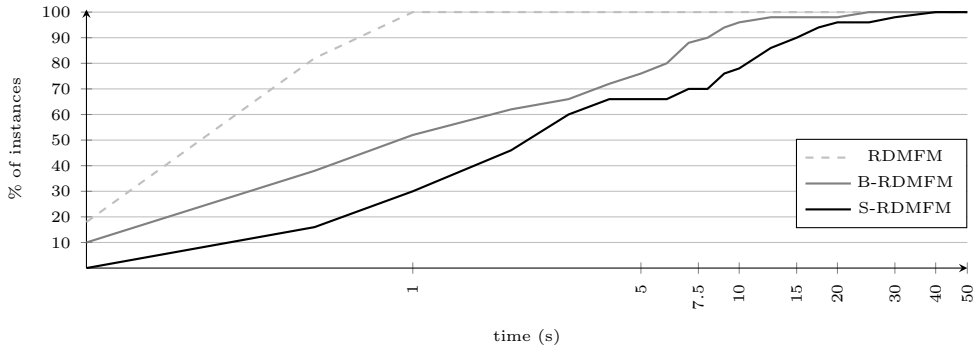


Figure 5.3: TE-MSTP-p, T_{rand}: performance profile of the LP solving time (s).

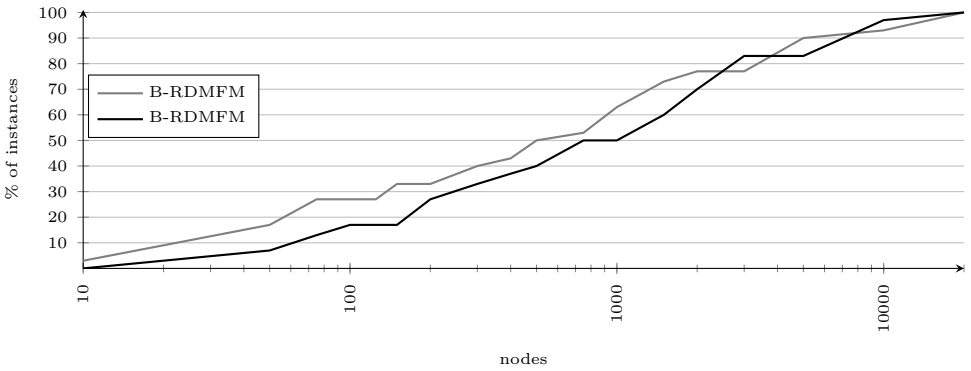


Figure 5.4: TE-MSTP-p, T_{rand}: performance profile of the B&B tree nodes.

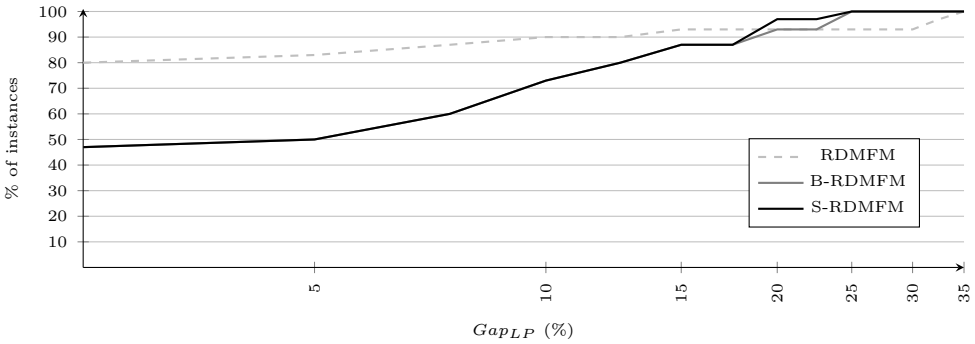


Figure 5.5: TE-MSTP-p, T_{3tc}: performance profile of Gap_{LP} (%).

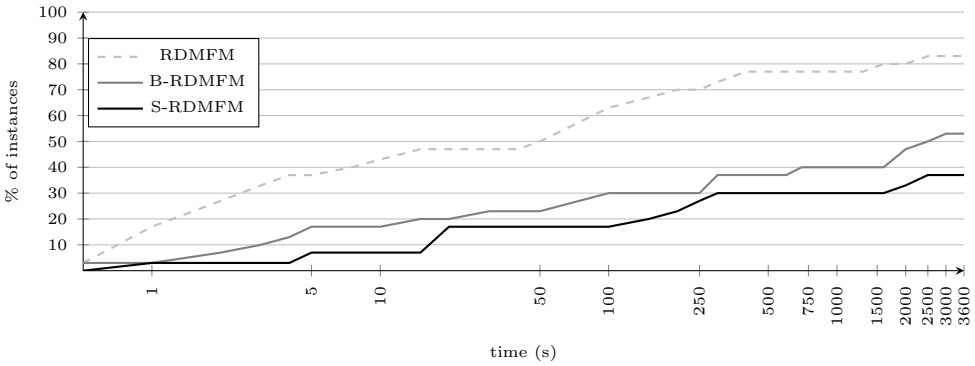


Figure 5.6: TE-MSTP-p, T_{3tc}: performance profile of the MIP solving time (s).

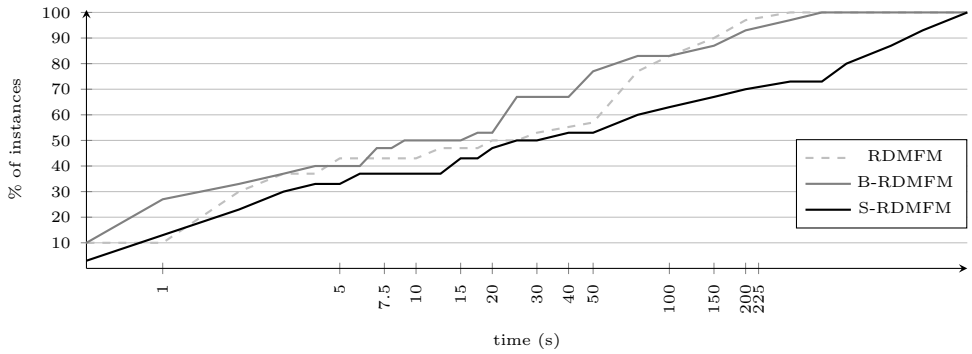


Figure 5.7: TE-MSTP-p, T_{3tc}: performance profile of the LP solving time (s).

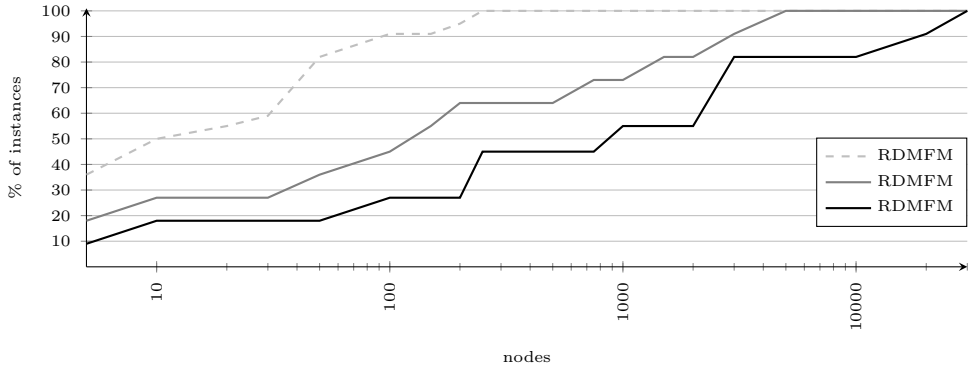


Figure 5.8: TE-MSTP-p, T_{3tc}: performance profile of the B&B tree nodes.

5.3 B&C algorithm

In this section, we propose a B&C algorithm for the TE-MSTP-p problem. In this algorithm, we make use of a Benders' decomposition of our strongest formulation, S-RDMFM. In Section 5.3.1, we describe the proposed Benders' decomposition. In Section 5.3.2, we analyze the performance of the B&C algorithm in solving instances of the TE-MSTP-p problem.

5.3.1 Benders' decomposition

Let us consider $\bar{z}$ and $\bar{w}$, a feasible assignment of variables z and w, such that they design a spanning tree for each VLAN. We define the slave problem of S-RDMFM in Formulation 5.11, S-RDMFM$_{LP}$, that routes the demands in each VLAN on the aforementioned trees, and calculates the cost of the corresponding solution. Formulation 5.12 represents the dual model of S-RDMFM$_{LP}$: variables α, γ, ζ, η, θ, ξ and σ are linked to constraints (5.11b), (5.11c), (5.11d), (5.11e), (5.11f), (5.11g), (5.11h), respectively; whereas dual constraints (5.12b) and (5.12c) correspond to variables x and y respectively.

Solving S-RDMFM$_D$ for different fixings of z and w, yields a Benders' cut $\beta_{\bar{z},\bar{w}}(w,z)$, that we iteratively add to the master problem S-RDMFM$_M$, following the Benders' method described in Algorithm 1.1, in Section 1.6. Let us define $\bar{\boldsymbol{\alpha}} = \sum_{t\in T}\sum_{k\in K^t}\alpha^{tk}_{o_k} - \alpha^{tk}_{d_k}$ and $\bar{\boldsymbol{\eta}} = \sum_{e\in E}\eta_e$. $\beta_{\bar{z},\bar{w}}(w,z)$ will be of the form $-\sum_{t\in T}[\sum_{k\in K^t}\sum_{(i,j)\in A}(\bar{\gamma}^{kt}_{ij}z^{o_k t}_{ij} + \bar{\zeta}^{kt}_{ij}z^{d_k t}_{ji}) - \sum_{e\in E}\bar{\sigma}_e w^t_e] + \bar{\boldsymbol{\alpha}} - \bar{\boldsymbol{\eta}}$.

5.3.2 Computational experiments for the B&C algorithm

We implemented the Benders' decomposition described in the previous section on a B&C framework, in the same way as it was done for the TE-MSTP and PUMF problems (see Sections 2.5.2 and 4.4.2 respectively). In order to generate a feasible solution that serves as a stabilizing point in the in-out cut loop, we use Algorithm 4.1, described in Section 2.5.2.

We tested this algorithm with instances of test sets T_{rand} and T_{3tc}; the results are described in Tables G.4 and G.5 in Appendix G. These results clearly indicate that the proposed algorithm is not an efficient alternative to simply solving the models in CPLEX. As it was the case for the B&C algorithm proposed for the TE-MSTP problem, the Benders' cuts seem to be weak which means that often we are unable to increase this lower bound at all, in the one hour time limit. And also likewise the slave problem for the TE-MSTP, the slave problem defined by Formulation 5.12 is not decomposable, which means that solving each cut is not sufficiently quick to make up for it.

$$\min \quad \sum_{e=\{i,j\}\in E} \sum_{s\in S_e} (f_e^s y_e^s + c_e^s \sum_{k\in K} \rho^k (x_{ij}^{ks} + x_{ji}^{ks})) \tag{5.11a}$$

$$\sum_{a\in\delta^+(i)} \sum_{s\in S_a} x_a^{tks} - \sum_{a\in\delta^-(i)} \sum_{s\in S_a} x_a^{tks} = \lambda_i^k, \qquad i\in N, t\in T, k\in K^t \tag{5.11b}$$

$$-\sum_{s\in S_a} x_a^{tks} \geq -\bar{z}_{ij}^{ut}, \qquad (i,j)\in A, u\in N, t\in T, k\in K^t : o_k = u \tag{5.11c}$$

$$-\sum_{s\in S_a} x_a^{tks} \geq -\bar{z}_{ji}^{vt}, \qquad (i,j)\in A, v\in N, t\in T, k\in K^t : d_k = v \tag{5.11d}$$

$$-\sum_{s\in S_e} y_e^s \geq -1, \qquad e\in E \tag{5.11e}$$

$$-b_e^{s-1} y_e^s + \sum_{t\in T} \sum_{k\in K^t} \min(\rho^k, b_e^{s-1})(x_{ij}^{tks} + x_{ji}^{tks}) \geq 0, \qquad e\in E, s\in S_e \tag{5.11f}$$

$$b_e^s y_e^s - \sum_{t\in T} \sum_{k\in K^t} \rho^k (x_{ij}^{tks} + x_{ji}^{tks}) \geq 0, \qquad e\in E, s\in S_e \tag{5.11g}$$

$$-\sum_{s\in S_e} y_e^s \geq -\sum_{t\in T} \bar{w}_e^t, \qquad e\in E \tag{5.11h}$$

$$x_a^{ks} \in \{0,1\}, \qquad a\in A, k\in K, s\in S_a : b_a^s \geq \rho^k \tag{5.11i}$$

$$y_a^s \in \{0,1\}, \qquad a\in A, s\in S_a \tag{5.11j}$$

Formulation 5.11: TE-MSTP-p problem: S-RDMFM$_{LP}(\bar{z}_{ij}^{ut}, \bar{w}^e)$.

$$\min \quad \sum_{t\in T} \sum_{k\in K^t} [\alpha_{o_k}^{tk} - \alpha_{d_k}^{tk} - \sum_{(i,j)\in A} (\bar{z}_{ij}^{o_k t} \gamma_{ij}^{kt} + \bar{z}_{ji}^{d_k t} \zeta_{ij}^{kt})] - \sum_{e\in E} [\eta_e - \sum_{t\in T} \bar{w}_e^t \sigma_e] \tag{5.12a}$$

$$\alpha_i^{tk} - \alpha_j^{tk} - \gamma_a^{kt} - \zeta_a^{kt} + \min\{\rho^k, b_e^{s-1}\}\theta_{\{i,j\}}^s - \rho^k \xi_e^s \leq c_{\{i,j\}}^s \rho^k,$$
$$a = (i,j)\in A, s\in S_a, t\in T, k\in K^t : b_a^s \geq \rho^k \tag{5.12b}$$

$$-\eta_e - b_e^{s-1}\theta_e^s + b_e^s \xi_e^s - \theta_e \leq f_e^s, \qquad e\in E, s\in S_e \tag{5.12c}$$

$$\gamma_a^{kt}, \zeta_a^{kt} \geq 0, \qquad a\in A, t\in T, k\in K^t \tag{5.12d}$$

$$\eta_e, \sigma_e \geq 0, \qquad e\in E \tag{5.12e}$$

$$\theta_e^s, \xi_e^s \geq 0, \qquad e\in E, s\in S_e \tag{5.12f}$$

Formulation 5.12: TE-MSTP-p problem: S-RDMFM$_D(\bar{z}_{ij}^{ut}, \bar{w}^e)$.

$$\min \quad \Psi \tag{5.13a}$$

$$\sum_{a \in \delta^-(j)} z_a^{ut} = 1, \qquad u, j \in N : u \neq j, t \in T \tag{5.13b}$$

$$z_{ij}^{ut} + z_{ji}^{ut} = w_e^t, \qquad u \in N, e = \{i, j\} \in E, t \in T \tag{5.13c}$$

$$\Psi \geq \beta_{\bar{z}^{k'}, \bar{w}^{k'}}(w, z), \qquad k' = 1...k \tag{5.13d}$$

$$z_{ij}^{ut} \in \{0, 1\}, \qquad u \in N, (i, j) \in A : j \neq u, t \in T \tag{5.13e}$$

$$w_e^t \in \{0, 1\}, \qquad e \in E, t \in T \tag{5.13f}$$

$$\Psi \geq 0 \tag{5.13g}$$

Formulation 5.13: TE-MSTP-p: S-RDMFM$_M^k$.

5.4 Summary and remarks

In this chapter, we considered another optimization problem, dealing with the design of networks implementing the MSTP, the TE-MSTP-p problem. In this problem, we make use of convex piecewise linear flow cost functions, in order to penalize heavily loaded edges. We were interested in investigating i) how similar are the optimal designs for this problem when compared to the ones obtained for the TE-MSTP problem, and ii) whether, given the good results obtained for the PUMF problem, this problem was easier to solve than the TE-MSTP problem.

To model the TE-MSTP-p problem, we proposed two MIP formulations, B-RDMFM and S-RDMFM, which are based on the combination of the most promising models previously proposed for the TE-MSTP and the PUMF problems. We tested the performance of these models, by adapting the instances of test sets T_{rand} and T_{3tc}, presented in Section 2.4.1. Even though these models obtain lower values for Gap_{LP}, than RDMFM obtained for the TE-MSTP problem, the TE-MSTP-p problem appears to take longer to be solved. This is probably explained by the large LPs of B-RDMFM and S-RMDFM. It is also interesting to note that the optimal solutions for the TE-MSTP-p somewhat differ from the ones obtained for the TE-MSTP problem: the average gap between the worst-case edge utilization observed in the solutions of the former, and the minimal one, calculated by solving the TE-MSTP problem is of 12% for instances of T_{rand} and of 9% for instances of T_{3tc}. One question that remains open, and it would interesting to investigate, is how the average edge utilization of the solutions of both problems relate.

In an attempt to counterweight the effect of the large LPs, we propose a B&C algorithm based on the Benders' decomposition of S-RDMFM. Unfortunately, this algorithm did not prove to be efficient, in solving the TE-MSTP-p problem.

The aforementioned results seem to imply that the bottleneck verified by solving network optimization problems dealing with multiple spanning trees is not in the type of objective function, but rather on the juxtaposition of these topologies and their share of common physical resources.

Chapter 6

Conclusion

In this thesis, we studied different network optimization problems, with single-path routing. In this type of routing, the entire flow associated with a given commodity must be relayed throughout the network over an unique path; *i.e.*, the flow cannot be "split" at any node. This type of routing is a common choice, as it tends to be a "cheaper" option than the alternative - multipath routing. Namely, online service providers enforce this type of routing in their large networks, by resorting to network management standards such as the MSTP [80206]. This protocol allows for the installation of many virtual networks over a single physical network. At the same time, the MSTP defines the topology of each VLAN as a spanning tree, therefore ensuring the compliance of the STP. Further information on these switching protocols can be found in the first sections of Chapter 1.

In this sense, we targeted our attention to optimization problems, whose aim is to find the best design for networks satisfying the MSTP, according to different performance metrics. These problems are highly combinatorial and often very hard to solve. This helps explain why the large majority of the contributions to the MSTP-related literature are on heuristic methods, that try to quickly create good solutions to the corresponding optimization problems. Regardless of the efficiency of these heuristics, network service providers can benefit from exact methods, which are able to provide provably optimal solutions, or at the very least yield lower bounds to the optimal values, that give some insight on the quality of the solutions given by the existing heuristics. The main contribution of our work is to fill this gap in the literature, by proposing mathematical programming formulations and exact methods for TE problems dealing with single-path routing, namely the one enforced by the MSTP.

In Chapter 2, we considered one first such problem, the TE-MSTP problem. In this problem, we must design networks implementing the MSTP, and route the different traffic demands, such that the worst-case edge utilization is minimized. We proposed three different MIP formulations, based on different strategies to model spanning trees and flow routing. A polyhedral study, complemented by an extensive array of computational experiments, have lead us to conclude that RDMFM is the most promising formulation. The LP relaxation of RDMFM yields the best lower bounds; and when implemented in

CPLEX, RDMFM tends to be the fastest model to solve our instances to optimality. The results of the computational experiments also emphasize the difficulty of solving the TE-MSTP problem to optimality, even for relatively small instances. This is not entirely unexpected, as the problem belongs to the $\mathcal{NP}$-hard complexity class of problems. The aforementioned difficulty is evidenced empirically not only by the lengthy computation times, but also by the often weak LP relaxations.

This lead us to develop two alternative methods to solve the TE-MSTP problem. The first was a B&C algorithm, based on the Benders' decomposition of RDMFM. Unfortunately, this method proved to be more inefficient than simply solving RDMFM with CPLEX - further on, we discuss potential reasons as for why. The second proposed method is a binary search algorithm, that iteratively solves "easier" sub-problems, in order to converge to a near-optimal solution for the TE-MSTP problem. This method engendered a comprehensive study of said sub-problems, the COCMST problem, in Chapter 3. This algorithm takes advantage of the fact that any feasible solution for the COCMST problem, is also feasible for the TE-MSTP problem, with a guaranteed worst-case edge utilization. Our computational experiments show that this method can be particularly useful to network service providers who seek, not necessarily the optimal solution, but one whose quality is provably "good". One interesting future research direction would be to improve on this method, namely by integrating it with fast heuristics that can quickly lower the upper bounds.

The difficulty mentioned above also lead us to look at another similar optimization problem, whose optimal designs may be akin to the ones of the TE-MSTP problem. This problem, the TE-MSTP-p problem, makes use of convex S-RDMFM functions to avoid heavy congested edges. In order to have a more complete understanding of how to model this type of cost functions, in Chapter 4, we first studied a more generic multicommodity flow problem, with single-path routing and S-RDMFM flow cost functions. We proposed, among other MIP formulations, a strengthened formulation, SM, whose linear relaxation always gives the optimal solution in the single commodity case of the PUMF problem. Furthermore, our computational experiments showed that SM also produces very tight LP bounds for the multi-commodity case.

Nonetheless, SM makes use of a larger number of variables than our other weaker models; as such, for some hard instances, the LPs of SM might be quite slower to solve, which also translates in larger MIP solving times. Therefore, as we had done before for the TE-MSTP problem, we resorted to a Benders' decomposition method in an attempt to reduce to LP solving time of SM, and thus have a more efficient method to solve the PUMF problem. However, once again, the corresponding B&C algorithm was shown not to be an efficient alternative method. The inefficiency of this method, in addition to the inefficiency of similar methods proposed for the TE-MSTP and TE-MSTP-p problems, begets a larger discussion regarding the use of Benders' decomposition methods in B&C frameworks.

As it had been suggested in Section 1.6, when we first introduced the Benders' decomposition method, the efficiency of the latter is very much problem dependent. Namely, an essential factor to make a Benders' decomposition work for a given problem,

is the ability of one to profit from the structure of the slave problem. Take, for instance, the uncapacitated facility problem, whose Benders' decomposition of the standard MIP model is a quintessential example of this [CP08][Mar12]. The standard formulation defines two binary variables: one that indicates whether or not a given facility should be open, and another one that assigns each costumer to a facility. Note that once we fix the set of facilities to be opened, the costumer allocation sub-problem becomes simple: we supply a costumer's full demand from the facility which provides it at a cheaper cost. Kipp Martin then shows how to translate this allocation to the solution of the corresponding dual slave problem, which finally yields the Benders' optimality cut. Furthermore, note that the allocation of a costumer is independent from the allocation of other costumers. Therefore, another important characteristic that is explored in this example is the decomposability of the slave problem. In general, this can result in significant speed-ups to the overall Benders' decomposition method solving time.

A key feature of this problem, that yields such a convenient structure, is the fact that it is uncapacitated. Conversely, the problems considered in this thesis, all have, in one sense or another, capacities on the traffic flows traversing the edges/arcs. This means that whenever we fix the design variables of these problems, the resulting sub-problems are still hard capacitated routing problems, that cannot be decomposed. Moreover, they are often infeasible problems, whose corresponding dual slave solution is not easy to infer, as in the case of the uncapacitated facility problem. These reasons can help explain the poor performance of the B&C algorithms proposed in this thesis, even when combined with state-of-the-art techniques such as the cut loop stabilization.

Alternatively, we can fix the routing of the commodities in the Benders' master problem. As discussed at the end of Section 2.5.1, and in Section 4.4.1, this can even translate in decomposable slave problems. However, in this situation, the number of integer variables in the master problems would not be significantly smaller than in the original models, as for all of them, the flow variables are always in larger quantity than any other type of variable.

The aforementioned issues point to the need of more suitable decomposition methods and/or techniques for capacitated network design problems. In Section 4.5.2, we discuss a situation where a "clever" implementation of the Benders' decomposition method allowed us to infer another strong model for the PUMF problem. To this effect, we use Benders' decomposition to project the strong valid inequalities of SM onto the space of variables of BM1. By observing the resulting Benders' cuts, we were able to identify classes of strong valid inequalities for BM1, that highly improve its LP bounds; we name SAM to the formulation resulting from the combination of both. Preliminary tests showed that for every instance in the proposed test sets, SAM obtains the same LP bound as SM. However they also reveal that, despite having less variables, SAM is slower to solve than SM; this is explained by the large quantity of valid inequalities. In the future, we would like to further study this model, and try to prove whether or not SAM and SM are equivalent formulations. Moreover, we would like to explore the use of row generation methods, that could profit from the strong valid inequalities of SAM, without the drawback of the large LPs. Namely, a valid option could be a Benders

decomposition of SAM, where only the aggregated x-variables are kept in the master.

Another interesting future research direction, would be to integrate SM and SAM in models used to solve other more complex multicommodity flow problems with single-path routing and piecewise linear costs. In Chapter 5, we combine the most promising formulations for the TE-MSTP and for the PUMF problems, to model the aforementioned TE-MSTP-p problem. However, other problems, such as the ones in [PF14b, PF14a], could benefit from doing the same.

Our computational experiments with instances of the TE-MSTP-p problem show that we are not able to capitalize on the good results of the PUMF problem - solving the TE-MSTP-p problems tends to be even harder than solving the TE-MSTP problem. The reasons for this differ for the two used test sets: for one test set, the better Gap_{LP} values obtained for the TE-MSTP-p problem are not sufficiently good to counter-weight the impact of the large LP solving time; for the other, the Gap_{LP} values are generally better for the TE-MSTP problem. This implies that the bottleneck for optimization problems for networks with the MSTP is not so much in the type of objective function, but rather on the design of juxtaposed multiple spanning trees, that share edge capacities or routing costs.

This important observation should been seen as a starting point for related future research: in order to be able to optimally solve problems dealing with the MSTP more efficiently, it is vital to develop models or methods that somehow circumvent this. In the meantime, we believe that the models and methods proposed in this thesis can provide a long-missing benchmark for evaluating the quality of other state-of-the-art heuristic methods.

References

[80298] IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Common Specifications Part 3: Media Access Control (MAC) Bridges. *ANSI/IEEE Std 802.1D, 1998 Edition*, pages i–355, 1998.

[80202] IEEE Standards for Local and Metropolitan Area Networks— Virtual Bridged Local Area Networks— Amendment 3: Multiple Spanning Trees. *IEEE Std 802.1s-2002 (Amendment to IEEE Std 802.1Q, 1998 Edition)*, pages 0_1–211, 2002.

[80206] IEEE Standard for Local and Metropolitan Area Networks Virtual Bridged Local Area Networks. *IEEE Std 802.1Q-2005 (Incorporates IEEE Std 802.1Q1998, IEEE Std 802.1u-2001, IEEE Std 802.1v-2001, and IEEE Std 802.1s-2002)*, pages 0_1–285, 2006.

[AdC03] Filipe Alvelos and JM Valério de Carvalho. Comparing branch-and-price algorithms for the unsplittable multicommodity flow problem. In *International Network Optimization Conference*, pages 7–12, 2003.

[BAM10] Theophilus Benson, Aditya Akella, and David A Maltz. Network traffic characteristics of data centers in the wild. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 267–280. ACM, 2010.

[Ben62] Jacques F Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252, 1962.

[BFGP13] Quentin Botton, Bernard Fortz, Luis Gouveia, and Michael Poss. Benders decomposition for the hop-constrained survivable network design problem. *INFORMS journal on computing*, 25(1):13–26, 2013.

[BG89] Anantharam Balakrishnan and Stephen C Graves. A composite algorithm for a concave-cost network flow problem. *Networks*, 19(2):175–202, 1989.

[BHV00] Cynthia Barnhart, Christopher A Hane, and Pamela H Vance. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326, 2000.

[BJN+98] Cynthia Barnhart, Ellis L Johnson, George L Nemhauser, Martin WP Savelsbergh, and Pamela H Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329, 1998.

[BMW89] Anantaram Balakrishnan, Thomas L Magnanti, and Richard T Wong. A dual-ascent procedure for large-scale uncapacitated network design. *Operations Research*, 37(5):716–740, 1989.

[BSL06] S. Balon, F. Skivée, and G. Leduc. How well do traffic engineering objective functions meet TE requirements? In *Proceedings of IFIP Networking 2006, Coimbra*, volume 3976. Springer LNCS, May 2006.

[CFM10] Ivan Contreras, Elena Fernández, and Alfredo Marín. Lagrangean bounds for the optimum communication spanning tree problem. *Top*, 18(1):140–157, 2010.

[CGM03] Keely L Croxton, Bernard Gendron, and Thomas L Magnanti. A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Management Science*, 49(9):1268–1273, 2003.

[CGM07] Keely L Croxton, Bernard Gendron, and Thomas L Magnanti. Variable disaggregation in network flow problems with piecewise linear costs. *Operations research*, 55(1):146–157, 2007.

[CJZ06] Wentao Chen, Depeng Jin, and Lieguang Zeng. Design of Multiple Spanning Trees for Traffic Engineering in Metro Ethernet. In *2006 International Conference on Communication Technology*, pages 1–4. IEEE, 2006.

[CKM05] Tibor Cinkler, András Kern, and István Moldován. Optimized QoS protection of Ethernet trees. In *Design of Reliable Communication Networks, 2005.(DRCN 2005). Proceedings. 5th International Workshop on*, pages 8–pp. IEEE, 2005.

[CMK+05] Tibor Cinkler, István Moldován, András Kern, Csaba Lukovszki, and Gyula Sallai. Optimizing QoS aware Ethernet spanning trees. *MSAN 2005*, 1:30–34, 2005.

[Con09] Ivan Contreras. *Network hub location: models, algorithms, and related problems.* PhD thesis, Universitat Politècnica de Catalunya, 2009.

[Coo71] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

[CP08] Uffe Gram Christensen and Anders Bjerg Pedersen. Lecture note on benders decomposition, 2008.

[Cre12] Ruth Cremer. Column generation for load balancing in multiple spanning tree routing. Master's thesis, RWTH Aachen, 2012.

[dSS07] A. de Sousa and G. Soares. Improving Load Balance and Minimizing Service Disruption on Ethernet Networks with IEEE 802.1S MSTP. In *Workshop on IP QoS and Traffic Control*, pages 25–35, 2007.

[DW60] George B Dantzig and Philip Wolfe. Decomposition principle for linear programs. *Operations research*, 8(1):101–111, 1960.

[FL03] Matteo Fischetti and Andrea Lodi. Local branching. *Mathematical programming*, 98(1-3):23–47, 2003.

[FLMH+13] Elena Fernández, Carlos Luna-Mota, Achim Hildenbrandt, Gerhard Reinelt, and Stefan Wiesberg. A flow formulation for the optimum communication spanning tree. *Electronic Notes in Discrete Mathematics*, 41:85–92, 2013.

[FLS15] Matteo Fischetti, Ivana Ljubic, and Markus Sinnl. Redesigning benders decomposition for large scale facility location. 2015.

[FSZ10] Matteo Fischetti, Domenico Salvagnin, and Arrigo Zanette. A note on the selection of benders cuts. *Mathematical Programming*, 124(1-2):175–182, 2010.

[FT00] Bernard Fortz and Mikkel Thorup. Internet traffic engineering by optimizing ospf weights. In *INFOCOM 2000. Nineteenth annual joint conference of the IEEE computer and communications societies. Proceedings. IEEE*, volume 2, pages 519–528. IEEE, 2000.

[FT04] Bernard Fortz and Mikkel Thorup. Increasing internet capacity using local search. *Computational Optimization and Applications*, 29(1):13–48, 2004.

[GG14] Bernard Gendron and Luis Gouveia. *Reformulations by Discretization for Piecewise Linear Integer Multicommodity Network Flow Problems.* 2014.

[GK77] M. Gerla and L. Kleinrock. Communication nets: stochastic message flow and delay. 25(1):48–60, 1977.

[GK04] Yashar Ganjali and Ali Keshavarzian. Load balancing in ad hoc networks: single-path routing vs. multi-path routing. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1120–1125. IEEE, 2004.

[GK06] E. Gourdin and O. Klopfenstein. Comparison of different qos-oriented objectives for multicommodity flow routing optimization. In *Proceedings of the International Conference on Telecommunications (ICT 2006)*, 2006.

[GPdS11] Luís Gouveia, Pedro Patrício, and Amaro de Sousa. Models for optimal survivable routing with a minimum number of hops: comparing disaggregated with aggregated models. *International Transactions in Operational Research*, 18(3):335–358, 2011.

[HDBF11] Trong-Viet Ho, Yves Deville, Olivier Bonaventure, and Pierre Francois. Traffic engineering for multiple spanning tree protocol in large data centers. In *23rd International Teletraffic Congress (ITC)*, pages 23–30. IEEE, 2011.

[HHLB11] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proc. of LION-5*, page 507523, 2011.

[Ho12] T-V. Ho. *Traffic engineering techniques for data center networks.* PhD thesis, Ecole polytechnique de Louvain, Université catholique de Louvain, 2012.

[Hu74] Te C Hu. Optimum communication spanning trees. *SIAM Journal on Computing*, 3(3):188–195, 1974.

[HZC06] X. He, M. Zhu, and Q. Chu. Traffic Engineering for Metro Ethernet Based on Multiple Spanning Trees. In *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06)*, pages 97–97. IEEE, 2006.

[I+05] Villy B Iversen et al. Teletraffic engineering handbook. *ITU-D SG*, 2:16, 2005.

[Inf07] Cisco Data Center Infrastructure. 2.5 design guide, cisco systems. *Inc, San Jose, CA*, 2007.

[JLK78] David S Johnson, Jan Karel Lenstra, and AHG Kan. The complexity of the network design problem. *Networks*, 8(4):279–285, 1978.

[Kar72] Richard M Karp. *Reducibility among combinatorial problems*. Springer, 1972.

[KCR08] Changhoon Kim, Matthew Caesar, and Jennifer Rexford. Floodless in Seattle: a scalable Ethernet architecture for large enterprises. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 3–14. ACM, 2008.

[Kle96] Jon M Kleinberg. Single-source unsplittable flow. In *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pages 68–77. IEEE, 1996.

[Kru56] Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[KSI04] Aleksandar Kolarov, Bhaskar Sengupta, and Atsushi Iwata. Design of multiple reverse spanning trees in next generation of ethernet-vpns. In *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, volume 3, pages 1390–1395. IEEE, 2004.

[Lev73] Leonid A Levin. Universal sequential search problems. *Problemy Peredachi Informatsii*, 9(3):115–116, 1973.

[LLL15] Steven SW Lee, Kuang-Yi Li, and Chieh-Ching Lin. Modeling and algorithm for multiple spanning tree provisioning in resilient and load balanced ethernet networks. *Mathematical Problems in Engineering*, 2015, 2015.

[LMPM14] Xuan Liu, Sudhir Mohanraj, Michał Pióro, and Deep Medhi. Multipath routing from a traffic engineering perspective: How beneficial is it? In *Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*, pages 143–154. IEEE, 2014.

[LOP07] Claude Lemaréchal, Adam Ouorou, and Georgios Petrou. A bundle-type algorithm for routing intelecommunication data networks. *Computational Optimization and Applications*, 44(3):385–409, 2007.

[LYD+03] Yujin Lim, Heeyeol Yu, Shirshanka Das, Scott Seongwook Lee, and Mario Gerla. QoS-aware multiple spanning tree mechanism over a bridged LAN environment. In *GLOBECOM '03. IEEE Global Telecommunications Conference (IEEE Cat. No.03CH37489)*, volume 6, pages 3068–3072. IEEE, 2003.

[Mar91] R. K. Martin. Using separation algorithms to generate mixed integer model reformulations. *Operations Research Letters*, 10(3):119–128, 1991.

[Mar12] Richard Kipp Martin. *Large scale linear and integer optimization: a unified approach*. Springer Science & Business Media, 2012.

[Med06] A. Meddeb. Multiple Spanning Tree Generation and Mapping Algorithms for Carrier Class Ethernets. In *IEEE Globecom 2006*, pages 1–5. IEEE, 2006.

[Med08] Aref Meddeb. Smart spanning tree bridging for carrier ethernets. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–5. IEEE, 2008.

[ML97] Herman Michiel and Koen Laevens. Teletraffic engineering in a broad-band era. *Proceedings of the IEEE*, 85(12):2007–2033, 1997.

[MSS09] G. Mirjalily, F. A. Sigari, and R. Saadat. Best Multiple Spanning Tree in Metro Ethernet Networks. In *2009 Second International Conference on Computer and Electrical Engineering*, pages 117–121. IEEE, 2009.

[MW95] Thomas L Magnanti and Laurence A Wolsey. Optimal trees. *Handbooks in operations research and management science*, 7:503–615, 1995.

[NMN01] Jaroslav Nešetřil, Eva Milková, and Helena Nešetřilová. Otakar Boruvka on minimum spanning tree problem translation of both the 1926 papers, comments, history. *Discrete Mathematics*, 233(1):3–36, 2001.

[OPTW07] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly. SNDlib 1.0–Survivable Network Design Library. In *Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium*, April 2007. http://sndlib.zib.de, extended version accepted in Networks, 2009.

[PF14a] D. Papadimitriou and B. Fortz. Methods for time-dependent combined network design and routing optimization. In *IEEE Global Communications Conference, GLOBECOM 2014, Austin, TX, USA, December 8-12, 2014*, pages 1303–1309, 2014.

[PF14b] D. Papadimitriou and B. Fortz. Time-dependent combined network design and routing optimization. In *IEEE International Conference on Communications, ICC 2014, Sydney, Australia, June 10-14, 2014*, pages 1124–1130, 2014.

[PM04] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufman, 2004.

[PNM+05] M. Padmaraj, S. Nair, M. Marchetti, G. Chiruvolu, M. Ali, and A. Ge. Metro Ethernet traffic engineering based on optimal multiple spanning trees˜. In *Second IFIP International Conference on Wireless and Optical Communications Networks, 2005. WOCN 2005.*, pages 568–572. IEEE, 2005.

[Pri57] Robert Clay Prim. Shortest connection networks and some generalizations. *Bell system technical journal*, 36(6):1389–1401, 1957.

[Rot08] Franz Rothlauf. *Design and Applications of Metaheuristics*. PhD thesis, Universität Mannheim, Habilitationsschrift, 2008.

[RVJ99] S Rüping, E Vonnahme, and J Jasperneite. Analysis of switched Ethernet networks with different topologies used in automation systems. In *Fieldbus Technology*, pages 351–358. Springer, 1999.

[SdSA+09] D. Santos, A. de Sousa, F. Alvelos, M. Dzida, M. Pióro, and M. Zagozdzon. Traffic Engineering of Multiple Spanning Tree Routing Networks : the Load Balancing Case. In *Next Generation Internet Networks, 2009. NGI '09*, pages 1–8, 2009.

[SdSA+10] D. Santos, A. de Sousa, F. Alvelos, M. Dzida, and M. Pióro. Optimization of link load balancing in multiple spanning tree routing networks. *Telecommunication Systems*, 48(1-2):109–124, 2010.

[Tys11] Jeff Tyson. How LAN switches work. *URL: http://www.howstuffworks.com/lan-switch.htm*, 2011.

[Wir97] Patricia E Wirth. The role of teletraffic modeling in the new communications paradigms. *Communications Magazine, IEEE*, 35(8):86–92, 1997.

[Wol98] Laurence A Wolsey. *Integer programming*, volume 42. Wiley New York, 1998.

Appendices

A Computational results for the TE-MSTP problem

Key:

- ins. : instance reference;
- LP-time (s) : time it takes to solve the LP relaxation;
- MIP-time (s) : time it takes to solve the (mixed-)integer program;
- B&C-time (s): time it takes to solve the B&C algorithm;
- IO-time (s): time it takes the in-out cut loop stabilization to be processed;
- B^* : best upper bound found by any of the models;
- B : best upper bound found by each model;
- B_{LP} : lower bound obtained by solving the LP relaxation;
- B_0 : lower bound obtained after the root node of the B&B tree has been processed;
- B_{end} : lower bound obtained after optimization is finished or interrupted;
- B_{io} : lower bound obtained at the end of the in-out cut loop stabilization;
- $Gap_{LP}(\%)$: gap between B^* and B_{LP};
- $Gap_0(\%)$: gap between B^* and B_0;
- $Gap_{end}(\%)$: gap between B and B_{end};
- $Gap_{io}(\%)$: gap between B^* and B_{io}.
- B&B Tree Nodes : number of nodes in the B&B enumeration tree;
- # User cuts: number of Benders' cuts generated as user cuts (see Section 2.5.2 for more details);
- # Lazy cons.: number of Benders' cuts generated as user cuts (see Section 2.5.2 for more details).

ins.	B^*	LP-time (s)			MIP-time (s)			B			$Gap_{LP}(\%)$			Gap_0 (%)			Gap_{end} (%)			B&B Tree Nodes		
		RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF
T_r^{1-1}	0.24	0.1	0.3	0.3	0	1	0	0.24	0.24	0.24	4.2	4.2	4.2	4.2	0.0	4.2	0.0	0.0	0.0	1E+02	6E+01	0E+00
T_r^{1-2}	0.24	0.1	0.5	0.3	2	15	2	0.24	0.24	0.24	12.5	12.5	12.5	12.5	12.5	12.5	0.0	0.0	0.0	4E+02	4E+03	3E+01
T_r^{1-3}	0.16	0.3	1.9	0.5	1225	-	74	0.16	0.21	0.16	54.8	44.7	41.8	44.3	44.7	41.7	0.0	51.1	0.0	2E+04	4E+04	1E+02
T_r^{1-4}	0.45	0.1	0.2	0.2	0	1	1	0.45	0.45	0.45	17.1	17.0	17.0	17.1	17.1	11.6	0.0	0.0	0.0	7E+01	2E+02	4E+01
T_r^{1-5}	0.20	0.1	0.6	0.4	0	1	1	0.20	0.20	0.20	5.0	5.0	5.0	5.0	5.0	5.0	0.0	0.0	0.0	8E+01	3E+02	8E+00
T_r^{2-1}	0.47	0.2	0.6	0.6	139	579	96	0.47	0.47	0.47	24.6	10.8	8.5	10.8	9.4	8.4	0.0	0.0	0.0	9E+03	3E+04	3E+03
T_r^{2-2}	0.65	0.1	1.0	0.5	2	186	21	0.65	0.65	0.65	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2E+02	2E+04	6E+02
T_r^{2-3}	0.34	0.3	2.8	0.9	-	-	-	0.34	0.48	0.34	12.4	12.4	12.4	12.4	12.4	12.4	11.8	38.0	11.8	3E+04	2E+04	9E+03
T_r^{2-4}	0.63	0.1	0.6	0.4	8	21	2	0.63	0.63	0.63	4.3	4.3	4.3	4.3	4.3	3.6	0.0	0.0	0.0	3E+03	7E+03	6E+02
T_r^{2-5}	0.88	0.1	1.3	0.5	0	1	1	0.88	0.88	0.88	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0E+00	2E+01	0E+00
T_r^{3-1}	0.19	0.2	1.5	1.0	30	1001	12	0.19	0.19	0.19	10.0	10.0	10.0	6.1	7.1	10.0	0.0	0.0	0.0	1E+03	4E+04	1E+02
T_r^{3-2}	0.32	0.1	0.4	0.4	2	-	2	0.32	0.32	0.32	3.1	3.1	3.1	3.1	3.1	3.1	0.0	3.1	0.0	2E+02	7E+05	7E+00
T_r^{3-3}	0.26	0.2	1.1	0.8	34	640	28	0.26	0.26	0.26	23.1	23.1	23.1	23.1	23.1	23.1	0.0	0.0	0.0	1E+03	2E+04	9E+01
T_r^{3-4}	0.18	0.3	1.2	1.6	252	1712	27	0.18	0.18	0.18	29.6	29.6	29.3	29.6	29.6	29.2	0.0	0.0	0.0	1E+04	8E+04	1E+02
T_r^{3-5}	0.21	0.2	1.1	1.4	37	951	113	0.21	0.21	0.21	35.1	34.6	32.5	35.1	35.1	32.3	0.0	0.0	0.0	1E+03	5E+04	4E+02
T_r^{4-1}	0.16	0.2	1.3	0.4	51	-	112	0.16	0.18	0.16	57.1	54.5	52.8	57.1	56.2	52.7	0.0	40.8	0.0	3E+03	9E+04	4E+02
T_r^{4-2}	0.25	0.2	1.5	0.4	59	2069	38	0.25	0.25	0.25	61.4	59.2	57.2	61.4	59.4	56.7	0.0	0.0	0.0	4E+03	5E+04	1E+02
T_r^{4-3}	0.22	0.1	1.3	0.4	1422	-	37	0.22	0.22	0.22	53.5	53.5	53.5	53.5	53.5	53.5	0.0	21.2	0.0	1E+05	2E+05	3E+02
T_r^{4-4}	0.24	0.1	1.4	0.4	90	2286	15	0.24	0.24	0.24	66.7	64.7	62.9	66.7	65.0	62.8	0.0	0.0	0.0	4E+03	6E+04	1E+01
T_r^{4-5}	0.10	0.2	1.1	0.4	13	-	23	0.10	0.10	0.10	62.4	61.4	58.9	60.0	58.9	46.7	0.0	42.7	0.0	1E+03	8E+04	2E+01
T_r^{5-1}	0.26	0.2	1.5	0.6	3299	-	954	0.26	0.32	0.26	53.1	43.3	40.7	42.8	43.1	40.6	0.0	48.0	0.0	1E+05	3E+04	4E+03
T_r^{5-2}	0.18	0.3	1.4	0.5	470	1225	53	0.18	0.18	0.18	33.3	33.3	33.3	33.3	33.3	33.3	0.0	0.0	0.0	2E+04	3E+04	1E+02
T_r^{5-3}	0.13	0.2	1.5	0.6	1023	-	52	0.13	0.16	0.13	48.2	48.2	46.5	48.2	48.2	46.4	0.0	38.5	0.0	3E+04	4E+04	7E+01
T_r^{5-4}	0.15	0.3	1.2	0.5	230	-	93	0.15	0.20	0.15	35.0	35.0	35.0	35.0	35.0	35.0	0.0	52.3	0.0	5E+03	3E+04	2E+02
T_r^{5-5}	0.18	0.3	1.7	0.5	1913	-	1816	0.18	0.21	0.18	48.7	38.6	34.4	38.5	38.3	34.2	0.0	43.5	0.0	3E+04	6E+04	5E+03

Table A.1: Test results for the TE-MSTP problem: $T^1_{rand} - T^5_{rand}$.

ins.	B^*	LP-time (s)			MIP-time (s)			B			Gap_{LP}(%)			Gap_0 (%)			Gap_{end} (%)			B&B Tree Nodes		
		RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF
T_r^{6-1}	0.25	0.3	2.3	2.3	189	-	586	0.25	0.25	0.25	52.6	52.6	52.6	52.6	52.6	52.6	0.0	36.8	0.0	5E+03	4E+04	9E+02
T_r^{6-2}	0.36	0.2	1.9	2.5	149	-	306	0.36	0.39	0.36	36.1	36.1	36.1	36.1	36.1	36.1	0.0	41.0	0.0	5E+03	3E+04	5E+02
T_r^{6-3}	0.45	0.2	1.9	2.5	114	3513	164	0.45	0.45	0.45	45.6	45.6	45.6	45.6	45.6	42.5	0.0	0.0	0.0	7E+03	5E+04	7E+01
T_r^{6-4}	0.23	0.2	2.9	4.8	-	-	380	0.23	0.24	0.23	62.0	58.5	57.2	62.0	62.0	56.5	20.6	57.9	0.0	2E+05	4E+04	1E+03
T_r^{6-5}	0.30	0.3	2.5	4.6	124	-	301	0.30	0.32	0.30	62.4	60.4	59.1	59.6	62.4	59.1	0.0	59.1	0.0	4E+03	4E+04	3E+02
T_r^{7-1}	0.54	0.3	3.9	1.4	-	-	159	0.54	0.54	0.54	15.6	15.6	15.6	15.6	14.8	15.6	4.9	14.8	0.0	3E+05	1E+05	2E+02
T_r^{7-2}	0.48	0.3	6.8	2.0	-	-	-	0.50	0.60	0.48	36.7	32.8	30.2	32.7	33.5	30.1	32.6	45.9	27.9	1E+04	2E+04	2E+03
T_r^{7-3}	0.33	0.3	7.0	1.7	-	-	-	0.33	0.36	0.34	41.6	35.5	32.1	35.8	36.2	32.0	33.3	39.9	32.0	2E+04	2E+04	2E+03
T_r^{7-4}	0.40	0.3	7.4	2.0	-	-	-	0.40	0.49	0.40	42.0	35.2	31.5	42.0	35.5	31.5	40.3	46.6	28.7	3E+04	2E+04	2E+03
T_r^{7-5}	0.52	0.3	12.3	1.7	-	-	-	0.52	0.64	0.56	28.8	28.0	22.9	28.8	28.8	22.9	21.7	42.2	24.5	3E+04	1E+04	3E+03
T_r^{8-1}	0.56	0.2	1.9	2.5	116	-	65	0.56	0.60	0.56	38.1	38.1	32.8	38.1	38.1	32.8	0.0	19.7	0.0	5E+03	6E+04	9E+01
T_r^{8-2}	0.56	0.2	1.9	2.6	107	647	93	0.56	0.56	0.56	25.0	25.0	25.0	25.0	25.0	25.0	0.0	0.0	0.0	2E+03	5E+03	1E+02
T_r^{8-3}	0.31	0.4	25.9	4.7	-	-	-	0.31	0.66	0.37	74.2	72.1	70.8	74.2	73.2	70.8	70.5	87.0	66.4	3E+04	7E+03	5E+02
T_r^{8-4}	0.52	0.2	1.5	1.2	2	8	9	0.52	0.52	0.52	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1E+02	1E+02	2E+01
T_r^{8-5}	0.54	0.4	15.8	2.6	-	-	-	0.54	0.60	0.61	70.9	70.7	67.8	68.4	70.9	66.2	66.1	73.8	51.6	1E+04	8E+03	6E+02
T_r^{9-1}	0.85	0.2	4.6	2.2	-	-	1039	0.87	0.93	0.85	0.6	0.6	0.6	0.6	0.6	0.6	2.2	9.1	0.0	6E+04	2E+04	4E+03
T_r^{9-2}	0.89	0.3	4.6	2.2	-	-	286	0.89	0.98	0.89	5.3	5.3	4.5	5.3	5.3	3.2	3.7	13.7	0.0	5E+04	3E+04	6E+02
T_r^{9-3}	0.75	0.3	3.3	2.5	499	-	788	0.75	0.79	0.75	11.6	10.5	7.5	9.6	11.2	7.2	0.0	16.0	0.0	3E+03	4E+04	2E+03
T_r^{9-4}	0.73	0.3	4.1	2.0	-	-	-	0.77	0.83	0.73	10.0	10.0	10.0	10.0	10.0	10.0	15.0	20.5	6.8	4E+04	1E+04	3E+03
T_r^{9-5}	0.76	0.2	3.8	2.3	582	-	61	0.76	0.76	0.76	7.9	6.9	5.7	6.7	7.9	4.4	0.0	7.9	0.0	3E+04	5E+04	4E+02
T_r^{10-1}	0.32	0.3	11.7	13.2	-	-	-	0.32	0.40	0.40	51.8	51.8	51.8	51.8	51.8	51.8	44.3	61.4	46.7	4E+04	1E+04	4E+02
T_r^{10-2}	0.26	0.5	12.1	47.2	-	-	-	0.26	0.46	0.28	61.5	61.5	61.5	61.5	61.5	61.5	60.4	78.3	42.9	4E+04	7E+03	9E+02
T_r^{10-3}	0.56	0.2	9.2	5.2	-	-	-	0.56	0.69	0.57	38.1	38.1	38.1	38.1	38.1	38.1	38.1	50.0	39.5	3E+04	3E+04	1E+03
T_r^{10-4}	0.29	0.8	17.7	19.4	-	-	-	0.29	0.59	0.36	67.0	61.9	59.2	62.6	62.6	59.1	26.4	80.5	62.0	2E+04	1E+04	5E+02
T_r^{10-5}	0.47	0.5	13.5	23.0	-	-	-	0.47	0.76	0.49	74.5	68.8	66.7	69.1	69.3	66.7	67.0	79.9	56.3	9E+03	6E+03	9E+02

Table A.2: Test results for the TE-MSTP problem: $T_{rand}^{6} - T_{rand}^{10}$.

ins.	B^*	LP-time (s)			MIP-time (s)			B			$Gap_{LP}(\%)$			Gap_0 (%)			Gap_{end} (%)			B&B Tree Nodes		
		RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF
T_{3tc}^{1-1}	0.16	0.2	0.6	0.3	0	0	0	0.16	0.16	0.16	3.2	3.2	3.2	3.2	3.2	3.2	0.0	0.0	0.0	5E+01	4E+01	5E+01
T_{3tc}^{1-2}	0.21	0.2	0.8	2.2	0	1	1	0.21	0.21	0.21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0E+00	1E+01	0E+00
T_{3tc}^{1-3}	0.18	0.2	0.8	2.3	1	2	1	0.18	0.18	0.18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2E+01	4E+01	3E+00
T_{3tc}^{1-4}	0.20	0.2	1.1	2.0	1	1	1	0.20	0.20	0.20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2E+02	0E+00	1E+01
T_{3tc}^{1-5}	0.14	0.2	0.9	2.1	1	7	2	0.14	0.14	0.14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3E+01	5E+02	1E+01
T_{3tc}^{2-1}	0.16	0.2	3.8	2.7	1	14	4	0.16	0.16	0.16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0E+00	4E+02	8E+01
T_{3tc}^{2-2}	0.35	0.2	1.2	0.5	0	1	1	0.35	0.35	0.35	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0E+00	0E+00	0E+00
T_{3tc}^{2-3}	0.31	0.2	1.0	0.7	0	0	1	0.31	0.31	0.31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0E+00	0E+00	0E+00
T_{3tc}^{2-4}	0.35	0.2	2.5	1.9	0	2	1	0.35	0.35	0.35	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0E+00	1E+01	0E+00
T_{3tc}^{2-5}	0.19	0.2	2.0	1.1	1	5	2	0.19	0.19	0.19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7E+01	7E+01	2E+02
T_{3tc}^{3-1}	0.24	0.3	12.1	20.5	1	130	25	0.24	0.24	0.24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0E+00	4E+03	0E+00
T_{3tc}^{3-2}	0.20	0.3	8.7	38.4	5	15	15	0.20	0.20	0.20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1E+02	3E+02	2E+01
T_{3tc}^{3-3}	0.22	0.3	16.4	18.7	3	82	62	0.22	0.22	0.22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8E+01	1E+03	3E+01
T_{3tc}^{3-4}	0.12	0.3	13.0	16.5	-	-	-	0.12	0.13	0.12	5.4	5.4	5.4	5.4	5.4	5.4	5.4	12.4	5.4	5E+04	3E+04	2E+04
T_{3tc}^{3-5}	0.22	0.2	5.8	12.7	5	21	11	0.22	0.22	0.22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2E+02	4E+02	9E+01
T_{3tc}^{4-1}	0.35	0.6	21.5	13.0	3	116	136	0.35	0.35	0.35	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2E+01	1E+03	5E+01
T_{3tc}^{4-2}	0.24	0.5	32.0	11.2	7	289	126	0.24	0.24	0.24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1E+02	4E+03	2E+02
T_{3tc}^{4-3}	0.45	0.4	6.4	6.0	1	6	4	0.45	0.45	0.45	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0E+00	0E+00	0E+00
T_{3tc}^{4-4}	0.54	0.3	5.1	3.1	1	5	5	0.54	0.54	0.54	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0E+00	1E+02	5E+01
T_{3tc}^{4-5}	0.31	0.3	11.3	30.6	9	3406	6	0.31	0.31	0.31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2E+02	8E+04	0E+00
T_{3tc}^{5-1}	0.17	0.3	45.7	42.3	795	83	88	0.17	0.17	0.17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2E+04	7E+02	4E+01
T_{3tc}^{5-2}	0.14	0.4	97.3	69.3	-	-	-	0.15	-	0.14	31.2	31.2	31.2	31.2	31.2	31.2	35.2	-	31.2	1E+04	9E+03	3E+03
T_{3tc}^{5-3}	0.18	0.3	87.6	69.5	110	-	312	0.18	0.23	0.18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	22.6	0.0	3E+03	6E+04	2E+01
T_{3tc}^{5-4}	0.21	0.3	50.6	60.3	-	-	-	0.21	0.24	0.21	14.5	14.5	14.5	14.5	14.5	14.5	7.4	27.7	16.6	5E+04	2E+04	2E+04
T_{3tc}^{5-5}	0.30	0.3	45.9	75.1	384	-	1586	0.30	0.32	0.30	34.9	34.9	34.9	34.9	34.9	34.9	0.0	40.5	0.0	4E+03	1E+04	1E+03
T_{3tc}^{6-1}	0.25	0.9	187.3	29.0	-	-	-	0.28	0.33	0.25	8.3	8.3	8.3	8.3	8.3	8.3	16.4	30.7	8.3	3E+04	1E+04	3E+03
T_{3tc}^{6-2}	0.60	0.6	173.1	133.9	3	230	288	0.60	0.60	0.60	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0E+00	1E+03	1E+01
T_{3tc}^{6-3}	0.38	0.7	138.4	108.2	6	1483	310	0.38	0.38	0.38	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3E+01	1E+04	1E+01
T_{3tc}^{6-4}	0.41	0.6	97.1	127.8	6	-	153	0.41	0.45	0.41	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.1	0.0	5E+01	2E+04	2E+01
T_{3tc}^{6-5}	0.28	0.7	217.9	105.6	-	-	1227	0.28	0.34	0.28	0.0	0.0	0.0	0.0	0.0	0.0	1.9	17.3	0.0	1E+04	7E+03	2E+02

Table A.3: Test results for the TE-MSTP problem: T_{3tc}.

ins.	B&C-time	IO-time	B	$Gap_{io}(\%)$	B&C Tree Nodes	# User cuts	# Lazy const.
T_r^{1-1}	80	1	0.24	8	2	1208	23
T_r^{1-2}	-	16	0.24	19	0	7361	171
T_r^{1-3}	-	2	0.24	100	0	4223	411
T_r^{1-4}	6	1	0.45	17	72	202	7
T_r^{1-5}	2	1	0.20	5	10	31	12
T_r^{2-1}	-	178	0.57	16	0	4972	465
T_r^{2-2}	74	29	0.65	0	0	449	213
T_r^{2-3}	-	3	0.46	100	0	3417	399
T_r^{2-4}	-	2	0.63	4	0	6281	276
T_r^{2-5}	2	1	0.88	0	0	12	6
T_r^{3-1}	-	1	0.31	100	1	12105	14
T_r^{3-2}	1383	2	0.32	7	12	3867	48
T_r^{3-3}	-	1	0.49	100	1	9809	13
T_r^{3-4}	-	1	0.23	100	0	4677	279
T_r^{3-5}	-	1	0.30	100	0	4403	342
T_r^{4-1}	-	1	0.20	100	0	4726	351
T_r^{4-2}	-	1	0.44	100	1	8706	25
T_r^{4-3}	-	1	0.22	100	0	4581	349
T_r^{4-4}	-	1	0.32	100	0	4383	320
T_r^{4-5}	-	1	0.10	100	0	5054	131
T_r^{5-1}	-	2	0.40	100	0	3980	518
T_r^{5-2}	-	1	0.27	100	0	4198	438
T_r^{5-3}	-	1	0.18	100	0	4336	331
T_r^{5-4}	-	1	0.19	100	1	5609	85
T_r^{5-5}	-	2	0.35	100	1	7087	19
T_r^{6-1}	-	2	0.29	100	0	3925	452
T_r^{6-2}	-	2	0.40	100	0	3651	366
T_r^{6-3}	-	1	0.45	100	0	4191	100
T_r^{6-4}	-	1	0.24	100	0	4218	378
T_r^{6-5}	-	1	0.35	100	0	4249	448
T_r^{7-1}	-	4	0.54	100	0	2780	135
T_r^{7-2}	-	3	0.93	100	1	5464	27
T_r^{7-3}	-	4	0.57	100	1	8113	14
T_r^{7-4}	-	3	0.60	100	0	3224	461
T_r^{7-5}	-	4	0.84	100	0	3164	437
T_r^{8-1}	-	2	0.72	100	0	3686	212
T_r^{8-2}	-	2	0.61	100	0	3666	265
T_r^{8-3}	-	5	0.39	100	1	2976	48
T_r^{8-4}	-	5	0.52	5	0	5728	148
T_r^{8-5}	-	5	0.65	100	0	2759	569
T_r^{9-1}	-	716	1.10	21	0	3022	345
T_r^{9-2}	-	5	1.31	100	0	2196	197
T_r^{9-3}	-	235	0.83	23	0	3532	341
T_r^{9-4}	-	5	1.00	100	0	2846	287
T_r^{9-5}	-	4	0.86	100	0	2699	150
T_r^{10-1}	-	4	0.36	100	0	3216	553
T_r^{10-2}	-	4	0.35	100	0	3342	517
T_r^{10-3}	-	4	0.68	100	0	2954	445
T_r^{10-4}	-	4	0.50	100	1	8752	77
T_r^{10-5}	-	4	0.55	100	0	3114	518

Table A.4: Test results for the B&C algorithm: T_{rand}.

ins.	B&C-time	IO-time	B	$Gap_{io}(\%)$	B&C Tree Nodes	# User cuts	# Lazy const.
T_{3tc}^{1-1}	6	1	0.16	3	40	3	5
T_{3tc}^{1-2}	1	1	0.21	0	0	0	1
T_{3tc}^{1-3}	4	2	0.18	0	0	5	3
T_{3tc}^{1-4}	8	2	0.20	0	6	19	21
T_{3tc}^{1-5}	20	5	0.14	0	28	90	19
T_{3tc}^{2-1}	-	248	0.17	24	0	3699	249
T_{3tc}^{2-2}	2	2	0.35	0	0	0	1
T_{3tc}^{2-3}	10	6	0.31	0	10	4	14
T_{3tc}^{2-4}	4	2	0.35	0	0	2	1
T_{3tc}^{2-5}	57	10	0.19	0	16	169	46
T_{3tc}^{3-1}	18	9	0.24	0	0	9	9
T_{3tc}^{3-2}	465	9	0.20	100	0	631	63
T_{3tc}^{3-3}	-	9	0.22	100	0	3354	580
T_{3tc}^{3-4}	-	8	0.15	100	0	2092	169
T_{3tc}^{3-5}	98	16	0.22	0	0	203	94
T_{3tc}^{4-1}	-	41	0.36	53	0	1946	122
T_{3tc}^{4-2}	-	23	0.25	100	0	1971	73
T_{3tc}^{4-3}	15	11	0.45	0	0	0	1
T_{3tc}^{4-4}	66	20	0.54	0	17	29	15
T_{3tc}^{4-5}	71	26	0.31	0	0	42	21
T_{3tc}^{5-1}	148	14	0.17	100	3	164	65
T_{3tc}^{5-2}	-	17	0.20	100	0	2000	154
T_{3tc}^{5-3}	802	17	0.18	100	0	699	143
T_{3tc}^{5-4}	-	14	0.24	100	0	2573	381
T_{3tc}^{5-5}	-	12	0.32	100	0	2304	191
T_{3tc}^{6-1}	-	43	0.32	100	0	1258	61
T_{3tc}^{6-2}	61	41	0.60	0	0	1	7
T_{3tc}^{6-3}	87	42	0.38	0	0	12	13
T_{3tc}^{6-4}	-	37	0.63	100	16	1767	8
T_{3tc}^{6-5}	-	49	0.41	100	0	921	75

Table A.5: Test results for the B&C algorithm: T_{3tc}.

B Computational results for the COCMST problem

Key:

- ins. : instance reference;
- LP-time (s) : time it takes to solve the LP relaxation;
- MIP-time (s) : time it takes to solve the (mixed-)integer program;
- Feas.-time (s) : time it takes to find a first feasible solution;
- B^* : best upper bound found by any of the models;
- B : best upper bound found by each model;
- B_{LP} : lower bound obtained by solving the LP relaxation;
- B_0 : lower bound obtained after the root node of the B&B tree has been processed;
- U^* : worst-case edge utilization obtained by solving the COCMST problem to optimality;
- U^1 : worst-case edge utilization corresponding to the first feasible solution found for the COCMST problem;
- $Gap_{LP}(\%)$: gap between B^* and B_{LP};
- $Gap_0(\%)$: gap between B^* and B_0;
- $Gap_U^*(\%)$: gap between B^* and U^*;
- $Gap_U^1(\%)$: gap between B^* and U^1;
- B&B Tree Nodes : number of nodes in the B&B enumeration tree.

ins.	B^*	LP-time (s)			MIP-time (s)			B			Gap_{LP}(%)			Gap_0 (%)			B&B Tree Nodes			Gap^*_U (%)			Feas.-time (s)			Gap^1_U (%)		
		RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF
T_r^{1-1}	130	0.1	0.2	0.2	0	2	0	130	130	130	16.7	4.5	3.7	1.7	2.6	0.0	7E+00	2E+01	0E+00	14.3	14.3	14.3	0	0	0	14.3	14.3	5.3
T_r^{1-2}	140	0.1	0.2	0.2	1	1	1	140	140	140	11.5	1.8	1.7	2.4	0.6	0.6	2E+01	5E+00	0E+00	14.3	14.3	14.3	0	1	0	14.3	14.3	14.3
T_r^{1-3}	155	0.1	0.5	0.6	147	697	43	155	155	155	32.9	13.2	11.5	11.8	11.6	10.8	8E+03	1E+04	2E+02	14.3	14.3	14.3	16	402	27	15.8	14.3	15.8
T_r^{1-4}	182	0.1	0.2	0.2	0	1	0	182	182	182	15.6	3.4	3.4	3.4	0.9	1.4	4E+01	0E+00	0E+00	12.8	12.8	12.8	0	0	0	5.6	5.6	16.0
T_r^{1-5}	84	0.1	0.2	0.2	0	0	0	84	84	84	14.3	0.8	0.8	0.8	0.0	0.0	0E+00	0E+00	0E+00	0.0	0.0	0.0	0	0	0	16.7	16.7	16.7
T_r^{2-1}	364	0.1	0.4	0.5	3	13	2	364	364	364	22.4	2.3	0.7	1.8	1.0	0.2	3E+02	2E+02	0E+00	16.7	16.7	16.7	0	11	0	16.7	14.6	13.6
T_r^{2-2}	368	0.1	0.3	0.3	3	9	5	368	368	368	19.5	3.7	0.7	3.2	1.4	0.5	6E+02	2E+02	4E+01	14.0	14.0	14.0	0	2	0	15.5	16.2	14.0
T_r^{2-3}	384	0.2	1.0	1.1	1136	188	14	384	384	384	29.7	3.3	0.7	3.7	1.9	0.5	2E+04	2E+03	1E+01	15.0	15.0	15.0	272	125	6	15.0	15.0	15.0
T_r^{2-4}	393	0.1	0.4	0.3	1	1	1	393	393	393	15.9	1.6	0.4	0.7	0.4	0.0	1E+02	0E+00	0E+00	16.1	16.1	16.1	0	1	1	16.1	16.1	9.6
T_r^{2-5}	352	0.1	0.3	0.3	1	1	1	352	352	352	15.8	1.4	0.0	1.0	0.4	0.0	3E+01	0E+00	0E+00	0.0	0.0	0.0	0	1	0	15.4	0.0	12.0
T_r^{3-1}	172	0.1	0.4	0.5	9	29	8	172	172	172	24.4	5.8	4.8	6.5	4.9	4.2	1E+03	1E+03	6E+01	0.0	12.5	0.0	0	23	2	15.2	6.7	15.2
T_r^{3-2}	175	0.1	0.3	0.4	2	2	1	175	175	175	22.9	2.1	1.2	2.7	1.3	0.9	1E+02	0E+00	0E+00	0.0	0.0	11.1	0	1	0	8.6	4.0	11.1
T_r^{3-3}	221	0.1	0.4	0.5	21	58	17	221	221	221	28.5	9.4	6.2	9.0	6.6	5.8	2E+03	2E+03	1E+02	7.1	15.2	13.3	5	26	2	16.1	16.1	16.1
T_r^{3-4}	153	0.1	0.3	0.5	25	24	21	153	153	153	24.2	6.3	5.9	6.9	5.3	4.0	5E+03	7E+02	2E+02	15.6	15.6	10.0	8	18	3	15.6	15.6	14.3
T_r^{3-5}	184	0.1	0.3	0.3	5	15	15	184	184	184	23.9	5.9	5.7	6.0	5.2	4.1	9E+02	6E+02	9E+01	11.1	11.1	11.1	0	2	2	11.1	14.7	11.1
T_r^{4-1}	115	0.1	0.3	0.6	54	70	28	115	115	115	34.8	10.6	5.1	9.8	7.9	4.1	6E+03	3E+03	4E+02	11.1	11.1	11.1	1	33	4	14.3	14.3	14.3
T_r^{4-2}	177	0.1	0.3	0.4	20	41	9	177	177	177	38.4	11.6	9.5	11.9	10.8	7.9	1E+03	2E+03	7E+01	15.6	15.6	15.6	1	25	1	9.5	15.6	15.6
T_r^{4-3}	110	0.1	0.3	0.3	1	1	1	110	110	110	23.6	0.6	0.0	0.0	0.6	0.0	0E+00	0E+00	0E+00	0.0	0.0	0.0	0	0	1	15.4	0.0	0.0
T_r^{4-4}	143	0.1	0.3	0.4	16	65	18	143	143	143	40.6	18.0	17.8	18.2	17.2	16.2	2E+03	4E+03	1E+02	0.0	0.0	7.7	5	31	1	14.3	0.0	14.3
T_r^{4-5}	60	0.1	0.3	0.3	3	22	9	60	60	60	28.3	10.9	10.0	10.0	9.4	5.8	7E+02	9E+02	5E+01	0.0	0.0	0.0	0	19	0	16.7	0.0	16.7
T_r^{5-1}	231	0.1	0.4	0.6	6	19	1	231	231	231	29.4	1.6	0.3	1.6	0.9	0.0	2E+02	3E+02	0E+00	13.3	13.3	13.3	4	19	1	13.3	13.3	13.3
T_r^{5-2}	157	0.1	0.6	0.5	11	25	18	157	157	157	31.2	5.7	3.7	5.7	4.4	2.3	5E+02	4E+02	1E+02	0.0	0.0	0.0	0	21	3	15.6	15.6	10.0
T_r^{5-3}	124	0.1	0.5	0.5	71	653	38	124	124	124	33.9	16.8	15.6	16.0	16.0	13.9	7E+03	1E+04	2E+02	13.3	13.3	13.3	19	497	22	13.3	13.3	13.3
T_r^{5-4}	121	0.1	0.5	0.5	7	7	4	121	121	121	27.3	4.5	3.2	3.1	2.8	1.6	3E+02	3E+01	0E+00	15.4	15.4	15.4	6	7	2	15.4	15.4	15.4
T_r^{5-5}	205	0.1	0.4	0.6	184	313	71	205	205	205	32.2	6.7	4.3	6.1	5.2	3.7	1E+04	8E+03	7E+02	15.6	15.6	15.6	41	188	12	15.6	15.6	14.3

Table B.1: Test results for the COCMST(0.2^ϵ) problem: $T^1_{rand} - T^5_{rand}$.

ins.	B^*	LP-time (s)			MIP-time (s)			B			$Gap_{LP}(\%)$			Gap_0 (%)			B&B Tree Nodes			Gap_U^* (%)			Feas.-time (s)			Gap_U^1 (%)		
		RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF
T_r^{6-1}	192	0.2	0.7	1.0	75	620	55	192	192	192	29.2	10.5	4.6	8.1	6.4	4.4	5E+03	1E+04	4E+02	13.6	13.6	13.6	1	288	0	9.5	9.5	12.6
T_r^{6-2}	266	0.2	0.9	0.9	21	265	52	266	266	266	28.9	10.9	9.1	11.7	9.7	6.8	2E+03	3E+03	2E+02	5.3	7.7	7.7	1	73	4	15.6	7.7	5.3
T_r^{6-3}	222	0.1	0.8	0.8	10	99	26	222	222	222	27.0	7.9	6.1	7.5	7.2	5.2	9E+02	1E+03	3E+01	16.0	16.0	16.0	0	73	0	5.6	16.0	5.6
T_r^{6-4}	124	0.1	0.6	0.8	2	13	6	124	124	124	21.6	3.6	1.2	3.1	1.4	0.0	5E+01	7E+01	0E+00	12.8	12.8	12.8	0	3	0	12.8	0.0	15.0
T_r^{6-5}	164	0.1	0.7	0.8	2	10	2	164	164	164	21.3	1.2	1.1	1.6	0.5	0.0	3E+01	1E+01	0E+00	16.7	16.7	16.7	0	10	0	13.5	16.7	16.7
T_r^{7-1}	389	0.2	1.6	2.1	268	130	21	389	389	389	26.2	4.2	0.8	3.8	1.9	0.6	1E+04	7E+02	6E+01	12.9	12.9	8.0	0	20	0	15.6	13.8	12.0
T_r^{7-2}	528	0.2	1.7	1.8	310	27	21	528	528	528	24.1	2.1	0.4	2.7	0.8	0.3	9E+03	8E+01	2E+01	11.1	11.1	11.1	42	17	7	15.8	11.1	11.1
T_r^{7-3}	399	0.2	1.4	1.9	915	2478	45	399	399	399	26.6	4.2	0.7	4.2	2.1	0.6	3E+04	2E+04	1E+02	10.8	10.8	10.8	14	973	9	13.2	15.4	14.7
T_r^{7-4}	419	0.2	1.5	2.0	572	-	372	419	425	419	27.7	4.8	1.6	5.0	2.7	1.5	1E+04	2E+04	7E+02	16.7	16.7	16.7	4	33	211	16.7	16.7	16.7
T_r^{7-5}	584	0.2	1.8	1.8	-	-	-	667	-	584	32.7	9.9	5.4	9.0	7.9	5.2	3E+04	1E+04	6E+03	16.1	-	13.3	2703	-	517	16.1	-	13.3
T_r^{8-1}	458	0.1	0.6	1.3	65	714	45	458	458	458	28.0	10.1	3.6	8.7	6.1	3.2	4E+03	1E+04	3E+02	15.2	15.2	15.2	10	16	7	8.7	9.7	12.5
T_r^{8-2}	444	0.1	0.7	1.1	38	355	57	444	444	444	22.5	6.8	4.7	6.3	5.1	4.4	2E+03	3E+03	1E+02	16.0	16.0	16.0	9	83	44	16.0	16.0	16.0
T_r^{8-3}	225	0.2	3.9	3.2	7	8	4	225	225	225	27.6	0.1	0.0	0.9	0.1	0.0	1E+02	0E+00	0E+00	7.0	8.8	3.1	2	8	5	13.9	8.8	3.1
T_r^{8-4}	305	0.1	0.6	1.3	2	30	10	305	305	305	22.3	6.7	5.3	6.0	5.6	3.9	1E+02	2E+02	2E+01	0.0	0.0	0.0	0	8	0	7.1	3.7	7.1
T_r^{8-5}	385	0.2	3.7	3.6	2004	630	121	385	385	385	29.6	5.1	0.4	5.5	3.6	0.3	4E+04	3E+03	9E+01	13.8	13.8	13.8	5	102	0	13.8	3.6	15.6
T_r^{9-1}	946	0.2	1.7	3.1	536	-	117	946	-	946	24.3	6.4	2.1	5.8	4.4	1.7	1E+04	1E+04	2E+02	16.3	-	16.3	23	-	58	14.7	-	16.3
T_r^{9-2}	776	0.2	1.7	2.5	653	501	142	776	776	776	22.0	3.5	1.3	4.0	2.2	1.2	2E+04	3E+03	3E+02	15.2	15.2	15.2	3	26	50	15.7	13.0	15.2
T_r^{9-3}	652	0.2	1.0	1.7	5	115	51	652	652	652	12.9	0.8	0.5	0.6	0.6	0.3	2E+02	5E+02	5E+01	16.1	16.1	15.2	1	10	8	16.1	9.7	15.2
T_r^{9-4}	814	0.2	1.8	2.8	3137	-	882	814	816	814	22.5	7.2	2.1	7.0	4.6	1.7	9E+04	1E+04	3E+03	15.1	15.1	15.1	34	503	56	11.7	14.1	11.0
T_r^{9-5}	607	0.2	1.2	2.1	12	69	7	607	607	607	16.8	1.4	0.5	0.8	0.5	0.3	6E+02	2E+02	0E+00	16.2	16.2	16.2	0	62	5	11.6	5.0	9.5
T_r^{10-1}	286	0.2	2.9	4.9	175	2403	46	286	286	286	30.1	5.2	1.4	5.7	3.3	1.3	5E+03	3E+03	3E+01	13.5	13.5	13.5	4	682	17	14.3	8.6	13.5
T_r^{10-2}	226	0.2	3.0	3.1	468	1160	192	226	226	226	27.0	6.0	2.2	6.0	4.6	1.8	1E+04	3E+03	1E+02	11.4	11.4	10.3	6	217	16	16.1	13.3	15.2
T_r^{10-3}	472	0.2	2.6	2.9	1454	-	648	472	-	472	29.0	8.2	2.3	6.7	6.4	2.2	4E+04	1E+04	1E+03	10.6	-	10.6	0	-	17	12.5	-	16.0
T_r^{10-4}	287	0.2	2.6	3.0	-	-	388	287	323	287	32.1	7.7	2.6	8.8	5.0	2.4	5E+04	6E+03	4E+02	9.4	14.7	9.4	1150	2155	366	9.4	14.7	9.4
T_r^{10-5}	376	0.2	2.8	6.4	450	2086	602	376	376	376	27.9	7.4	5.1	7.3	6.7	5.1	1E+04	1E+04	1E+03	16.1	16.1	16.1	22	188	201	9.6	11.9	16.1

Table B.2: Test results for the COCMST(0.2^ϵ) problem: $T_{rand}^6 - T_{rand}^{10}$.

ins.	B^*	LP-time (s)			MIP-time (s)			B			Gap_{LP}(%)			Gap_0 (%)			B&B Tree Nodes			Gap_U^* (%)			Feas.-time (s)			Gap_U^1 (%)		
		RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF
T_{3tc}^{1-1}	14454	0.1	0.4	0.2	0	0	0	14454	14454	14454	1.3	0.0	0.0	0.0	0.0	0.0	0E+00	0E+00	0E+00	9.1	9.1	9.1	0	0	0	12.1	3.8	9.0
T_{3tc}^{1-2}	17270	0.2	0.6	0.4	0	4	1	17270	17270	17270	6.6	4.6	3.8	4.1	3.8	0.7	8E+00	2E+01	0E+00	0.8	0.8	0.8	0	1	0	0.3	-0.8	0.3
T_{3tc}^{1-3}	14906	0.2	0.5	0.5	0	2	1	14906	14906	14906	7.3	0.2	0.2	0.1	0.1	0.1	0E+00	0E+00	0E+00	13.6	13.6	13.6	0	1	0	13.6	10.6	13.0
T_{3tc}^{1-4}	19228	0.1	0.5	0.5	0	2	1	19228	19228	19228	12.2	0.2	0.1	0.0	0.1	0.1	0E+00	0E+00	0E+00	16.2	16.2	16.2	0	1	0	2.0	16.2	14.8
T_{3tc}^{1-5}	15730	0.1	0.6	0.5	1	2	1	15730	15730	15730	14.4	0.0	0.0	0.0	0.0	0.0	0E+00	0E+00	0E+00	4.2	4.2	4.2	0	1	0	13.1	12.2	14.4
T_{3tc}^{2-1}	19432	0.2	1.3	1.2	24	51	2	19432	19432	19432	8.7	0.8	0.4	0.9	0.5	0.0	3E+03	4E+02	0E+00	16.4	16.4	16.4	0	7	2	13.3	7.0	15.0
T_{3tc}^{2-2}	24572	0.2	0.6	0.4	0	1	1	24572	24572	24572	2.7	0.0	0.0	1.1	0.0	0.0	0E+00	0E+00	0E+00	-0.2	-0.2	-0.2	0	1	0	-0.2	-0.2	-0.2
T_{3tc}^{2-3}	30284	0.2	0.8	0.5	0	1	1	30284	30284	30284	4.3	0.0	0.0	1.8	0.0	0.0	0E+00	0E+00	0E+00	8.3	8.3	8.3	0	1	0	14.4	8.3	7.6
T_{3tc}^{2-4}	29476	0.2	0.7	0.5	0	2	1	29476	29476	29476	11.2	0.0	0.0	1.0	0.0	0.0	0E+00	0E+00	0E+00	-0.4	-0.4	-0.4	0	2	1	8.6	-0.4	16.2
T_{3tc}^{2-5}	22962	0.2	0.9	0.7	1	5	1	22962	22962	22962	12.2	0.2	0.1	0.1	0.0	0.0	4E+01	0E+00	0E+00	16.0	16.0	16.0	0	4	1	11.2	13.7	9.2
T_{3tc}^{3-1}	28350	0.2	4.5	13.0	5	203	9	28350	28350	28350	14.3	0.1	0.1	0.1	0.1	0.1	9E+01	2E+02	0E+00	15.0	15.0	15.0	0	168	1	15.5	16.5	17.6
T_{3tc}^{3-2}	26718	0.2	2.3	3.6	29	246	63	26718	26718	26718	12.0	0.7	0.4	1.3	0.5	0.4	2E+03	6E+02	2E+02	13.9	13.9	13.9	0	62	6	14.6	9.9	12.6
T_{3tc}^{3-3}	29282	0.2	4.4	5.2	84	1029	49	29282	29282	29282	15.3	1.1	0.4	1.5	0.6	0.3	2E+03	2E+03	2E+01	8.4	8.4	8.4	0	563	19	14.9	12.6	8.4
T_{3tc}^{3-4}	18748	0.2	2.9	6.8	-	-	1859	18748	18916	18748	14.7	5.4	2.7	5.1	3.9	2.6	1E+05	1E+04	1E+04	14.5	14.7	14.5	3	653	152	16.9	17.1	17.4
T_{3tc}^{3-5}	28908	0.2	1.3	1.9	6	62	10	28908	28908	28908	7.8	1.2	0.6	0.8	0.8	0.4	4E+02	1E+02	2E+01	17.3	17.3	17.3	0	11	1	17.1	17.3	17.3
T_{3tc}^{4-1}	46862	0.3	3.7	13.4	1018	1951	560	46862	46862	46862	15.1	0.7	0.3	0.5	0.5	0.3	2E+04	3E+03	4E+02	12.4	12.4	12.4	0	724	15	6.0	11.1	14.0
T_{3tc}^{4-2}	35502	0.3	3.2	11.3	-	-	2224	35502	35740	35502	11.8	2.5	0.8	2.2	1.6	0.7	9E+04	5E+03	5E+03	17.1	16.0	17.1	0	1154	14	17.1	18.4	15.2
T_{3tc}^{4-3}	38904	0.3	2.1	1.8	1	6	4	38904	38904	38904	4.9	0.0	0.0	0.2	0.0	0.0	0E+00	0E+00	0E+00	14.5	14.5	14.5	0	6	1	12.6	14.5	16.0
T_{3tc}^{4-4}	63120	0.2	2.0	2.0	4	28	8	63120	63120	63120	4.2	0.4	0.3	0.4	0.3	0.2	1E+02	4E+01	2E+01	16.4	16.4	16.4	0	9	1	16.9	16.2	10.8
T_{3tc}^{4-5}	39758	0.2	5.4	4.7	90	2630	41	39758	39758	39758	7.7	1.9	0.6	1.7	1.1	0.5	5E+03	8E+03	2E+02	16.9	16.9	16.9	0	219	5	14.9	15.3	16.7
T_{3tc}^{5-1}	21530	0.2	2.7	14.5	5	213	13	21530	21530	21530	5.8	0.8	0.2	0.7	0.7	0.1	4E+01	5E+01	0E+00	11.0	11.0	11.0	0	140	1	14.9	11.0	11.8
T_{3tc}^{5-2}	19774	0.3	4.8	32.1	16	-	117	19774	20430	19774	14.7	0.6	0.6	0.6	0.5	0.4	2E+02	2E+02	3E+01	17.9	18.1	17.9	0	3028	17	14.0	20.5	20.4
T_{3tc}^{5-3}	24894	0.3	5.9	63.0	58	-	280	24894	-	24894	19.3	0.3	0.2	0.4	0.3	0.2	6E+02	3E+02	3E+01	14.9	-	14.9	0	-	34	16.1	-	15.8
T_{3tc}^{5-4}	30022	0.2	6.1	17.7	118	-	1295	30022	30130	30022	16.9	3.0	3.0	3.0	3.0	2.9	5E+03	9E+02	9E+02	9.5	5.3	9.5	2	2861	22	10.8	13.3	3.0
T_{3tc}^{5-5}	40398	0.2	3.6	25.2	344	2429	466	40398	40398	40398	18.5	2.3	2.0	3.1	2.2	1.9	8E+03	1E+03	4E+02	8.4	8.4	8.4	0	592	11	14.3	13.9	15.1
T_{3tc}^{6-1}	-	0.3	41.1	31.0	-	-	-	48930	-	49538	-	-	-	-	-	-	3E+04	5E+02	3E+03	16.7	-	17.0	472	-	390	17.2	-	17.0
T_{3tc}^{6-2}	65180	0.4	38.3	114.4	9	108	36	65180	65180	65180	13.7	0.1	0.0	0.0	0.1	0.0	3E+00	0E+00	0E+00	16.2	16.2	16.2	0	93	2	3.4	16.2	15.1
T_{3tc}^{6-3}	50674	0.4	43.8	98.6	110	669	132	50674	50674	50674	14.4	0.3	0.3	0.3	0.3	0.2	1E+03	2E+01	1E+01	4.8	4.8	4.8	0	105	2	2.3	4.8	8.0
T_{3tc}^{6-4}	68456	0.3	5.5	73.0	471	-	204	68456	68570	68456	11.6	0.9	0.3	0.8	0.8	0.3	4E+03	2E+03	1E+02	16.5	16.5	16.5	9	520	27	15.3	16.8	15.7
T_{3tc}^{6-5}	-	0.3	35.1	79.8	-	-	-	59306	-	57366	-	-	-	-	-	-	2E+04	1E+02	6E+02	15.6	-	13.5	905	-	2621	15.6	-	14.1

Table B.3: Test results for the COCMST(0.2^ϵ) problem: T_{3tc}.

ins.	B^*	LP-time (s)			MIP-time (s)			B			Gap_{LP}(%)			Gap_0 (%)			B&B Tree Nodes			Gap_U^* (%)			Feas.-time (s)			Gap_U^1 (%)		
		RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF
T_r^{1-1}	131	0.1	0.2	0.2	1	2	0	131	131	131	17.4	3.9	2.8	1.0	1.5	1.2	0E+00	4E+00	0E+00	0.0	0.0	0.0	0	1	0	0.0	0.0	0.0
T_r^{1-2}	157	0.1	0.3	0.2	1	5	1	157	157	157	21.1	11.6	11.0	11.9	9.6	3.8	2E+02	2E+02	0E+00	0.0	4.0	0.0	0	4	1	0.0	4.0	0.0
T_r^{1-3}	207	0.1	0.8	0.5	1202	-	66	207	-	207	49.8	34.8	33.5	33.7	33.5	33.0	4E+04	4E+04	2E+02	0.0	-	0.0	1203	-	45	0.0	-	0.0
T_r^{1-4}	198	0.1	0.2	0.2	1	1	0	198	198	198	20.9	10.2	10.1	9.7	7.1	4.1	2E+02	7E+01	0E+00	0.0	1.4	0.0	0	0	0	0.0	1.4	1.4
T_r^{1-5}	84	0.1	0.2	0.2	0	0	0	84	84	84	13.7	0.2	0.2	0.2	0.0	0.0	0E+00	0E+00	0E+00	0.0	0.0	0.0	0	0	0	0.0	0.0	0.0
T_r^{2-1}	371	0.1	0.6	0.5	6	15	4	371	371	371	23.9	3.9	1.7	3.8	1.9	1.3	6E+02	3E+02	9E+00	2.8	2.8	2.8	3	8	1	2.8	2.8	2.8
T_r^{2-2}	374	0.1	0.4	0.4	6	11	15	374	374	374	20.8	4.8	1.4	3.7	2.1	1.2	1E+03	2E+02	7E+01	3.9	3.9	3.9	1	8	2	3.9	3.9	3.9
T_r^{2-3}	437	0.2	1.2	1.2	-	-	-	-	-	437	38.2	14.3	12.0	15.0	12.8	11.5	3E+04	1E+04	9E+03	-	-	0.0	-	-	1762	-	-	0.0
T_r^{2-4}	403	0.1	0.4	0.4	2	4	2	403	403	403	17.3	3.1	1.6	1.5	1.4	1.2	6E+02	2E+02	2E+02	4.1	4.1	4.1	1	4	1	2.1	4.1	2.1
T_r^{2-5}	352	0.1	0.3	0.3	1	1	1	352	352	352	15.8	1.4	0.0	1.2	0.4	0.0	6E+01	0E+00	0E+00	0.0	0.0	0.0	0	1	0	4.3	0.0	4.3
T_r^{3-1}	172	0.1	0.4	0.6	6	21	7	172	172	172	24.4	5.2	3.7	5.8	3.2	2.4	5E+02	5E+02	3E+01	0.0	0.0	0.0	0	16	2	0.0	1.8	1.8
T_r^{3-2}	175	0.1	0.4	0.4	2	1	1	175	175	175	22.9	1.6	0.5	1.8	0.2	0.0	1E+02	0E+00	0E+00	0.0	0.0	0.0	0	1	0	0.0	0.0	0.0
T_r^{3-3}	229	0.1	0.4	0.6	32	150	13	229	229	229	31.0	12.4	9.4	11.6	10.2	5.1	2E+03	3E+03	6E+01	2.5	2.5	2.5	16	149	11	3.7	2.5	2.5
T_r^{3-4}	168	0.1	0.4	0.5	80	361	20	168	168	168	31.0	13.5	12.8	14.1	12.3	10.9	1E+04	1E+04	1E+02	0.0	0.0	0.0	24	244	16	0.0	0.0	0.0
T_r^{3-5}	190	0.1	0.3	0.5	9	88	20	190	190	190	26.3	8.6	8.0	8.2	8.1	6.6	8E+02	3E+03	1E+02	3.0	3.0	3.0	0	80	15	3.0	3.0	3.0
T_r^{4-1}	126	0.1	0.4	0.5	84	354	19	126	126	126	40.5	18.4	13.1	17.2	16.1	12.6	5E+03	1E+04	9E+01	0.0	0.0	0.0	54	298	14	0.0	0.0	0.0
T_r^{4-2}	183	0.1	0.3	0.5	34	224	16	183	183	183	40.4	14.5	12.2	14.2	13.2	8.0	1E+03	6E+03	8E+01	2.6	2.6	2.6	20	219	15	2.6	2.6	2.6
T_r^{4-3}	110	0.1	0.3	0.3	0	1	1	110	110	110	23.6	0.5	0.0	0.0	0.3	0.0	0E+00	0E+00	0E+00	0.0	0.0	0.0	0	1	1	2.9	0.0	0.0
T_r^{4-4}	143	0.1	0.3	0.4	21	108	13	143	143	143	40.6	17.7	17.3	18.1	17.3	14.4	2E+03	4E+03	1E+02	0.0	0.0	0.0	1	100	2	0.0	0.0	0.0
T_r^{4-5}	60	0.1	0.4	0.3	2	17	1	60	60	60	28.3	10.7	9.6	9.0	9.2	0.0	2E+02	2E+02	0E+00	0.0	0.0	0.0	0	17	1	0.0	0.0	0.0
T_r^{5-1}	293	0.1	0.6	0.6	-	-	-	-	-	293	44.4	22.4	21.2	22.3	21.7	20.9	8E+04	5E+04	3E+04	-	-	2.5	-	-	2343	-	-	2.5
T_r^{5-2}	157	0.1	0.6	0.6	55	51	17	157	157	157	31.2	5.5	2.9	5.7	4.2	2.4	2E+03	5E+02	4E+01	0.0	0.0	0.0	50	51	17	0.0	0.0	0.0
T_r^{5-3}	133	0.1	0.6	0.5	258	-	24	133	-	133	38.3	22.3	21.1	21.5	20.2	16.0	1E+04	5E+04	7E+01	2.5	-	2.5	255	-	16	2.5	-	2.5
T_r^{5-4}	126	0.1	0.6	0.5	40	157	34	126	126	126	30.2	8.0	6.8	6.6	6.6	3.7	2E+03	3E+03	3E+02	0.0	0.0	0.0	26	151	4	0.0	0.0	2.2
T_r^{5-5}	217	0.1	0.4	0.6	-	-	228	217	217	217	35.9	11.8	9.2	11.3	10.1	7.6	8E+04	1E+05	2E+03	0.0	0.0	0.0	2741	711	62	0.0	0.0	0.0

Table B.4: Test results for the COCMST(0.05^{ϵ}) problem: $T_{rand}^1 - T_{rand}^5$.

ins.	B^*	LP-time (s)			MIP-time (s)			B			Gap_{LP}(%)			Gap_0 (%)			B&B Tree Nodes			Gap_U^* (%)			Feas.-time (s)			Gap_U^1 (%)		
		RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF
T_r^{6-1}	207	0.1	0.8	0.9	208	-	151	207	-	207	34.3	17.0	11.3	15.0	13.3	10.9	1E+04	6E+04	8E+02	0.0	-	0.0	89	-	7	0.0	-	2.6
T_r^{6-2}	266	0.1	0.7	0.8	19	373	23	266	266	266	28.9	10.7	8.7	11.2	9.2	5.3	1E+03	4E+03	4E+01	3.6	0.0	0.0	1	172	4	2.7	0.0	2.7
T_r^{6-3}	228	0.1	0.8	1.2	12	95	25	228	228	228	28.9	10.2	7.7	9.4	9.4	6.6	7E+02	8E+02	8E+01	2.9	2.9	2.9	0	1	5	2.9	2.9	2.9
T_r^{6-4}	126	0.1	1.0	1.1	6	18	6	126	126	126	22.8	5.0	2.4	4.6	3.2	1.5	7E+02	2E+02	0E+00	0.0	0.0	0.0	0	4	2	0.0	0.0	0.0
T_r^{6-5}	183	0.1	1.0	1.1	39	342	146	183	183	183	29.5	10.8	10.5	11.4	10.1	9.5	4E+03	3E+03	7E+02	2.2	2.2	2.2	1	2599	81	2.2	2.2	2.2
T_r^{7-1}	391	0.2	1.6	2.0	267	915	19	391	391	391	26.6	4.7	1.0	4.2	2.7	0.8	9E+03	6E+03	2E+01	3.6	3.6	3.6	3	474	0	0.0	3.6	3.6
T_r^{7-2}	547	0.2	1.7	2.1	-	-	1555	547	-	547	26.7	5.5	3.5	6.1	4.1	3.4	5E+04	1E+04	6E+03	4.0	-	4.0	1244	-	182	4.0	-	4.0
T_r^{7-3}	409	0.2	1.4	1.9	-	-	-	434	-	409	28.4	6.6	3.1	6.4	4.4	2.8	5E+04	1E+04	9E+03	2.9	-	2.9	3486	-	365	2.9	-	1.0
T_r^{7-4}	435	0.2	1.5	2.3	-	-	1029	-	437	435	30.3	8.3	4.4	8.3	6.0	4.3	4E+04	9E+03	2E+03	-	4.8	4.8	-	4	472	-	25.9	4.8
T_r^{7-5}	603	0.2	1.6	2.0	-	-	-	-	-	603	34.8	12.5	7.4	11.6	9.8	7.2	2E+04	7E+03	5E+03	-	-	3.7	-	-	2257	-	-	3.7
T_r^{8-1}	490	0.1	0.6	1.3	166	1647	230	490	490	490	32.7	16.0	9.2	14.4	12.1	8.9	8E+03	7E+03	1E+03	2.3	2.3	2.3	25	1229	135	0.0	0.0	0.0
T_r^{8-2}	470	0.1	0.6	1.5	45	844	41	470	470	470	26.8	11.9	9.9	11.4	10.9	9.5	1E+03	4E+03	5E+01	0.0	0.0	0.0	39	815	36	0.0	3.4	0.0
T_r^{8-3}	225	0.2	4.5	3.2	11	12	4	225	225	225	27.6	0.0	0.0	0.9	0.0	0.0	7E+01	0E+00	0E+00	3.1	3.1	3.1	9	12	4	3.1	3.1	3.1
T_r^{8-4}	305	0.1	0.7	1.4	2	23	10	305	305	305	21.0	5.0	3.0	3.4	3.2	2.1	6E+01	4E+01	1E+01	0.0	0.0	0.0	0	10	0	0.0	3.7	0.0
T_r^{8-5}	386	0.2	3.7	3.5	3515	1434	56	386	386	386	29.8	5.3	0.7	5.7	3.7	0.6	8E+04	4E+03	2E+01	3.6	3.6	3.6	5	623	18	-8.0	3.6	3.6
T_r^{9-1}	1006	0.2	2.4	3.1	-	-	1796	-	-	1006	28.2	10.8	6.1	10.0	8.2	5.9	3E+04	7E+03	5E+03	-	-	3.0	-	-	381	-	-	4.5
T_r^{9-2}	791	0.2	2.3	3.0	490	1304	413	791	791	791	23.5	5.0	2.4	5.6	3.2	2.1	1E+04	4E+03	9E+02	2.9	2.9	2.9	57	587	199	2.9	2.9	2.9
T_r^{9-3}	683	0.2	1.4	2.9	144	3065	127	683	683	683	16.8	4.6	3.3	4.3	3.9	2.9	6E+03	1E+04	2E+02	3.4	3.4	3.4	67	368	91	4.3	0.0	3.4
T_r^{9-4}	821	0.2	2.1	2.9	-	-	755	821	-	821	23.1	7.5	2.0	7.4	4.7	1.6	6E+04	7E+03	2E+03	3.9	-	3.9	422	-	564	3.9	-	3.9
T_r^{9-5}	613	0.2	1.3	2.4	17	162	48	613	613	613	17.6	2.0	1.3	1.6	1.2	1.0	8E+02	7E+02	5E+01	3.8	3.8	3.8	2	69	21	3.8	3.8	3.8
T_r^{10-1}	292	0.2	3.0	3.3	3186	-	151	292	-	292	31.5	7.1	3.2	7.8	5.6	3.0	7E+04	6E+03	1E+02	3.0	-	0.0	348	-	79	3.0	-	0.0
T_r^{10-2}	233	0.2	3.0	3.2	-	-	431	233	233	233	29.2	8.8	4.8	8.7	7.5	4.7	7E+04	8E+03	7E+02	3.7	3.7	3.7	512	719	190	3.7	3.7	3.7
T_r^{10-3}	481	0.2	2.5	2.8	-	-	648	481	-	481	30.4	9.9	4.1	8.5	8.1	4.0	6E+04	1E+04	1E+03	0.0	-	0.0	6	-	209	4.5	-	0.0
T_r^{10-4}	295	0.2	2.6	2.8	-	-	1941	-	-	295	33.9	10.2	5.2	11.5	8.1	5.0	3E+04	3E+03	2E+03	-	-	3.3	-	-	665	-	-	3.3
T_r^{10-5}	393	0.2	3.3	7.4	2714	-	626	393	397	393	31.0	11.3	8.7	11.3	10.6	8.5	5E+04	1E+04	1E+03	-2.2	4.1	-2.2	41	2567	163	-2.2	4.1	2.1

Table B.5: Test results for the COCMST(0.05^{ϵ}) problem: $T_{rand}^{6} - T_{rand}^{10}$.

ins.	B^*	LP-time (s)			MIP-time (s)			B			Gap_{LP}(%)			Gap_0 (%)			B&B Tree Nodes			Gap_U^* (%)			Feas.-time (s)			Gap_U^1 (%)		
		RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF
T_{3tc}^{1-1}	15164	0.1	0.4	0.2	0	1	0	15164	15164	15164	5.9	3.2	3.2	2.5	2.2	2.3	3E+01	3E+01	2E+01	-0.3	-0.3	-0.3	0	0	0	2.4	1.2	1.2
T_{3tc}^{1-2}	17270	0.1	0.5	0.4	0	3	1	17270	17270	17270	6.6	2.3	0.8	1.1	0.8	0.3	0E+00	2E+01	0E+00	0.8	0.8	0.8	0	1	0	0.3	2.8	0.3
T_{3tc}^{1-3}	14946	0.2	0.6	0.5	0	1	1	14946	14946	14946	7.5	0.0	0.0	0.3	0.0	0.0	0E+00	0E+00	0E+00	4.6	4.6	4.6	0	1	0	4.6	4.6	4.6
T_{3tc}^{1-4}	19470	0.1	0.6	0.6	2	11	2	19470	19470	19470	13.3	1.0	0.5	0.6	0.5	0.3	2E+02	1E+02	8E+01	2.0	2.0	2.0	0	1	0	2.0	0.2	2.0
T_{3tc}^{1-5}	15730	0.1	0.6	0.5	0	2	1	15730	15730	15730	14.4	0.0	0.0	0.0	0.0	0.0	0E+00	0E+00	0E+00	4.2	4.2	4.2	0	1	0	5.5	5.9	5.9
T_{3tc}^{2-1}	20178	0.2	1.9	1.8	64	371	22	20178	20178	20178	12.1	2.2	1.3	1.9	1.3	1.2	6E+03	8E+03	2E+03	4.8	4.8	4.8	1	37	7	4.7	4.6	0.8
T_{3tc}^{2-2}	24572	0.2	0.6	0.4	0	1	1	24572	24572	24572	2.7	0.0	0.0	1.1	0.0	0.0	0E+00	0E+00	0E+00	-0.2	-0.2	-0.2	0	1	0	-0.2	-0.2	-0.2
T_{3tc}^{2-3}	30290	0.2	0.7	0.4	0	1	1	30290	30290	30290	4.4	0.0	0.0	0.0	0.0	0.0	0E+00	0E+00	0E+00	4.3	4.3	4.3	0	2	0	1.8	4.3	3.9
T_{3tc}^{2-4}	29476	0.2	0.7	0.5	0	2	1	29476	29476	29476	11.2	0.0	0.0	0.0	0.0	0.0	0E+00	0E+00	0E+00	-0.4	-0.4	-0.4	0	2	0	-0.4	-0.4	2.3
T_{3tc}^{2-5}	23030	0.2	0.8	0.7	2	12	2	23030	23030	23030	12.4	0.4	0.2	0.2	0.2	0.1	9E+01	1E+02	3E+00	0.9	0.9	0.9	0	3	1	3.7	3.8	2.8
T_{3tc}^{3-1}	28578	0.2	4.5	14.8	51	624	308	28578	28578	28578	15.0	0.7	0.6	0.7	0.6	0.5	2E+03	1E+03	2E+02	2.0	2.0	2.0	0	30	1	1.8	2.0	1.6
T_{3tc}^{3-2}	26842	0.2	3.7	5.6	108	339	69	26842	26842	26842	12.4	1.0	0.3	1.7	0.8	0.2	4E+03	8E+02	2E+02	3.1	3.1	3.1	2	87	6	-1.1	0.5	2.6
T_{3tc}^{3-3}	29444	0.2	4.4	5.1	197	2207	220	29444	29444	29444	15.8	1.5	0.6	1.8	0.8	0.5	5E+03	3E+03	2E+02	0.6	0.6	0.6	0	1217	9	-2.0	-2.0	0.9
T_{3tc}^{3-4}	19238	0.2	5.7	7.9	-	-	3021	19450	-	19238	16.8	6.5	2.9	6.0	5.0	2.5	7E+04	6E+03	1E+04	4.2	-	3.8	745	-	298	3.8	-	6.1
T_{3tc}^{3-5}	30366	0.2	1.8	2.8	37	1003	41	30366	30366	30366	12.2	5.5	3.2	4.5	3.9	3.1	2E+03	5E+03	6E+02	6.1	6.1	6.1	1	140	12	5.8	5.3	5.2
T_{3tc}^{4-1}	47126	0.3	6.3	11.6	785	-	540	47126	47474	47126	15.6	1.2	0.4	0.9	0.7	0.4	1E+04	2E+03	3E+02	-0.1	-0.8	-0.1	0	961	14	3.7	-0.8	2.2
T_{3tc}^{4-2}	35822	0.3	6.4	11.4	-	-	2258	36408	36032	35822	12.6	3.1	0.8	2.7	2.3	0.7	8E+04	4E+03	5E+03	6.7	6.7	6.7	3	3510	439	6.4		6.2
T_{3tc}^{4-3}	38928	0.3	2.6	2.8	1	7	5	38928	38928	38928	4.9	0.0	0.0	0.0	0.0	0.0	0E+00	0E+00	0E+00	2.7	2.7	2.7	0	7	1	2.8	2.7	-0.4
T_{3tc}^{4-4}	64492	0.2	3.6	3.1	104	142	41	64492	64492	64492	6.3	1.3	0.6	1.8	0.6	0.5	2E+04	3E+03	1E+03	5.2	5.2	5.2	0	5	1	4.5	5.2	4.8
T_{3tc}^{4-5}	41616	0.2	8.7	4.9	-	-	715	41616	42092	41616	11.8	4.0	2.0	3.6	2.4	2.0	1E+05	1E+04	1E+04	4.7	4.9	4.7	3	154	8	4.1	3.2	3.6
T_{3tc}^{5-1}	21566	0.2	3.2	22.3	7	714	27	21566	21566	21566	6.0	0.8	0.2	0.7	0.6	0.1	4E+02	1E+02	0E+00	0.8	0.8	0.8	1	608	24	3.9	0.8	3.9
T_{3tc}^{5-2}	19950	0.2	4.4	25.1	144	-	1145	19950	-	19950	15.4	1.4	1.4	1.4	1.2	1.3	3E+03	2E+02	4E+02	6.3	-	6.3	1	-	361	6.3	-	4.5
T_{3tc}^{5-3}	25582	0.2	8.2	24.1	-	-	-	25582	-	25588	21.5	3.0	2.9	3.1	2.9	2.9	3E+04	2E+02	1E+03	0.9	-	-0.7	7	-	39	0.9	-	-0.7
T_{3tc}^{5-4}	30288	0.2	14.0	17.3	371	-	2277	30288	31232	30288	17.6	3.7	3.7	3.7	3.7	3.5	1E+04	2E+03	2E+03	1.7	1.7	1.7	103	1802	256	1.7	1.7	2.2
T_{3tc}^{5-5}	40764	0.2	3.7	24.8	304	-	599	40764	40764	40764	19.2	3.2	2.8	4.0	3.1	2.7	5E+03	2E+03	5E+02	2.4	2.4	2.4	22	1054	205	2.0	2.4	2.4
T_{3tc}^{6-1}	-	0.4	85.8	25.4	-	-	-	-	-	-	-	-	-	-	-	-	2E+04	2E+02	6E+02	-	-	-	-	-	-	-	-	-
T_{3tc}^{6-2}	65570	0.4	37.7	105.0	107	3212	115	65570	65570	65570	14.2	0.4	0.3	0.3	0.4	0.3	7E+02	3E+02	1E+01	2.5	2.5	2.5	0	2757	2	3.4	2.5	0.2
T_{3tc}^{6-3}	51404	0.4	86.7	91.9	898	-	-	51404	-	51404	15.6	1.4	1.3	1.4	1.4	1.3	6E+03	4E+02	3E+02	0.9	-	0.9	0	-	2	2.3	-	2.0
T_{3tc}^{6-4}	69166	0.4	7.3	78.5	-	-	2857	69166	70950	69166	12.5	1.9	1.1	1.8	1.7	1.1	2E+04	1E+03	2E+03	4.9	0.1	4.9	12	1662	128	4.9	0.1	0.1
T_{3tc}^{6-5}	-	0.4	39.9	65.1	-	-	-	-	-	-	-	-	-	-	-	-	1E+04	5E+01	4E+02	-	-	-	-	-	-	-	-	-

Table B.6: Test results for the COCMST(0.05^{ϵ}) problem: T_{3tc}.

ins.	B^*	LP-time (s)			MIP-time (s)			B			Gap_{LP}(%)			Gap_0 (%)			B&B Tree Nodes			Gap_U^* (%)			Feas.-time (s)			Gap_U^1 (%)		
		RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF
T_r^{1-1}	131	0.1	0.2	0.2	0	2	0	131	131	131	17.4	3.5	2.0	0.8	1.4	1.2	0E+00	0E+00	0E+00	0.0	0.0	0.0	0	1	0	0.0	0.0	0.0
T_r^{1-2}	157	0.1	0.2	0.2	1	6	3	157	157	157	21.1	11.1	10.3	11.3	8.8	7.1	3E+02	2E+02	5E+01	0.0	0.0	0.0	1	5	1	0.0	0.0	0.0
T_r^{1-3}	207	0.1	0.7	0.5	1933	-	66	207	-	207	49.8	34.7	33.4	33.7	33.5	33.0	5E+04	4E+04	2E+02	0.0	-	0.0	1935	-	45	0.0	-	0.0
T_r^{1-4}	198	0.1	0.2	0.2	0	1	1	198	198	198	20.5	9.9	9.8	9.6	4.9	2.7	5E+01	4E+01	5E+00	0.0	0.0	0.0	0	1	0	0.0	0.0	0.0
T_r^{1-5}	84	0.1	0.2	0.2	0	0	0	84	84	84	13.2	0.0	0.0	0.3	0.0	0.0	0E+00	0E+00	0E+00	0.0	0.0	0.0	0	0	0	0.0	0.0	0.0
T_r^{2-1}	384	0.1	0.5	0.5	31	79	31	384	384	384	26.5	7.0	4.8	6.6	5.1	4.5	3E+03	2E+03	6E+02	0.7	0.7	0.7	4	46	10	0.0	0.0	0.0
T_r^{2-2}	375	0.1	0.4	0.4	9	10	8	375	375	375	21.0	4.8	1.4	3.7	2.1	1.0	2E+03	1E+02	1E+02	0.0	0.0	0.0	3	8	1	0.0	0.0	0.0
T_r^{2-3}	442	0.2	1.2	1.2	-	-	-	-	-	442	38.9	15.1	12.7	15.8	14.1	12.5	3E+04	1E+04	8E+03	-	-	0.0	-	-	360	-	-	0.0
T_r^{2-4}	406	0.1	0.4	0.4	4	5	2	406	406	406	17.6	3.5	1.8	1.8	1.5	1.2	8E+02	3E+02	6E+01	0.0	0.0	0.0	3	5	2	0.0	0.0	0.0
T_r^{2-5}	352	0.1	0.3	0.3	1	1	1	352	352	352	15.8	1.4	0.0	1.2	0.1	0.0	5E+01	0E+00	0E+00	0.0	0.0	0.0	0	1	0	0.0	0.0	0.0
T_r^{3-1}	172	0.1	0.4	0.6	6	9	4	172	172	172	24.4	4.9	3.3	5.0	3.3	2.2	4E+02	1E+02	5E+00	0.0	0.0	0.0	2	3	2	0.0	0.0	0.0
T_r^{3-2}	175	0.1	0.3	0.4	2	1	1	175	175	175	22.9	1.4	0.3	1.7	0.2	0.0	9E+01	0E+00	0E+00	0.0	0.0	0.0	0	1	0	0.0	0.0	0.0
T_r^{3-3}	250	0.1	0.4	0.6	25	289	16	250	250	250	36.8	19.7	17.0	19.1	17.6	12.9	1E+03	6E+03	7E+01	0.0	0.0	0.0	24	287	14	0.0	0.0	0.0
T_r^{3-4}	168	0.1	0.4	0.5	66	145	24	168	168	168	31.0	13.1	12.4	13.8	11.9	11.0	5E+03	5E+03	2E+02	0.0	0.0	0.0	28	69	12	0.0	0.0	0.0
T_r^{3-5}	192	0.1	0.3	0.5	50	201	22	192	192	192	27.1	9.5	8.8	9.2	9.1	7.1	4E+03	8E+03	1E+02	0.0	0.0	0.0	49	199	20	0.0	0.0	0.0
T_r^{4-1}	126	0.1	0.4	0.5	338	356	19	126	126	126	40.5	18.4	13.0	17.2	16.1	12.6	3E+04	1E+04	9E+01	0.0	0.0	0.0	223	297	14	0.0	0.0	0.0
T_r^{4-2}	185	0.1	0.3	0.5	34	130	10	185	185	185	41.1	15.4	13.1	15.0	14.1	7.0	1E+03	7E+03	6E+01	0.0	0.0	0.0	5	63	7	0.0	0.0	0.0
T_r^{4-3}	110	0.1	0.3	0.3	0	1	1	110	110	110	23.6	0.4	0.0	0.0	0.3	0.0	0E+00	0E+00	0E+00	0.0	0.0	0.0	0	1	1	0.0	0.0	0.0
T_r^{4-4}	143	0.1	0.3	0.4	17	66	17	143	143	143	40.6	17.7	17.2	18.0	17.2	14.4	2E+03	3E+03	1E+02	0.0	0.0	0.0	3	40	1	0.0	0.0	0.0
T_r^{4-5}	60	0.1	0.3	0.3	2	17	1	60	60	60	28.3	10.6	9.5	8.6	9.2	0.0	2E+02	2E+02	0E+00	0.0	0.0	0.0	0	17	1	0.0	0.0	0.0
T_r^{5-1}	305	0.2	0.6	0.6	-	-	-	-	-	305	46.6	25.5	24.1	25.3	24.6	23.6	8E+04	4E+04	3E+04	-	-	0.0	-	-	1805	-	-	0.0
T_r^{5-2}	157	0.1	0.6	0.6	44	28	13	157	157	157	31.2	5.5	2.6	5.7	3.7	2.2	2E+03	3E+02	3E+01	0.0	0.0	0.0	37	26	11	0.0	0.0	0.0
T_r^{5-3}	135	0.1	0.7	0.5	139	3155	30	135	135	135	39.3	23.4	22.2	22.6	21.4	14.5	7E+03	5E+04	9E+01	0.0	0.0	0.0	136	3088	22	0.0	0.0	0.0
T_r^{5-4}	126	0.1	0.7	0.6	31	53	35	126	126	126	30.2	7.9	6.6	6.2	6.3	3.7	2E+03	1E+03	7E+01	0.0	0.0	0.0	17	32	34	0.0	0.0	0.0
T_r^{5-5}	217	0.1	0.6	0.6	-	-	163	225	-	217	35.9	11.8	9.1	11.2	10.1	7.6	7E+04	5E+04	8E+02	0.0	-	0.0	2628	-	92	0.0	-	0.0

Table B.7: Test results for the COCMST(0.01^ϵ) problem: $T_{rand}^1 - T_{rand}^5$.

ins.	B^*	LP-time (s)			MIP-time (s)			B			Gap_{LP}(%)			Gap_0 (%)			B&B Tree Nodes			Gap_U^* (%)			Feas.-time (s)			Gap_U^1 (%)		
		RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF
T_r^{6-1}	207	0.1	0.6	0.9	202	1430	798	207	207	207	34.3	17.0	11.3	14.8	13.0	10.7	1E+04	1E+04	8E+03	0.0	0.0	0.0	19	688	7	0.0	0.0	0.0
T_r^{6-2}	266	0.1	0.8	0.8	17	347	23	266	266	266	28.9	10.7	8.5	11.3	9.2	5.0	9E+02	2E+03	4E+01	0.0	0.0	0.0	6	159	15	0.0	0.0	0.0
T_r^{6-3}	237	0.1	0.9	1.1	17	254	26	237	237	237	31.6	13.6	10.9	12.9	12.4	8.9	7E+02	3E+03	7E+01	0.0	0.0	0.0	1	71	5	0.0	0.0	0.0
T_r^{6-4}	126	0.1	1.0	1.0	3	18	17	126	126	126	22.8	4.8	2.1	4.4	3.2	1.2	3E+02	2E+02	4E+01	0.0	0.0	0.0	0	4	2	0.0	0.0	0.0
T_r^{6-5}	191	0.1	1.0	1.1	92	1307	194	191	191	191	32.5	14.4	14.0	15.0	14.0	13.0	6E+03	1E+04	5E+02	0.0	0.0	0.0	82	1298	171	0.0	0.0	0.0
T_r^{7-1}	398	0.2	1.6	2.0	-	-	178	398	398	398	27.9	6.4	2.6	6.0	4.2	2.4	1E+05	3E+04	3E+02	0.0	0.0	0.0	3	882	131	0.0	0.0	0.0
T_r^{7-2}	559	0.2	1.9	2.1	-	-	1634	562	-	559	28.3	7.5	5.4	8.0	6.3	5.1	3E+04	1E+04	4E+03	0.0	-	0.0	2008	-	580	0.0	-	0.0
T_r^{7-3}	409	0.2	1.5	1.8	-	-	872	-	-	409	28.4	6.5	3.1	6.5	5.2	2.8	5E+04	1E+04	2E+03	-	-	0.0	-	-	307	-	-	0.0
T_r^{7-4}	438	0.2	1.4	2.3	-	-	3322	-	-	438	30.8	9.0	4.8	8.8	6.5	4.7	3E+04	8E+03	8E+03	-	-	0.0	-	-	527	-	-	0.0
T_r^{7-5}	655	0.2	1.6	2.1	-	-	-	-	-	655	39.9	19.4	14.4	18.5	17.0	14.2	2E+04	7E+03	5E+03	-	-	0.0	-	-	832	-	-	0.0
T_r^{8-1}	493	0.1	0.6	1.4	75	972	76	493	493	493	33.1	16.5	9.5	15.0	13.5	9.2	2E+03	5E+03	3E+02	0.0	0.0	0.0	46	432	20	0.0	0.0	0.0
T_r^{8-2}	470	0.1	0.6	1.6	25	668	45	470	470	470	26.8	11.9	9.8	11.2	10.2	9.5	7E+02	2E+03	5E+01	0.0	0.0	0.0	24	614	44	0.0	0.0	0.0
T_r^{8-3}	241	0.2	4.2	3.8	1731	-	800	241	-	241	32.4	6.7	6.6	7.3	6.6	6.4	4E+04	2E+03	5E+02	-14.8	-	-14.8	166	-	11	-14.8	-	-14.8
T_r^{8-4}	305	0.1	0.8	1.4	2	14	4	305	305	305	20.7	4.1	2.3	2.7	2.1	1.1	9E+01	1E+01	0E+00	0.0	0.0	0.0	0	6	0	0.0	0.0	0.0
T_r^{8-5}	400	0.2	3.7	3.7	-	-	1644	400	411	400	32.2	8.7	4.2	9.0	6.9	4.0	6E+04	1E+04	1E+03	0.0	-17.4	0.0	6	687	786	-1.9	-17.4	-6.6
T_r^{9-1}	1017	0.2	3.0	3.5	-	-	-	-	-	1017	28.8	10.9	6.5	10.1	7.6	6.2	4E+04	7E+03	5E+03	-	-	0.8	-	-	1995	-	-	0.8
T_r^{9-2}	834	0.2	3.3	3.2	2733	-	1829	834	-	834	27.4	9.3	6.8	9.3	7.5	6.6	4E+04	6E+03	3E+03	0.0	-	0.0	473	-	333	0.0	-	0.7
T_r^{9-3}	686	0.2	1.8	2.8	181	2246	558	686	686	686	17.2	4.6	2.9	4.2	3.6	2.4	4E+03	7E+03	7E+02	0.0	0.0	0.0	170	1928	559	0.0	0.0	0.0
T_r^{9-4}	838	0.2	2.4	2.9	-	-	-	-	-	838	24.7	9.2	3.7	9.0	6.2	3.3	4E+04	5E+03	7E+03	-	-	-1.4	-	-	1056	-	-	-1.4
T_r^{9-5}	621	0.2	1.3	2.6	49	515	178	621	621	621	18.7	3.2	2.5	2.8	2.5	2.3	2E+03	3E+03	4E+02	0.0	0.0	0.0	28	166	50	0.0	0.0	0.0
T_r^{10-1}	292	0.2	2.3	5.1	532	-	198	292	-	292	31.5	7.1	3.1	7.7	4.6	2.9	2E+04	6E+03	2E+02	0.0	-	0.0	13	-	98	0.0	-	0.0
T_r^{10-2}	-	0.2	2.7	2.9	-	-	-	-	-	-	-	-	-	-	-	-	6E+04	6E+03	4E+03	-	-	-	-	-	-	-	-	-
T_r^{10-3}	481	0.2	2.5	3.1	-	-	1338	506	-	481	30.4	9.9	4.1	8.3	8.1	4.1	4E+04	1E+04	3E+03	0.0	-	0.0	1258	-	415	-2.4	-	0.0
T_r^{10-4}	303	0.2	2.6	2.9	-	-	1740	310	-	303	35.6	12.6	7.7	13.7	10.2	6.8	3E+04	4E+03	1E+03	0.0	-	0.0	1812	-	1222	0.0	-	0.0
T_r^{10-5}	393	0.2	2.9	4.6	-	-	1270	393	-	393	31.0	11.2	8.5	11.3	10.4	8.2	7E+04	8E+03	1E+03	-2.2	-	-2.2	158	-	1117	0.0	-	-3.7

Table B.8: Test results for the COCMST(0.01^ϵ) problem: $T_{rand}^6 - T_{rand}^{10}$.

ins.	B^*	LP-time (s)			MIP-time (s)			B			Gap_{LP}(%)			Gap_0 (%)			B&B Tree Nodes			Gap_U^* (%)			Feas.-time (s)			Gap_U^1 (%)		
		RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF	RD	MF	RDMF
T_{3tc}^{1-1}	15446	0.1	0.4	0.2	0	1	0.321	15446	15446	15446	7.7	2.5	2.4	2.4	1.8	1.7	4E+01	3E+01	3E+01	-1.5	-1.5	-1.5	0	1	0	-1.5	-1.5	-1.8
T_{3tc}^{1-2}	17334	0.1	0.6	0.4	0	2	0.839	17334	17334	17334	6.9	2.0	0.3	0.7	0.1	0.2	0E+00	0E+00	0E+00	-0.8	-0.8	-0.8	0	1	0	-0.4	-0.8	-0.8
T_{3tc}^{1-3}	15220	0.1	0.6	0.6	1	19	2.698	15220	15220	15220	9.2	1.6	1.5	1.6	1.5	1.1	4E+01	1E+02	9E+00	0.4	0.4	0.4	0	2	0	0.4	0.4	0.4
T_{3tc}^{1-4}	19538	0.1	0.6	0.6	2	15	1.524	19538	19538	19538	13.6	1.2	0.6	0.7	0.6	0.4	1E+02	2E+02	1E+01	0.8	0.8	0.8	0	2	0	0.2	0.4	0.2
T_{3tc}^{1-5}	15782	0.1	0.6	0.7	1	22	2.129	15782	15782	15782	14.7	0.3	0.3	0.3	0.3	0.1	2E+01	2E+01	0E+00	1.3	1.3	1.3	0	11	2	1.3	1.3	1.3
T_{3tc}^{2-1}	20358	0.2	2.0	1.7	95	210	10.645	20358	20358	20358	12.9	2.1	1.2	1.8	1.3	1.1	7E+03	4E+03	5E+02	0.2	0.2	0.2	5	85	3	-0.2	-0.2	0.4
T_{3tc}^{2-2}	24572	0.2	0.7	0.4	0	1	0.572	24572	24572	24572	2.7	0.0	0.0	1.1	0.0	0.0	0E+00	0E+00	0E+00	-0.2	-0.2	-0.2	0	1	0	-0.2	-0.2	-0.2
T_{3tc}^{2-3}	30356	0.2	0.9	0.5	0	2	0.711	30356	30356	30356	4.6	0.2	0.2	0.2	0.0	0.0	0E+00	0E+00	0E+00	-0.1	-0.1	-0.1	0	1	0	0.1	0.4	-0.1
T_{3tc}^{2-4}	29476	0.2	0.7	0.6	1	2	1.421	29476	29476	29476	11.2	0.0	0.0	0.0	0.0	0.0	0E+00	0E+00	0E+00	-0.4	-0.4	-0.4	0	3	1	-0.4	-0.4	-0.4
T_{3tc}^{2-5}	23048	0.2	0.9	0.9	3	16	1.695	23048	23048	23048	12.5	0.4	0.2	0.2	0.2	0.1	2E+02	2E+02	4E+01	-0.4	-0.4	-0.4	1	10	1	-0.4	0.2	-0.4
T_{3tc}^{3-1}	28578	0.2	4.7	18.1	68	380	86.501	28578	28578	28578	15.0	0.7	0.5	0.6	0.5	0.4	2E+03	5E+02	3E+01	2.0	2.0	2.0	0	34	1	1.6	2.0	1.6
T_{3tc}^{3-2}	26982	0.2	4.1	6.5	151	1264	110.536	26982	26982	26982	12.8	1.4	0.7	2.1	1.2	0.6	6E+03	4E+03	5E+02	-1.1	-1.1	-1.1	2	194	7	-1.1	-0.8	-0.8
T_{3tc}^{3-3}	29542	0.2	4.0	5.5	1168	-	193.564	29542	29542	29542	16.1	1.8	0.8	2.0	1.0	0.8	2E+04	4E+03	1E+02	-1.7	-2.0	-1.7	0	1406	16	-2.0	-1.7	-1.7
T_{3tc}^{3-4}	20520	0.2	5.8	5.0	-	-	-	-	-	20520	22.0	11.8	8.4	11.3	9.9	7.9	6E+04	5E+03	1E+04	-	-	1.7	-	-	2966	-	-	1.7
T_{3tc}^{3-5}	31162	0.2	1.9	2.7	54	1727	47.324	31162	31162	31162	14.5	7.7	5.0	6.6	5.6	4.9	3E+03	9E+03	7E+02	1.6	1.6	1.6	4	625	5	1.6	2.1	1.6
T_{3tc}^{4-1}	47126	0.3	5.5	12.1	570	3127	72.231	47126	47126	47126	15.6	1.2	0.2	0.9	0.4	0.1	7E+03	2E+03	1E+01	-0.1	-0.1	-0.1	0	296	15	-0.8	-0.8	-0.8
T_{3tc}^{4-2}	36070	0.3	7.6	12.1	3600	-	-	36490	38662	36070	13.2	3.6	1.2	3.2	2.7	1.0	7E+04	3E+03	9E+03	1.9	2.2	1.9	107	2683	221	1.9	2.3	1.9
T_{3tc}^{4-3}	38962	0.3	2.4	3.8	1	13	6.494	38962	38962	38962	5.0	0.1	0.1	0.0	0.0	0.0	0E+00	0E+00	0E+00	-0.4	-0.4	-0.4	0	11	1	0.4	-0.4	-0.4
T_{3tc}^{4-4}	64912	0.2	3.7	2.8	203	158	37.598	64912	64912	64912	6.9	1.5	0.7	2.2	0.7	0.6	3E+04	2E+03	1E+03	1.5	1.5	1.5	1	11	2	0.6	0.6	1.5
T_{3tc}^{4-5}	41956	0.2	8.9	5.5	-	-	970.385	41956	42388	41956	12.6	3.9	2.0	3.8	2.4	1.9	9E+04	9E+03	1E+04	1.1	1.0	1.1	29	392	43	1.2	0.4	1.0
T_{3tc}^{5-1}	21590	0.2	3.5	14.5	14	1916	89.054	21590	21590	21590	6.1	0.8	0.3	0.7	0.7	0.1	3E+02	1E+03	6E+01	-0.1	-0.1	-0.1	1	324	14	-0.1	-0.1	-0.1
T_{3tc}^{5-2}	19950	0.3	4.6	23.4	147	-	2789.044	19950	-	19950	15.4	1.3	1.3	1.3	1.3	1.1	2E+03	2E+02	7E+02	6.3	-	6.3	3	-	1328	6.3	-	6.3
T_{3tc}^{5-3}	25586	0.2	5.4	34.8	-	-	-	25586	-	25588	21.5	3.0	2.9	3.1	2.9	2.9	4E+04	2E+02	1E+03	-0.7	-	-0.7	7	-	47	-0.7	-	-0.7
T_{3tc}^{5-4}	31266	0.2	16.9	17.1	2268	-	-	31266	-	31326	20.2	6.6	6.6	6.6	6.6	6.4	7E+04	1E+03	2E+03	-2.0	-	-2.0	68	-	1551	-1.0	-	-2.0
T_{3tc}^{5-5}	41242	0.2	4.6	22.1	1321	-	1282.397	41242	-	41242	20.1	4.3	4.0	5.1	4.2	4.0	2E+04	1E+03	2E+03	0.0	-	0.0	172	-	113	-1.1	-	-1.1
T_{3tc}^{6-1}	-	0.3	94.5	24.7	-	-	-	-	-	-	-	-	-	-	-	-	2E+04	1E+02	1E+03	-	-	-	-	-	-	-	-	-
T_{3tc}^{6-2}	65578	0.4	37.6	105.8	145	-	156.778	65578	-	65578	14.2	0.4	0.2	0.2	0.3	0.2	1E+03	7E+01	2E+01	-0.3	-	-0.3	0	-	2	-0.3	-	0.2
T_{3tc}^{6-3}	51504	0.4	91.0	92.4	1744	-	3182.067	51504	-	51504	15.8	1.4	1.3	1.5	1.4	1.1	1E+04	2E+02	3E+02	-0.6	-	-0.6	0	-	43	-1.0	-	-1.0
T_{3tc}^{6-4}	69238	0.3	7.2	150.4	-	-	3297.404	69280	71412	69238	12.6	2.0	1.1	1.9	1.9	1.1	2E+04	1E+03	3E+03	0.1	0.1	0.1	97	3133	44	0.3	0.1	0.1
T_{3tc}^{6-5}	-	0.4	43.9	75.9	-	-	-	-	-	-	-	-	-	-	-	-	1E+04	2E+01	5E+02	-	-	-	-	-	-	-	-	-

Table B.9: Test results for the COCMST(0.01^ϵ) problem: T_{3tc}.

C Computational results for the TE-MSTP decision problem

Key:

- ins. : instance reference;
- Feas.-time (s) : time it takes to find a first feasible solution;
- U : worst-case edge utilization corresponding to the solution found for the TE-MSTP decision problem;
- $Gap_U^D(\%)$: gap between B^* and U;
- B&B Tree Nodes : number of nodes in the B&B enumeration tree.

ins.	Feas.-time (s)						B&B Tree Nodes						Gap_U^D					
	$\epsilon = 0.2$		$\epsilon = 0.05$		$\epsilon = 0.01$		$\epsilon = 0.2$		$\epsilon = 0.05$		$\epsilon = 0.05$		$\epsilon = 0.2$		$\epsilon = 0.05$		$\epsilon = 0.01$	
	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF
T_r^{1-1}	0	0	0	0	0	0	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	16.7	5.6	0.0	0.0	0.0	0.0
T_r^{1-2}	0	1	1	1	1	1	0E+00	0E+00	2E+02	0E+00	9E+01	0E+00	16.7	16.7	0.0	0.0	0.0	0.0
T_r^{1-3}	6	87	117	126	346	126	3E+02	3E+02	3E+03	1E+02	6E+03	1E+02	16.7	16.7	0.0	0.0	0.0	0.0
T_r^{1-4}	0	0	0	0	0	0	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	5.9	19.1	1.5	1.5	0.0	0.0
T_r^{1-5}	0	0	0	0	0	0	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	20.0	20.0	0.0	0.0	0.0	0.0
T_r^{2-1}	1	1	49	8	76	23	0E+00	0E+00	2E+03	1E+02	3E+03	4E+02	20.0	17.9	2.9	2.9	0.0	0.0
T_r^{2-2}	0	1	2	1	47	12	0E+00	0E+00	2E+02	0E+00	4E+03	2E+02	18.4	18.4	4.1	4.1	0.0	0.0
T_r^{2-3}	86	56	-	99	-	239	7E+02	5E+01	2E+04	3E+02	2E+04	3E+02	17.6	17.6	-	2.9	-	0.0
T_r^{2-4}	0	0	3	2	2	2	0E+00	0E+00	8E+02	1E+02	3E+02	6E+01	17.0	10.6	4.3	4.3	0.0	0.0
T_r^{2-5}	0	0	0	0	0	0	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	18.2	13.6	4.5	4.5	0.0	0.0
T_r^{3-1}	3	4	1	1	11	11	1E+02	0E+00	0E+00	0E+00	3E+02	1E+02	7.1	14.3	0.0	1.8	0.0	0.0
T_r^{3-2}	0	0	0	0	0	0	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	9.4	12.5	0.0	0.0	0.0	0.0
T_r^{3-3}	3	2	6	11	29	10	5E+01	0E+00	2E+02	7E+01	1E+03	3E+01	17.9	19.2	3.8	3.8	0.0	0.0
T_r^{3-4}	1	13	39	11	6	14	0E+00	4E+01	2E+03	3E+01	4E+02	5E+01	16.7	16.7	0.0	3.7	0.0	0.0
T_r^{3-5}	0	12	65	13	132	15	0E+00	4E+01	3E+03	5E+01	6E+03	5E+01	12.5	17.2	3.1	3.1	0.0	0.0
T_r^{4-1}	0	2	14	15	2	15	0E+00	0E+00	9E+02	4E+01	5E+01	4E+01	16.7	18.8	0.0	0.0	0.0	0.0
T_r^{4-2}	9	7	20	9	33	7	6E+02	4E+01	8E+02	4E+01	1E+03	1E+01	18.4	18.4	2.6	0.0	0.0	0.0
T_r^{4-3}	0	1	0	1	0	2	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	18.2	9.1	3.0	0.0	0.0	0.0
T_r^{4-4}	0	2	3	2	14	1	0E+00	0E+00	2E+02	0E+00	8E+02	0E+00	16.7	8.3	0.0	0.0	0.0	0.0
T_r^{4-5}	0	0	6	1	1	1	0E+00	0E+00	5E+02	0E+00	0E+00	0E+00	20.0	20.0	0.0	0.0	0.0	0.0
T_r^{5-1}	7	28	-	101	1050	1161	3E+02	4E+01	7E+04	4E+02	4E+04	4E+03	15.4	17.9	-	2.6	-	0.0
T_r^{5-2}	0	23	86	67	25	29	0E+00	3E+01	3E+03	8E+01	1E+03	3E+01	18.5	18.5	3.7	3.7	0.0	0.0
T_r^{5-3}	8	28	366	36	190	17	3E+02	3E+01	1E+04	7E+01	5E+03	5E+01	7.7	15.4	0.0	2.6	0.0	0.0
T_r^{5-4}	19	19	9	104	148	5	1E+03	3E+01	3E+02	1E+02	4E+03	0E+00	15.9	15.9	2.3	0.0	0.0	0.0
T_r^{5-5}	17	80	271	184	-	57	8E+02	3E+02	7E+03	7E+02	6E+04	8E+01	18.5	18.5	3.7	3.7	-	0.0

Table C.1: Test results for the TE-MSTP(Λ) decision problem: $T_{rand}^1 - T_{rand}^5$.

ins.	Feas.-time (s)						B&B Tree Nodes						Gap_U^D					
	$\epsilon = 0.2$		$\epsilon = 0.05$		$\epsilon = 0.01$		$\epsilon = 0.2$		$\epsilon = 0.05$		$\epsilon = 0.05$		$\epsilon = 0.2$		$\epsilon = 0.05$		$\epsilon = 0.01$	
	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF
T_r^{6-1}	0	6	20	139	3	35	0E+00	0E+00	1E+03	3E+02	8E+01	5E+01	15.8	15.8	0.0	0.0	0.0	0.0
T_r^{6-2}	1	4	34	30	5	5	0E+00	0E+00	2E+03	1E+02	2E+02	0E+00	13.9	18.5	3.7	3.7	0.0	0.0
T_r^{6-3}	0	0	1	5	1	9	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	5.9	5.9	2.9	2.9	0.0	0.0
T_r^{6-4}	0	6	0	6	0	8	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	14.7	14.7	0.0	0.0	0.0	0.0
T_r^{6-5}	0	0	14	53	216	121	0E+00	0E+00	6E+02	1E+02	9E+03	4E+02	15.6	20.0	2.2	2.2	0.0	0.0
T_r^{7-1}	0	0	2	0	16	172	0E+00	0E+00	0E+00	0E+00	2E+02	3E+02	18.5	13.6	3.7	3.7	0.0	0.0
T_r^{7-2}	-	109	67	149	-	3475	2E+04	1E+02	5E+02	2E+02	2E+04	9E+03	-	16.7	4.2	4.2	-	0.0
T_r^{7-3}	35	199	474	207	-	331	6E+02	3E+02	4E+03	2E+02	3E+04	2E+02	18.2	18.2	3.0	3.0	-	0.0
T_r^{7-4}	222	174	-	156	-	3109	3E+03	3E+02	3E+04	8E+01	3E+04	8E+03	20.0	17.5	-	0.0	-	0.0
T_r^{7-5}	1126	211	-	-	-	-	8E+03	4E+02	1E+04	2E+03	1E+04	2E+03	15.4	19.2	-	-	-	-
T_r^{8-1}	4	6	121	32	35	78	2E+02	0E+00	3E+03	2E+01	7E+02	7E+01	14.3	14.3	0.0	2.4	0.0	0.0
T_r^{8-2}	1	32	8	136	48	52	0E+00	7E+01	2E+02	2E+02	1E+03	4E+01	17.9	19.0	3.6	2.4	0.0	0.0
T_r^{8-3}	5	24	-	954	-	2854	0E+00	0E+00	2E+04	8E+02	5E+04	1E+03	16.1	16.1	-	3.2	-	-1.1
T_r^{8-4}	0	0	0	0	0	4	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	7.7	7.7	0.0	0.0	0.0	0.0
T_r^{8-5}	4	0	6	617	-	908	0E+00	0E+00	0E+00	7E+01	2E+04	5E+02	16.0	18.5	0.0	3.7	-	0.0
T_r^{9-1}	15	103	760	318	228	300	2E+02	6E+01	7E+03	1E+03	2E+03	6E+02	19.5	17.2	4.7	4.3	0.8	0.0
T_r^{9-2}	31	107	444	166	13	310	6E+02	6E+01	4E+03	3E+02	8E+01	5E+02	13.1	16.4	4.5	3.0	0.0	0.7
T_r^{9-3}	8	38	61	91	290	406	1E+02	2E+01	2E+03	2E+02	2E+03	1E+03	16.1	14.3	3.6	3.1	0.0	0.0
T_r^{9-4}	3	61	397	164	329	835	0E+00	7E+01	4E+03	6E+02	3E+03	3E+03	17.8	19.2	4.1	4.1	0.0	0.0
T_r^{9-5}	0	29	12	41	12	24	0E+00	2E+01	2E+02	3E+01	2E+02	6E+01	13.2	19.3	2.6	0.0	0.0	0.0
T_r^{10-1}	3	20	-	-	992	-	0E+00	0E+00	3E+04	1E+03	8E+03	6E+02	18.8	9.4	-	-	-	-
T_r^{10-2}	19	21	127	2177	-	-	3E+02	0E+00	3E+03	4E+02	2E+04	1E+03	12.8	17.9	2.6	3.8	-	-
T_r^{10-3}	0	216	1084	362	797	-	0E+00	7E+01	2E+04	2E+02	1E+04	2E+03	14.3	19.0	3.6	1.8	0.0	-
T_r^{10-4}	487	765	1105	-	1993	-	5E+03	7E+02	6E+03	6E+02	3E+04	6E+02	19.5	19.5	3.4	-	0.0	-
T_r^{10-5}	3	21	-	464	10	1337	0E+00	0E+00	5E+04	3E+02	7E+01	1E+03	19.1	19.1	-	4.3	-	-0.7

Table C.2: Test results for the TE-MSTP(Λ) decision problem: $T_{rand}^6 - T_{rand}^{10}$.

ins.	Feas.-time (s)						Nodes						Gap_U^D					
	$\epsilon = 0.2$		$\epsilon = 0.05$		$\epsilon = 0.01$		$\epsilon = 0.2$		$\epsilon = 0.05$		$\epsilon = 0.05$		$\epsilon = 0.2$		$\epsilon = 0.05$		$\epsilon = 0.01$	
	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF	RD	RDMF
T_{3tc}^{1-1}	0	0	0	0	0	0	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	13.8	9.9	2.5	1.3	-1.4	-1.8
T_{3tc}^{1-2}	0	0	0	0	0	0	0E+00	0E+00	0E+00	1E+00	0E+00	0E+00	0.3	0.3	0.3	0.3	-0.4	-0.8
T_{3tc}^{1-3}	0	0	0	0	0	0	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	15.7	14.9	4.8	4.8	0.4	0.4
T_{3tc}^{1-4}	0	0	0	0	0	0	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	2.0	17.4	2.0	2.0	0.2	0.2
T_{3tc}^{1-5}	0	0	0	0	0	1	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	15.1	16.8	5.9	6.2	1.4	1.4
T_{3tc}^{2-1}	0	2	1	4	4	1	0E+00	0E+00	0E+00	5E+01	3E+02	0E+00	15.4	17.7	1.3	2.4	0.7	0.4
T_{3tc}^{2-2}	0	0	0	0	0	0	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2
T_{3tc}^{2-3}	0	0	0	0	0	1	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	16.8	8.3	1.8	4.1	0.1	0.7
T_{3tc}^{2-4}	0	0	0	1	0	0	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	9.4	19.4	-0.4	2.4	-0.4	-0.4
T_{3tc}^{2-5}	0	1	0	1	1	1	0E+00	0E+00	0E+00	0E+00	1E+02	0E+00	12.6	10.1	4.2	2.9	0.4	-0.4
T_{3tc}^{3-1}	0	1	0	1	0	1	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	18.3	21.4	1.8	1.7	1.7	1.7
T_{3tc}^{3-2}	0	10	0	32	1	14	0E+00	0E+00	0E+00	4E+01	0E+00	0E+00	17.1	11.7	1.6	0.3	-1.1	-0.2
T_{3tc}^{3-3}	0	25	0	21	0	115	0E+00	0E+00	0E+00	0E+00	0E+00	1E+01	17.5	17.4	-2.0	1.6	-2.0	-2.0
T_{3tc}^{3-4}	9	61	2990	2155	37	251	2E+02	2E+01	3E+04	1E+04	5E+02	2E+03	16.0	20.2	5.3	6.2	1.1	1.4
T_{3tc}^{3-5}	0	3	1	6	1	3	0E+00	0E+00	0E+00	2E+01	0E+00	0E+00	20.7	20.8	6.3	5.4	1.6	1.6
T_{3tc}^{4-1}	0	45	0	342	0	475	0E+00	0E+00	0E+00	2E+01	0E+00	3E+01	6.4	17.9	3.8	2.0	-0.8	-0.8
T_{3tc}^{4-2}	0	33	4	231	4	34	0E+00	0E+00	0E+00	3E+01	0E+00	0E+00	20.7	21.0	7.2	7.1	2.5	2.1
T_{3tc}^{4-3}	0	1	0	1	0	1	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	14.4	19.0	2.9	-0.4	0.4	-0.4
T_{3tc}^{4-4}	0	3	0	3	1	3	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	20.3	7.8	4.7	5.2	0.6	1.6
T_{3tc}^{4-5}	0	5	1	12	2	8	0E+00	0E+00	0E+00	6E+01	0E+00	0E+00	17.5	16.3	5.0	0.3	0.3	1.1
T_{3tc}^{5-1}	0	1	2	9	7	42	0E+00	0E+00	0E+00	0E+00	8E+01	0E+00	17.5	13.4	-0.4	1.9	-0.4	-0.1
T_{3tc}^{5-2}	0	12	147	-	-	-	0E+00	0E+00	1E+03	6E+03	2E+04	9E+02	16.2	8.6	9.9	-	-	-
T_{3tc}^{5-3}	0	82	217	1014	672	1931	0E+00	0E+00	1E+03	6E+02	7E+03	2E+03	19.2	19.2	4.3	4.3	-0.7	-0.7
T_{3tc}^{5-4}	2	14	210	264	-	905	0E+00	0E+00	2E+03	9E+01	2E+04	2E+03	16.3	11.2	1.4	-2.0	-	-2.0
T_{3tc}^{5-5}	0	113	564	178	38	216	0E+00	5E+01	3E+03	2E+02	2E+02	3E+02	16.6	18.7	3.2	3.9	-0.4	-0.6
T_{3tc}^{6-1}	246	371	433	-	-	3585	2E+03	6E+01	1E+03	4E+03	8E+03	2E+03	18.5	19.9	3.9	-	-	2.2
T_{3tc}^{6-2}	0	2	0	2	0	2	0E+00	0E+00	0E+00	0E+00	0E+00	0E+00	3.5	17.7	3.5	0.3	-0.3	0.3
T_{3tc}^{6-3}	0	3	0	2	0	893	0E+00	0E+00	0E+00	0E+00	0E+00	1E+01	2.3	8.7	2.3	2.1	-1.0	-0.9
T_{3tc}^{6-4}	5	423	6	512	4	874	0E+00	4E+01	0E+00	2E+01	0E+00	9E+02	20.1	13.5	4.5	3.4	1.1	0.1
T_{3tc}^{6-5}	74	1366	1063	-	-	-	5E+02	6E+01	4E+03	2E+03	6E+03	3E+03	18.6	19.1	3.8	-	-	-

Table C.3: Test results for the TE-MSTP(Λ) decision problem: T_{3tc}.

ins.	TE-MSTP dec. prob.		COCMST prob.		ins.	TE-MSTP dec. prob.		COCMST prob.	
	RDM	RDMFM	RDM	RDMFM		RDM	RDMFM	RDM	RDMFM
T_r^{1-1}	0	0	0	0	T_{3tc}^{1-1}	0	0	0	0
T_r^{1-2}	2	1	1	1	T_{3tc}^{1-2}	0	0	0	1
T_r^{1-3}	440	42	62	26	T_{3tc}^{1-3}	0	1	0	1
T_r^{1-4}	0	0	0	0	T_{3tc}^{1-4}	0	1	0	1
T_r^{1-5}	0	0	0	0	T_{3tc}^{1-5}	0	1	0	1
T_r^{2-1}	31	7	9	8	T_{3tc}^{2-1}	0	1	0	1
T_r^{2-2}	0	1	0	1	T_{3tc}^{2-2}	0	0	0	0
T_r^{2-3}	-	-	-	-	T_{3tc}^{2-3}	0	0	0	0
T_r^{2-4}	0	1	0	1	T_{3tc}^{2-4}	0	1	0	1
T_r^{2-5}	0	1	0	1	T_{3tc}^{2-5}	0	1	0	1
T_r^{3-1}	0	0	0	0	T_{3tc}^{3-1}	0	4	0	7
T_r^{3-2}	0	1	0	1	T_{3tc}^{3-2}	0	4	0	4
T_r^{3-3}	166	11	12	12	T_{3tc}^{3-3}	0	7	0	7
T_r^{3-4}	114	9	105	15	T_{3tc}^{3-4}	-	-	-	-
T_r^{3-5}	121	45	25	29	T_{3tc}^{3-5}	0	2	0	2
T_r^{4-1}	36	36	144	19	T_{3tc}^{4-1}	0	10	0	10
T_r^{4-2}	81	17	38	17	T_{3tc}^{4-2}	0	10	0	7
T_r^{4-3}	196	93	314	100	T_{3tc}^{4-3}	0	4	0	6
T_r^{4-4}	23	8	12	9	T_{3tc}^{4-4}	0	1	0	1
T_r^{4-5}	6	0	5	0	T_{3tc}^{4-5}	0	3	0	4
T_r^{5-1}	-	633	3321	1048	T_{3tc}^{5-1}	0	8	0	9
T_r^{5-2}	63	46	1095	25	T_{3tc}^{5-2}	-	-	-	-
T_r^{5-3}	374	33	99	22	T_{3tc}^{5-3}	0	17	0	16
T_r^{5-4}	100	157	88	26	T_{3tc}^{5-4}	-	-	-	-
T_r^{5-5}	2927	507	-	150	T_{3tc}^{5-5}	576	1209	-	1491
T_r^{6-1}	229	416	115	1312	T_{3tc}^{6-1}	-	-	-	-
T_r^{6-2}	279	125	129	41	T_{3tc}^{6-2}	0	16	0	34
T_r^{6-3}	43	118	17	67	T_{3tc}^{6-3}	0	15	0	19
T_r^{6-4}	47	213	1105	224	T_{3tc}^{6-4}	0	11	0	13
T_r^{6-5}	319	701	440	576	T_{3tc}^{6-5}	0	27	0	31
T_r^{7-1}	1042	3108	1767	2880					
T_r^{7-2}	-	-	-	-					
T_r^{7-3}	-	-	-	-					
T_r^{7-4}	-	-	-	-					
T_r^{7-5}	-	-	-	-					
T_r^{8-1}	76	108	450	113					
T_r^{8-2}	51	42	17	43					
T_r^{8-3}	-	-	-	-					
T_r^{8-4}	0	2	0	2					
T_r^{8-5}	-	-	-	-					
T_r^{9-1}	0	10	0	8					
T_r^{9-2}	65	627	13	15					
T_r^{9-3}	95	163	337	53					
T_r^{9-4}	-	1058	-	-					
T_r^{9-5}	8	77	26	36					
T_r^{10-1}	-	-	-	-					
T_r^{10-2}	-	-	-	-					
T_r^{10-3}	-	-	-	-					
T_r^{10-4}	-	-	-	-					
T_r^{10-5}	-	-	-	-					

Table C.4: Running time (s) for the TE-MSTP(-0.05^{ϵ}) decision problem and the COCMST(-0.05^{ϵ}) problems.

D Computational results for the BSA

Key:

- ins. : instance reference;
- MIP-time (s) : time it takes to solve the (mixed-)integer program;
- iter. : number of iterations of BSA;
- B : best upper bound found by RDMFM;
- B_{end} : lower bound obtained after optimization is finished or interrupted;
- B_{io} : lower bound obtained at the end of the in-out cut loop stabilization;
- UB : upper bound at the end of BSA;
- LB : lower bound at the end of BSA.

ins.	RDMFM		BSA		
	MIP-time (s)	$B - B_{end}$	time (s)	$UB - LB$	iter.
T_r^{1-1}	1	0.00	1	0.01	3
T_r^{1-2}	3	0.00	1	0.01	4
T_r^{1-3}	93	0.00	106	0.01	5
T_r^{1-4}	1	0.00	1	0.01	5
T_r^{1-5}	0	0.00	1	0.01	3
T_r^{2-1}	76	0.00	30	0.01	6
T_r^{2-2}	3	0.00	2	0.00	5
T_r^{2-3}	-	0.03	-	0.04	5
T_r^{2-4}	2	0.01	6	0.01	5
T_r^{2-5}	1	0.00	1	0.00	1
T_r^{3-1}	14	0.00	3	0.01	5
T_r^{3-2}	2	0.00	1	0.01	5
T_r^{3-3}	30	0.00	66	0.01	7
T_r^{3-4}	36	0.00	15	0.01	5
T_r^{3-5}	44	0.00	67	0.01	6
T_r^{4-1}	86	0.00	226	0.01	4
T_r^{4-2}	39	0.00	68	0.01	5
T_r^{4-3}	291	0.00	80	0.01	4
T_r^{4-4}	20	0.00	29	0.01	6
T_r^{4-5}	16	0.00	16	0.01	4
T_r^{5-1}	-	0.09	-	0.01	6
T_r^{5-2}	58	0.00	229	0.01	5
T_r^{5-3}	69	0.00	134	0.01	5
T_r^{5-4}	126	0.00	73	0.01	5
T_r^{5-5}	434	0.00	1591	0.01	5
T_r^{6-1}	130	0.00	92	0.01	6
T_r^{6-2}	240	0.00	360	0.01	5
T_r^{6-3}	106	0.00	133	0.01	6
T_r^{6-4}	939	0.00	582	0.01	5
T_r^{6-5}	295	0.00	130	0.01	5
T_r^{7-1}	133	0.00	-	0.01	5
T_r^{7-2}	-	0.13	-	0.14	3
T_r^{7-3}	-	0.09	-	0.16	3
T_r^{7-4}	-	0.11	-	0.14	3
T_r^{7-5}	-	0.16	-	0.14	3
T_r^{8-1}	58	0.00	41	0.00	6
T_r^{8-2}	140	0.00	67	0.01	6
T_r^{8-3}	-	0.29	-	0.19	2
T_r^{8-4}	7	0.00	1	0.00	3
T_r^{8-5}	-	0.35	-	0.29	2
T_r^{9-1}	94	0.01	251	0.01	2
T_r^{9-2}	574	0.01	1019	0.01	4
T_r^{9-3}	188	0.01	142	0.00	5
T_r^{9-4}	935	0.00	-	0.01	5
T_r^{9-5}	82	0.01	169	0.00	5
T_r^{10-1}	-	0.11	3395	0.01	5
T_r^{10-2}	-	0.12	-	0.09	4
T_r^{10-3}	-	0.20	-	0.29	2
T_r^{10-4}	-	0.21	-	0.01	7
T_r^{10-5}	-	0.33	-	0.08	4

Table D.1: Test results for the BSA: T_{rand}.

ins.	RDMFM		BSA		
	MIP-time (s)	$B - B_{end}$	time (s)	$UB - LB$	iter.
T_{3tc}^{1-1}	0	0.00	1	0.00	2
T_{3tc}^{1-2}	1	0.00	2	0.00	1
T_{3tc}^{1-3}	1	0.00	2	0.01	1
T_{3tc}^{1-4}	1	0.00	2	0.00	1
T_{3tc}^{1-5}	2	0.00	2	0.00	3
T_{3tc}^{2-1}	3	0.00	4	0.01	3
T_{3tc}^{2-2}	1	0.00	0	0.00	0
T_{3tc}^{2-3}	1	0.00	1	0.01	2
T_{3tc}^{2-4}	1	0.00	2	0.00	1
T_{3tc}^{2-5}	1	0.00	2	0.01	2
T_{3tc}^{3-1}	41	0.00	18	0.00	3
T_{3tc}^{3-2}	49	0.00	13	0.01	4
T_{3tc}^{3-3}	49	0.00	17	0.00	4
T_{3tc}^{3-4}	-	0.01	-	0.01	5
T_{3tc}^{3-5}	10	0.00	9	0.00	4
T_{3tc}^{4-1}	140	0.00	17	0.00	4
T_{3tc}^{4-2}	72	0.00	19	0.00	3
T_{3tc}^{4-3}	4	0.00	9	0.00	3
T_{3tc}^{4-4}	3	0.00	6	0.00	4
T_{3tc}^{4-5}	14	0.00	18	0.00	4
T_{3tc}^{5-1}	166	0.00	31	0.00	3
T_{3tc}^{5-2}	-	0.04	-	0.01	5
T_{3tc}^{5-3}	408	0.00	75	0.00	4
T_{3tc}^{5-4}	-	0.05	2320	0.01	5
T_{3tc}^{5-5}	2521	0.00	451	0.01	6
T_{3tc}^{6-1}	-	0.01	-	0.01	5
T_{3tc}^{6-2}	265	0.00	110	0.00	2
T_{3tc}^{6-3}	292	0.00	92	0.00	3
T_{3tc}^{6-4}	196	0.00	116	0.01	4
T_{3tc}^{6-5}	1846	0.00	233	0.01	5

Table D.2: Test results for the BSA: T_{3tc}.

E Computational results for the PUMF problem

Key:

- ins. : instance reference;
- LP-time (s) : time it takes to solve the LP relaxation;
- MIP-time (s) : time it takes to solve the (mixed-)integer program;
- B^* : best upper bound found by any of the models;
- B : best upper bound found by each model;
- B_{LP} : lower bound obtained by solving the LP relaxation;
- B_0 : lower bound obtained after the root node of the B&B tree has been processed;
- B_{end} : lower bound obtained after optimization is finished or interrupted;
- $Gap_{LP}(\%)$: gap between B^* and B_{LP};
- $Gap_0(\%)$: gap between B^* and B_0;
- $Gap_{end}(\%)$: gap between B and B_{end};
- B&B Tree Nodes : number of nodes in the B&B enumeration tree,

ins.	B^*	LP-time (s)			MIP-time (s)			B			$Gap_{LP}(\%)$			Gap_0 (%)			Gap_{end} (%)			B&B Tree Nodes		
		BM1	BM2	SM	BM1	BM2	SM	BM1	BM2	SM	BM1	BM2	SM	BM1	BM2	SM	BM1	BM2	SM	BM1	BM2	SM
T_1^{1-1}	523	17	13.8	0.8	18	13.8	4	523	523	523	5.1	5.1	0.0	4.5	5.1	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{1-2}	532	15	10.7	0.8	15	10.7	4	532	532	532	2.6	2.6	0.0	2.6	2.6	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{1-3}	564	17.2	13.4	1.1	19	13.4	4	564	564	564	5.9	5.9	0.0	4.3	5.9	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{1-4}	707	16.3	13.2	0.8	18	13.2	4	707	707	707	6.0	6.0	0.0	4.2	6.0	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{1-5}	472	15.7	12.6	0.8	17	12.6	4	472	472	472	6.6	6.6	0.0	3.1	6.6	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{2-1}	609	21.7	15.2	1.1	22	15.2	5	609	609	609	3.7	3.7	0.0	3.3	3.7	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{2-2}	702	20.1	15.2	1	26	15.2	5	702	702	702	10.8	10.8	0.0	4.9	4.9	0.0	0.0	0.0	0.0	3E+01	2E+01	0.0E+00
T_1^{2-3}	645	20.9	17.3	1	23	17.3	5	645	645	645	9.2	9.2	0.0	8.9	9.2	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{2-4}	546	21.7	17.5	1	23	17.5	5	546	546	546	5.4	5.4	0.0	4.3	5.4	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{2-5}	486	20.3	15.4	1	22	15.4	5	486	486	486	2.3	2.3	0.0	2.3	2.3	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{3-1}	723	32.6	29.4	1.4	37	29.4	7	723	723	723	3.1	3.1	0.0	2.0	3.1	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{3-2}	647	26.6	23.2	1.4	30	23.2	7	647	647	647	4.1	4.1	0.0	3.2	4.1	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{3-3}	799	28.6	20.4	1.3	40	20.4	7	799	799	799	12.9	12.9	0.0	6.4	6.4	0.0	0.0	0.0	0.0	5E+01	4E+01	0.0E+00
T_1^{3-4}	658	25.7	18.6	1.4	29	18.6	7	658	658	658	1.8	1.8	0.0	1.1	1.8	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{3-5}	800	35	29.9	1.3	38	29.9	7	800	800	800	2.7	2.7	0.0	1.5	2.7	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{4-1}	2415	10.2	11.8	0.9	13	11.8	7	2415	2415	2415	2.3	2.3	0.0	2.3	0.0	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{4-2}	2271	11.1	12.4	0.9	13	12.4	7	2271	2271	2271	0.5	0.5	0.0	0.5	0.5	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{4-3}	2716	10	10.7	0.9	21	10.7	12	2716	2716	2716	17.4	17.4	1.5	9.6	5.5	1.2	0.0	0.0	0.0	1E+01	2E+01	0.0E+00
T_1^{4-4}	2078	10.3	10.2	0.9	13	10.2	7	2078	2078	2078	3.8	3.8	0.0	0.7	0.6	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{4-5}	2407	9.9	13.4	0.9	13	13.4	7	2407	2407	2407	2.2	2.2	0.0	1.3	2.2	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{5-1}	2067	259.8	280.5	6.8	576	280.5	89	2067	2067	2067	12.9	12.9	0.0	3.6	3.6	0.0	0.0	0.0	0.0	2E+02	4E+02	0.0E+00
T_1^{5-2}	1870	347.6	301.4	6.6	417	301.4	93	1870	1870	1870	6.0	6.0	0.0	2.5	6.0	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{5-3}	1811	249.1	259.9	6.6	479	259.9	89	1811	1811	1811	4.8	4.8	0.0	0.8	0.8	0.0	0.0	0.0	0.0	4E+01	5E+01	0.0E+00
T_1^{5-4}	1728	269.2	212.5	6.5	375	212.5	93	1728	1728	1728	3.5	3.5	0.0	1.1	3.5	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{5-5}	1753	270.9	222.2	6.9	383	222.2	94	1753	1753	1753	7.6	7.6	0.0	1.3	1.3	0.0	0.0	0.0	0.0	6E+01	6E+01	0.0E+00
T_1^{6-1}	1354	337.8	353.5	6.1	2278	353.5	78	1354	1362	1354	9.2	9.2	0.0	2.4	2.4	0.0	0.0	?	0.0	2E+03	6E+02	0.0E+00
T_1^{6-2}	1303	407.8	416.2	6.3	-	416.2	70	1303	1303	1303	12.7	12.7	0.0	6.2	6.2	0.0	?	?	0.0	1E+03	8E+01	0.0E+00
T_1^{6-3}	1264	355.9	399.9	5.9	535	399.9	70	1264	1281	1264	7.9	7.9	0.0	4.6	4.6	0.0	0.0	?	0.0	8E+01	2E+01	0.0E+00
T_1^{6-4}	856	319.5	360.5	6.2	399	360.5	72	856	856	856	1.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{6-5}	1585	326.4	332.7	5.9	819	332.7	70	1585	1585	1585	15.4	15.4	0.0	7.4	7.4	0.0	0.0	?	0.0	2E+03	2E+01	0.0E+00

Table E.1: Test results for the PUMF problem: $T_1^1 - T_1^6$.

ins.	B^*	LP-time (s)			MIP-time (s)			B			Gap_{LP}(%)			Gap_0 (%)			Gap_{end} (%)			B&B Tree Nodes		
		BM1	BM2	SM	BM1	BM2	SM	BM1	BM2	SM	BM1	BM2	SM	BM1	BM2	SM	BM1	BM2	SM	BM1	BM2	SM
T_1^{7-1}	1340	417.2	396.5	8.3	652	396.5	109	1340	1340	1340	5.7	5.7	0.0	2.6	2.6	0.0	0.0	0.0	0.0	8E+01	8E+01	0.0E+00
T_1^{7-2}	1563	327.8	422	8.2	881	422	113	1563	1563	1563	5.1	5.1	0.0	0.7	0.7	0.0	0.0	0.0	0.0	6E+01	5E+01	0.0E+00
T_1^{7-3}	1597	324	236.6	8.3	-	236.6	112	1597	1614	1597	8.9	8.9	0.0	1.5	1.5	0.0	0.0	?	0.0	2E+03	4E+02	0.0E+00
T_1^{7-4}	1445	353	326.2	8.1	438	326.2	110	1445	1445	1445	6.1	6.1	0.0	1.0	6.1	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{7-5}	1533	364.8	271.6	7.3	532	271.6	117	1533	1533	1533	5.7	5.7	0.0	0.7	0.7	0.0	0.0	0.0	0.0	7E+01	7E+01	0.0E+00
T_1^{8-1}	2029	197.5	223.6	4.7	333	223.6	74	2029	2029	2029	7.6	7.6	0.1	2.1	2.1	0.1	0.0	0.0	0.0	4E+01	6E+01	0.0E+00
T_1^{8-2}	1646	207	194.6	4.8	279	194.6	67	1646	1646	1646	5.7	5.7	0.0	0.9	5.7	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{8-3}	1896	207	180.2	4.8	465	180.2	70	1896	1896	1896	8.2	8.2	0.1	1.3	1.3	0.1	0.0	0.0	0.0	9E+01	5E+01	0.0E+00
T_1^{8-4}	1754	201.3	208.8	4.9	1012	208.8	67	1754	1754	1754	14.6	14.6	0.0	8.6	5.9	0.0	0.0	0.0	0.0	1E+03	8E+02	0.0E+00
T_1^{8-5}	1650	214.1	193.8	4.8	375	193.8	66	1650	1650	1650	6.4	6.4	0.0	0.8	0.8	0.0	0.0	0.0	0.0	2E+01	1E+01	0.0E+00
T_1^{9-1}	2285	273.4	274.4	6	533	274.4	90	2285	2285	2285	9.4	9.4	0.0	1.2	2.0	0.0	0.0	0.0	0.0	5E+01	8E+01	0.0E+00
T_1^{9-2}	2335	284.8	274.5	6	2319	274.5	95	2335	2335	2335	18.5	18.5	0.0	7.4	8.4	0.0	0.0	?	0.0	6E+02	5E+02	0.0E+00
T_1^{9-3}	2065	293	274.3	5.9	462	274.3	86	2065	2065	2065	5.8	5.8	0.0	1.1	1.1	0.0	0.0	0.0	0.0	4E+01	4E+01	0.0E+00
T_1^{9-4}	2105	282.2	284	6.2	547	284	86	2105	2105	2105	4.0	4.0	0.0	1.6	4.0	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{9-5}	1710	264.8	257.9	6	509	257.9	92	1710	1710	1710	10.3	10.3	0.0	2.5	1.9	0.0	0.0	0.0	0.0	6E+01	7E+01	0.0E+00
T_1^{10-1}	2045	487.7	549.9	8.6	809	549.9	128	2045	2045	2045	8.0	8.0	0.0	1.2	1.2	0.0	0.0	0.0	0.0	1E+02	1E+02	0.0E+00
T_1^{10-2}	1829	489	587	8.8	999	587	118	1829	1829	1829	5.9	5.9	0.0	1.3	5.9	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{10-3}	1926	484.7	511.7	9.1	-	511.7	128	1926	1963	1926	12.9	12.9	0.2	4.5	4.5	0.2	?	?	0.0	1E+03	0E+00	0.0E+00
T_1^{10-4}	1365	489.2	394.8	8.8	665	394.8	121	1365	1365	1365	4.8	4.8	0.0	2.5	4.8	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{10-5}	1579	466.2	528.5	9.1	749	528.5	120	1579	1579	1579	4.6	4.6	0.0	1.4	4.6	0.0	0.0	0.0	0.0	0E+00	0E+00	0.0E+00
T_1^{11-1}	3331	473.8	435.3	9.5	770	435.3	99	3331	3331	3331	7.4	7.4	0.0	0.5	0.6	0.0	0.0	0.0	0.0	9E+00	3E+01	0.0E+00
T_1^{11-2}	3521	422	447.2	9	3037	447.2	89	3521	3521	3521	8.8	8.8	0.0	3.7	3.7	0.0	0.0	0.0	0.0	7E+02	4E+02	0.0E+00
T_1^{11-3}	2955	411.9	344	8.7	647	344	88	2955	2955	2955	6.6	6.6	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1E+01	1E+01	0.0E+00
T_1^{11-4}	2251	451.2	455.7	8.6	1002	455.7	87	2251	2251	2251	6.9	6.9	0.0	2.6	2.1	0.0	0.0	0.0	0.0	3E+02	3E+02	0.0E+00
T_1^{11-5}	2701	440.9	432.3	9.1	714	432.3	97	2701	2701	2701	6.6	6.6	0.0	0.8	1.5	0.0	0.0	0.0	0.0	5E+01	5E+01	0.0E+00

Table E.2: Test results for the PUMF problem: $T_1^7 - T_1^{11}$.

ins.	B^*	LP-time (s)			MIP-time (s)			B			Gap_{LP}(%)			Gap_0 (%)			Gap_{end} (%)			B&B Tree Nodes		
		BM1	BM2	SM	BM1	BM2	SM	BM1	BM2	SM	BM1	BM2	SM	BM1	BM2	SM	BM1	BM2	SM	BM1	BM2	SM
atlanta	4.29E+08	0.2	0.1	0.6	0	0	3	4.29E+08	4.29E+08	4.29E+08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0E+00	0.0E+00	0E+00
atlanta-0.38	3.19E+05	0.1	0.1	0.3	4	2	17	3.19E+05	3.19E+05	3.19E+05	0.3	0.3	0.3	0.0	0.2	0.3	0.0	0.0	0.0	1.9E+02	7.3E+02	7E+02
atlanta-0.71	6.69E+05	0.1	0.1	0.6	4	1	58	6.69E+05	6.69E+05	6.69E+05	0.5	0.5	0.5	0.2	0.2	0.5	0.0	0.0	0.0	5.1E+02	2.2E+02	1E+03
atlanta-1.05	2.44E+07	0.2	0.1	0.9	15	6	92	2.44E+07	2.44E+07	2.44E+07	0.9	0.9	0.9	0.3	0.6	0.6	0.0	0.0	0.0	1.8E+04	5.2E+02	2E+03
france	2.72E+08	0.7	1.1	3.3	719	418	2141	2.72E+08	2.72E+08		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.9E+05	1.3E+05	2E+04
france-0.38	4.59E+05	0.5	0.4	1.5	-	1403	-	4.59E+05	4.59E+05	4.59E+05	0.1	0.1	0.1	0.1	0.1	0.1	?	0.0	0.0	1.4E+06	3.1E+05	1E+05
france-0.71	5.11E+07	0.7	0.6	4.2	-	-	-	5.11E+07	5.11E+07	5.19E+07	0.1	0.1	0.1	0.1	0.1	0.1	?	?	1.5	5.5E+05	5.4E+05	2E+04
france-1.05	2.36E+08	0.7	0.5	3.3	-	1213	-	2.36E+08	2.36E+08	2.36E+08	0.0	0.0	0.0	0.0	0.0	0.0	?	0.0	0.1	1.2E+06	3.2E+05	5E+04
newyork	2.85E+03	0.2	0.5	0.2	1	1	4	2.85E+03	2.85E+03	2.85E+03	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0E+00	0.0E+00	0E+00
newyork-0.38	4.22E+03	0.4	1.0	2.0	-	97	312	4.22E+03	4.22E+03	4.22E+03	3.6	3.6	2.9	0.7	3.0	2.3	?	0.0	0.0	1.5E+05	3.9E+03	2E+03
newyork-0.71	3.69E+04	0.8	1.1	3.7	-	-	-	3.69E+04	3.69E+04	2.89E+06	6.1	6.1	1.3	4.4	5.4	1.0	?	?	2.2	2.0E+05	1.2E+05	7E+03
newyork-1.05	1.92E+06	1.2	1.4	4.0	-	3534	-	1.92E+06	1.92E+06	1.97E+06	5.8	5.8	0.1	5.4	0.2	0.1	?	0.0	3.0	2.2E+05	1.2E+05	9E+03
pdh	1.92E+07	0.1	0.0	0.0	0	2	0	1.92E+07	1.92E+07	1.92E+07	7.6	7.6	0.0	1.5	2.8	0.0	0.0	0.0	0.0	2.2E+02	5.6E+01	0E+00
pdh-0.38	7.42E+03	0.1	0.0	0.1	0	0	0	7.42E+03	7.42E+03	7.42E+03	19.2	19.2	0.0	3.1	4.0	0.0	0.0	0.0	0.0	7.0E+00	0.0E+00	0E+00
pdh-0.71	8.47E+05	0.0	0.0	0.0	5	3	0	8.47E+05	8.47E+05	8.47E+05	98.9	98.9	0.0	0.6	1.2	0.0	0.0	0.0	0.0	1.9E+03	1.6E+03	0E+00
pdh-1.05	3.87E+06	0.1	0.0	0.0	40	7	0	3.87E+06	3.87E+06	3.87E+06	99.5	99.5	0.0	9.5	11.6	0.0	0.0	0.0	0.0	5.0E+02	1.1E+03	0E+00
sun	4.49E+04	0.4	0.3	0.5	977	67	1190	4.49E+04	4.49E+04	4.49E+04	0.2	0.2	0.2	0.1	0.1	0.1	0.0	0.0	0.0	7.7E+04	6.7E+03	1E+04
sun-0.38	3.37E+03	0.3	0.3	0.4	12	16	53	3.38E+03	3.38E+03	3.38E+03	1.4	1.4	1.4	0.4	0.8	0.9	0.0	0.0	0.0	1.7E+03	8.0E+02	1E+03
sun-0.71	4.32E+05	0.3	0.4	0.8	291	17	738	4.32E+05	4.32E+05	4.32E+05	0.4	0.4	0.4	0.1	0.1	0.3	0.0	0.0	0.0	5.4E+05	1.2E+03	9E+03
sun-1.05	1.51E+06	0.5	0.4	0.5	-	47	315	1.51E+06	1.51E+06	1.51E+06	1.8	1.8	1.8	0.9	1.6	1.7	?	0.0	0.0	1.4E+06	8.6E+03	4E+03
ta1	1.66E+07	0.9	0.9	0.8	31	26	208	1.66E+07	1.66E+07	1.66E+07	3.1	3.1	0.0	0.6	2.1	0.0	0.0	0.0	0.0	1.3E+03	1.2E+03	1E+04
ta1-0.38	5.22E+08	1.0	1.1	2.0	593	218	761	5.22E+08	5.22E+08	5.22E+08	95.1	95.1	0.1	88.1	95.0	0.1	0.0	0.0	0.0	7.6E+03	2.0E+03	5E+03
ta1-0.71	6.02E+09	1.3	1.4	3.7	-	40	-	6.02E+09	6.73E+09	6.02E+09	46.6	46.6	1.2	36.1	46.6	1.1	?	0.0	1.2	7.8E+04	8.1E+02	5E+03
ta1-1.05	1.57E+10	1.2	1.7	3.0	421	591	-	1.57E+10	1.57E+10	1.59E+10	6.0	6.0	0.0	0.6	6.0	0.0	0.0	0.0	1.0	1.0E+04	9.3E+03	1E+04

Table E.3: Test results for the PUMF problem: T_2.

F Computational results for the NPUMF problem

Key:

- ins. : instance reference;
- LP-time (s) : time it takes to solve the LP relaxation;
- MIP-time (s) : time it takes to solve the (mixed-)integer program;
- B^* : best upper bound found by any of the models;
- B : best upper bound found by each model;
- B_{LP} : lower bound obtained by solving the LP relaxation;
- B_0 : lower bound obtained after the root node of the B&B tree has been processed;
- B_{end} : lower bound obtained after optimization is finished or interrupted;
- $Gap_{LP}(\%)$: gap between B^* and B_{LP};
- $Gap_0(\%)$: gap between B^* and B_0;
- $Gap_{end}(\%)$: gap between B and B_{end};
- B&B Tree Nodes : number of nodes in the B&B enumeration tree,

ins.	B^*	LP-time (s)		MIP-time (s)		B		Gap_{LP}(%)				Gap_0 (%)		Gap_{end} (%)		B&B Tree Nodes	
		BM1	SM-n	BM1	SM-n	BM1	SM-n	BM1	SM-n	BM1-r	SM-r	BM1	SM-n	BM1	SM-n	BM1	SM-n
$1-c4$	10207	0.1	1.8	14	32	10207	10207	25.1	3.7	7646.260605	2	1.1	0.8	0.0	0.0	1E+03	4E+02
$1-c8$	9597	0.1	2.9	41	75	9597	9597	30.1	3.5	6705.183813	2	1.2	1.3	0.0	0.0	2E+03	6E+02
$1-\not\in 4$	10177	0.1	1.7	3	10	10177	10177	21.5	3.8	7984.529557	1	0.7	0.7	0.0	0.0	1E+02	3E+01
$1-\not\in 8$	10079	0.1	2.2	4	17	10079	10079	19.8	4.1	8083.445501	2	1.1	0.8	0.0	0.0	1E+02	8E+01
$2-c4$	12883	0.1	1.7	47	83	12883	12883	27.7	8.2	9313.575426	3	1.3	1.0	0.0	0.0	2E+03	6E+02
$2-c8$	12470	0.1	2.6	595	-	12470	12498	34.2	9.3	8211.444385	4	2.8	2.5	0.0	??	2E+04	9E+03
$2-\not\in 4$	13641	0.1	1.8	33	127	13641	13641	22.3	8.6	10598.81988	3	2.2	1.8	0.0	0.0	1E+03	7E+02
$2-\not\in 8$	14255	0.1	2.2	84	431	14255	14255	22.0	11.1	11125.52301	4	2.9	3.6	0.0	0.0	4E+03	2E+03
$3-c4$	20328	0.3	5.3	-	-	20328	20371	27.2	4.8	14807.3984	2	3.1	2.5	??	??	4E+04	9E+03
$3-c8$	19591	0.3	28.5	-	-	19591	19864	31.8	5.3	13354.08852	2	3.7	3.1	??	??	4E+04	8E+02
$3-\not\in 4$	21104	0.4	4.6	969	-	21104	21137	21.1	4.3	16653.3312	2	2.1	1.9	0.0	??	1E+04	2E+03
$3-\not\in 8$	20572	0.4	12.5	-	-	20572	20675	19.3	5.2	16597.06817	2	3.0	2.6	??	??	4E+04	1E+03
$4-c4$	21355	0.6	8.4	503	1566	21355	21355	24.4	4.9	16138.54498	2	1.6	1.1	0.0	0.0	8E+03	5E+03
$4-c8$	20566	0.5	29.6	-	-	20566	20850	30.8	4.5	14230.18649	2	2.5	1.8	??	??	2E+04	1E+03
$4-\not\in 4$	21415	0.5	6.9	276	2546	21415	21415	19.8	4.3	17177.88578	2	1.3	1.3	0.0	0.0	6E+03	3E+03
$4-\not\in 8$	21418	0.5	15.3	404	-	21418	21504	20.4	4.1	17046.06578	2	1.4	1.3	0.0	??	8E+03	1E+03
$5-c4$	16915	0.6	5.4	-	-	17005	16915	25.7	5.9	12567.39921	3	2.8	2.4	??	??	3E+04	1E+04
$5-c8$	16392	0.6	27.4	-	-	16392	16638	30.7	6.5	11359.71712	3	4.1	3.5	??	??	3E+04	1E+03
$5-\not\in 4$	16963	0.6	6.7	-	-	16963	16967	21.5	6.1	13322.88494	3	2.8	2.6	??	??	4E+04	7E+03
$5-\not\in 8$	16728	0.6	13.7	-	-	16728	16803	20.6	5.6	13283.35776	3	2.7	2.7	??	??	4E+04	1E+03
$6-c4$	16861	2.9	56.1	-	-	16861	17252	31.3	4.0	11589.92552	3	2.9	2.8	??	??	9E+03	5E+02
$6-c8$	16623	2.8	134.1	-	-	16623	17783	36.3	4.9	10591.75126	4	4.3	4.1	??	??	7E+03	2E+02
$6-\not\in 4$	16624	3.8	31.5	-	-	16624	16852	28.4	3.3	11900.04989	2	2.3	2.2	??	??	2E+04	9E+02
$6-\not\in 8$	16791	2.9	65.3	-	-	16791	16835	27.6	3.5	12161.67142	2	2.3	2.3	??	??	2E+04	5E+02
$7-c4$	33419	6.5	149.8	-	-	33419	-	23.5	2.6	25555.81873	1	2.6	2.4	??	??	2E+03	5E+01
$7-c8$	34614	7.0	662.7	-	-	34614	-	33.5	9.0	23020.37038	9	9.4	-	??	??	7E+02	0E+00
$7-\not\in 4$	33878	8.3	173.5	-	-	33878	-	20.4	3.0	26966.92622	2	2.9	2.0	??	??	3E+03	3E+01
$7-\not\in 8$	34201	7.0	345.7	-	-	34201	-	19.8	2.3	27421.72417	2	2.3	-	??	??	3E+03	0E+00
$8-c4$	31156	6.9	344.9	-	-	31156	-	29.7	4.9	21895.99737	5	4.9	4.9	??	??	2E+03	0E+00
$8-c8$	30695	7.8	1187.0	-	-	30695	-	33.8	6.7	20307.89458	7	6.8	-	??	??	1E+03	0E+00
$8-\not\in 4$	30910	7.1	219.9	-	-	30910	-	27.2	4.3	22497.90858	4	4.2	4.0	??	??	2E+03	3E+01
$8-\not\in 8$	31187	6.9	568.0	-	-	31187	-	27.7	5.4	22536.0502	3	5.4	-	??	??	2E+03	0E+00
$9-c4$	-	24.6	299.8	-	-	-	-	-	-	26197.44823	10	-	-	??	??	1E+03	0E+00
$9-c8$	-	24.4	1267.5	-	-	-	-	-	-	24158.29878	-	-	-	??	??	8E+01	0E+00
$9-\not\in 4$	37863	28.0	266.3	-	-	37863	-	26.3	5.8	27906.12901	5	5.7	-	??	??	6E+02	0E+00
$9-\not\in 8$	-	18.4	604.4	-	-	-	-	-	-	27387.97002	7	-	-	??	??	2E+03	0E+00
$10-c4$	27568	38.7	752.9	-	-	27568	-	-	15.0	15675.77476	15	15.1	-	??	??	5E+02	0E+00
$10-c8$	33371	41.0	-	-	-	33371	-	55.8	-	14746.14173	-	31.1	-	??	??	0E+00	0E+00
$10-\not\in 4$	26793	38.9	330.6	-	-	26793	-	40.2	12.3	16029.92789	9	12.2	-	??	??	5E+02	0E+00
$10-\not\in 8$	26540	41.0	665.3	-	-	26540	-	39.9	12.2	15942.07923	12	12.1	-	??	??	5E+02	0E+00

Table F.1: Test results for the NPUMF problem.

G Computational results for the TE-MSTP-p problem

Key:

- ins. : instance reference;
- LP-time (s) : time it takes to solve the LP relaxation;
- MIP-time (s) : time it takes to solve the (mixed-)integer program;
- B&C-time (s): time it takes to solve the B&C algorithm;
- IO-time (s): time it takes the in-out cut loop stabilization to be processed;
- B^* : best upper bound found by any of the models;
- B : best upper bound found by each model;
- B_{LP} : lower bound obtained by solving the LP relaxation;
- B_0 : lower bound obtained after the root node of the B&B tree has been processed;
- B_{end} : lower bound obtained after optimization is finished or interrupted;
- B_{io} : lower bound obtained at the end of the in-out cut loop stabilization;
- U^* : worst-case edge utilization obtained by solving the TE-MSTP-p problem to optimality;
- $Gap_{LP}(\%)$: gap between B^* and B_{LP};
- $Gap_0(\%)$: gap between B^* and B_0;
- $Gap_{end}(\%)$: gap between B and B_{end};
- $Gap_{io}(\%)$: gap between B^* and B_{io}.
- $Gap^*_U(\%)$: gap between B^* and U^*;
- B&B Tree Nodes : number of nodes in the B&B enumeration tree;
- # User cuts: number of Benders' cuts generated as user cuts (see Section 2.5.2 for more details);
- # Lazy cons.: number of Benders' cuts generated as user cuts (see Section 2.5.2 for more details);

ins.	B^*	LP-time (s)		MIP-time (s)		B		Gap_{LP}(%)		Gap_0 (%)		Gap_{end} (%)		B&B Tree Nodes		Gap^*_U (%)	
		B	S	B	S	B	S	B	S	B	S	B	S	B	S	B	S
T_r^{1-1}	238	0.0	0.1	0	1	238	238	7.8	7.6	7.8	7.6	0.0	0.0	4E+01	5E+01	25	14
T_r^{1-2}	271	0.1	0.1	1	1	271	271	6.2	5.3	6.2	5.3	0.0	0.0	4E+01	9E+01	25	25
T_r^{1-3}	193	0.5	2.6	81	142	193	193	24.4	19.9	24.4	19.9	0.0	0.0	7E+02	2E+03	33	33
T_r^{1-4}	641	0.1	0.2	0	1	641	641	17.6	17.6	17.6	17.6	0.0	0.0	5E+01	2E+02	1	1
T_r^{1-5}	116	0.0	0.1	0	0	116	116	2.5	2.5	2.5	2.5	0.0	0.0	1E+01	2E+01	23	23
T_r^{2-1}	1687	0.3	0.6	2	79	1688	1687	3.5	3.5	3.5	3.5	0.0	0.0	6E+01	2E+03	8	8
T_r^{2-2}	3922	0.3	0.6	2	6	3922	3922	6.6	6.6	6.6	6.6	0.0	0.0	6E+01	2E+02	7	7
T_r^{2-3}	959	2.9	2.9	-	-	959	-	13.2	13.2	13.2	13.2	3.6	-	2E+04	1E+04	6	-
T_r^{2-4}	4775	0.2	0.5	3	57	4775	4775	5.4	5.4	5.4	5.4	0.0	0.0	2E+02	2E+03	8	10
T_r^{2-5}	8111	0.3	0.6	1	16	8111	8111	0.8	0.8	0.8	0.8	0.0	0.0	2E+01	3E+02	0	0
T_r^{3-1}	276	0.3	1.0	16	43	276	276	16.5	13.5	16.5	13.5	0.0	0.0	5E+02	1E+03	13	13
T_r^{3-2}	302	0.5	0.9	11	30	302	302	7.6	6.8	7.6	6.8	0.0	0.0	2E+02	6E+02	0	0
T_r^{3-3}	434	0.5	0.5	35	226	434	434	13.4	12.3	13.4	12.3	0.0	0.0	8E+02	5E+03	28	28
T_r^{3-4}	240	0.3	1.1	13	23	240	240	14.0	11.4	14.0	11.4	0.0	0.0	4E+02	7E+02	22	22
T_r^{3-5}	305	0.3	0.5	5	8	305	305	7.5	6.4	7.5	6.4	0.0	0.0	1E+02	2E+02	24	24
T_r^{4-1}	159	0.3	1.5	228	45	159	159	29.9	14.9	29.9	14.9	0.0	0.0	7E+03	1E+03	27	27
T_r^{4-2}	325	0.6	0.9	150	99	325	325	42.3	13.2	42.3	13.2	0.0	0.0	2E+03	1E+03	30	30
T_r^{4-3}	162	0.3	1.6	61	12	162	162	27.5	7.3	27.5	7.3	0.0	0.0	1E+03	2E+02	0	0
T_r^{4-4}	219	0.3	1.3	64	19	219	219	41.5	17.6	41.5	17.6	0.0	0.0	1E+03	3E+02	25	25
T_r^{4-5}	60	0.1	0.3	0	1	60	60	9.4	9.4	9.4	9.4	0.0	0.0	1E+01	2E+01	0	0
T_r^{5-1}	436	0.9	2.0	140	23	436	436	11.9	5.7	11.9	5.7	0.0	0.0	1E+03	7E+01	13	13
T_r^{5-2}	240	0.5	1.1	625	609	240	240	29.8	23.9	29.8	23.9	0.0	0.0	1E+04	8E+03	5	5
T_r^{5-3}	123	0.3	0.5	8	21	123	123	13.0	12.1	13.0	12.1	0.0	0.0	1E+02	4E+02	35	35
T_r^{5-4}	149	0.5	0.6	55	153	149	149	17.7	16.0	17.7	16.0	0.0	0.0	1E+03	2E+03	44	44
T_r^{5-5}	313	1.2	1.8	403	229	313	313	18.8	13.3	18.8	13.3	0.0	0.0	3E+03	2E+03	31	31

Table G.1: Test results for the TE-MSTP-p problem: $T^1_{rand} - T^5_{rand}$.

ins.	B^*	LP-time (s)		MIP-time (s)		B		$Gap_{LP}(\%)$		Gap_0 (%)		Gap_{end} (%)		B&B Tree Nodes		Gap^*_U (%)	
		B	S	B	S	B	S	B	S	B	S	B	S	B	S	B	S
T_r^{6-1}	333	1.1	3.6	1117	242	333	333	33.3	16.6	33.3	16.6	0.0	0.0	1E+04	2E+03	21	21
T_r^{6-2}	644	1.3	2.1	568	774	644	644	37.4	24.5	37.4	24.5	0.0	0.0	4E+03	7E+03	21	21
T_r^{6-3}	598	1.1	3.2	352	-	598	605	41.8	30.2	41.8	30.2	0.0	8.1	4E+03	3E+04	6	6
T_r^{6-4}	193	0.7	1.9	177	112	193	193	29.9	19.1	29.9	19.1	0.0	0.0	3E+03	2E+03	13	13
T_r^{6-5}	353	0.9	2.4	239	794	353	353	37.6	24.5	37.6	24.5	0.0	0.0	3E+03	8E+03	17	17
T_r^{7-1}	1206	3.7	6.5	1394	-	1206	-	19.8	19.8	19.8	19.8	0.0	-	6E+03	9E+03	0	-
T_r^{7-2}	2006	6.0	9.9	-	-	2006	-	19.0	19.0	19.0	19.0	10.2	-	8E+03	5E+03	4	-
T_r^{7-3}	879	4.3	6.4	2324	-	879	1059	7.3	7.2	7.3	7.2	0.0	20.2	8E+03	9E+03	-3	34
T_r^{7-4}	1201	4.4	8.4	-	-	1201	1763	19.4	19.1	19.4	19.1	5.3	41.3	9E+03	7E+03	3	26
T_r^{7-5}	3140	7.0	11.7	-	-	3140	-	36.7	36.7	36.7	36.7	27.9	-	6E+03	5E+03	26	-
T_r^{8-1}	2715	2.1	3.7	291	-	2715	74472	46.2	42.7	46.2	42.7	0.0	96.5	2E+03	2E+04	18	47
T_r^{8-2}	2188	1.6	3.0	150	2763	2188	2188	38.3	34.6	38.3	34.6	0.0	0.0	8E+02	1E+04	0	0
T_r^{8-3}	569	8.9	31.5	-	-	569	588	56.8	32.0	56.8	32.0	49.6	29.9	4E+03	4E+03	18	18
T_r^{8-4}	1155	0.7	2.1	24	57	1155	1155	12.7	11.4	12.7	11.4	0.0	0.0	4E+02	6E+02	0	0
T_r^{8-5}	1065	22.0	27.1	-	-	1155	1065	45.3	17.9	45.3	17.9	41.4	8.9	3E+03	2E+03	-31	-31
T_r^{9-1}	44840	6.9	14.7	3398	-	44840	-	24.2	24.2	24.2	24.2	0.0	-	1E+04	6E+03	15	-
T_r^{9-2}	34225	5.4	11.0	2793	-	34225	360487	15.7	15.7	15.7	15.7	0.0	91.6	1E+04	5E+03	3	40
T_r^{9-3}	14782	3.7	8.6	392	-	14782	129567	32.2	32.0	32.2	32.0	0.0	90.7	1E+03	8E+03	8	40
T_r^{9-4}	14348	6.0	10.4	-	-	14348	69999	33.5	33.5	33.5	33.5	18.2	85.0	9E+03	5E+03	4	32
T_r^{9-5}	9522	3.6	8.7	590	-	9522	18224	22.1	22.1	22.1	22.1	0.0	51.6	3E+03	1E+04	0	16
T_r^{10-1}	600	6.8	13.8	-	-	600	-	38.9	16.6	38.9	16.6	25.5	-	5E+03	3E+03	0	-
T_r^{10-2}	392	6.2	17.2	-	1892	436	392	40.1	13.9	40.1	13.9	30.0	0.0	6E+03	5E+03	22	10
T_r^{10-3}	2977	9.4	15.1	-	-	2977	-	68.2	55.5	68.2	55.5	65.0	-	4E+03	2E+03	0	-
T_r^{10-4}	663	8.4	19.3	-	-	663	682	46.7	26.0	46.7	26.0	35.5	19.6	5E+03	5E+03	9	17
T_r^{10-5}	1410	11.6	10.2	-	-	1410	-	56.9	43.7	56.9	43.7	48.9	-	3E+03	2E+03	-1	-

Table G.2: Test results for the TE-MSTP-p problem: $T^6_{rand} - T^{10}_{rand}$.

ins.	B^*	LP-time (s)		MIP-time (s)		B		Gap_{LP}(%)		Gap_0 (%)		Gap_{end} (%)		B&B Tree Nodes		Gap^*_U (%)	
		B	S	B	S	B	S	B	S	B	S	B	S	B	S	B	S
T^{1-1}_{3tc}	21176	0.1	0.2	0	1	21176	21176	0.0	0.0	0.0	0.0	0.0	0.0	0E+00	3E+00	11	11
T^{1-2}_{3tc}	28644	0.6	1	4	15	28644	28644	1.9	1.9	1.9	1.9	0.0	0.0	1E+02	3E+02	25	25
T^{1-3}_{3tc}	19742	0.9	1.5	90	244	19742	19742	6.5	6.5	6.5	6.5	0.0	0.0	1E+03	2E+03	0	0
T^{1-4}_{3tc}	29950	0.7	1.2	13	121	29950	29950	2.5	2.5	2.5	2.5	0.0	0.0	3E+02	2E+03	0	0
T^{1-5}_{3tc}	18710	0.7	1.5	270	2340	18710	18710	5.4	5.4	5.4	5.4	0.0	0.0	2E+03	2E+04	0	0
T^{2-1}_{3tc}	31586	2.2	2.8	55	286	31586	31586	1.2	1.2	1.2	1.2	0.0	0.0	7E+02	3E+03	23	23
T^{2-2}_{3tc}	52956	0.3	0.9	1	4	52956	52956	0.0	0.0	0.0	0.0	0.0	0.0	8E+00	3E+01	0	0
T^{2-3}_{3tc}	66190	0.4	0.9	4	19	66190	66190	0.7	0.7	0.7	0.7	0.0	0.0	1E+02	4E+02	0	0
T^{2-4}_{3tc}	63167	1.2	2.4	257	1996	63167	63167	1.3	1.3	1.3	1.3	0.0	0.0	4E+03	2E+04	0	0
T^{2-5}_{3tc}	40886	0.7	3.2	2	17	40886	40886	0.2	0.2	0.2	0.2	0.0	0.0	1E+01	1E+02	27	27
T^{3-1}_{3tc}	45356	23.9	95.3	-	-	45356	54030	12.5	12.5	12.5	12.5	10.3	30.0	2E+03	2E+03	0	14
T^{3-2}_{3tc}	43016	6.4	12.6	1753	-	43016	47182	2.8	2.8	2.8	2.8	0.0	10.0	6E+03	1E+04	13	31
T^{3-3}_{3tc}	48736	15.2	54.7	-	-	48736	54084	8.6	8.6	8.6	8.6	6.2	20.0	4E+03	4E+03	10	0
T^{3-4}_{3tc}	22454	8.1	19.5	-	-	22454	23448	7.9	7.9	7.9	7.9	4.1	10.0	8E+03	7E+03	22	25
T^{3-5}_{3tc}	52912	2	5.8	23	157	52912	52912	0.8	0.8	0.8	0.8	0.0	0.0	2E+02	1E+03	24	24
T^{4-1}_{3tc}	98343	41.1	479.1	1869	-	98343	106878	0.8	0.8	0.8	0.8	0.0	10.0	1E+03	2E+03	0	0
T^{4-2}_{3tc}	64990	20.2	70.4	-	-	64990	68310	1.9	1.9	1.9	1.9	0.8	10.0	4E+03	3E+03	18	13
T^{4-3}_{3tc}	101085	22.4	31.2	2290	-	101085	104398	0.6	0.6	0.6	0.6	0.0	0.0	3E+03	4E+03	0	0
T^{4-4}_{3tc}	333944	3.3	15	622	-	333944	337424	1.7	1.7	1.7	1.7	0.0	0.0	5E+03	1E+04	0	0
T^{4-5}_{3tc}	92233	6.6	20.5	-	-	92233	93467	2.3	2.3	2.3	2.3	0.4	0.0	9E+03	6E+03	0	0
T^{5-1}_{3tc}	28604	41.2	192.8	2663	-	28604	-	5.4	5.4	5.4	5.4	0.0	-	1E+03	9E+02	30	-
T^{5-2}_{3tc}	22854	53.6	1042.8	-	-	22854	24736	12.0	12.0	12.0	12.0	9.3	20.0	7E+02	4E+02	8	24
T^{5-3}_{3tc}	34338	60.7	487.5	-	-	34338	34746	19.2	19.1	19.2	19.1	18.3	20.0	8E+02	4E+02	0	0
T^{5-4}_{3tc}	47400	50	232.2	-	-	47728	47400	24.1	23.8	24.1	23.8	23.0	20.0	1E+03	6E+02	19	17
T^{5-5}_{3tc}	78105	24.2	131.2	-	-	78105	85514	22.8	18.7	22.8	18.7	17.8	20.0	2E+03	1E+03	10	28
T^{6-1}_{3tc}	96713	141.1	501	-	-	96713	129759	8.5	8.4	8.5	8.4	7.7	30.0	4E+02	2E+02	20	50
T^{6-2}_{3tc}	276324	309.7	1315.7	-	-	276324	-	18.0	17.9	18.0	17.9	17.8	-	1E+02	4E+01	0	-
T^{6-3}_{3tc}	113150	264.3	817.8	-	-	113150	-	13.9	13.8	13.9	13.8	13.5	-	2E+02	7E+01	0	-
T^{6-4}_{3tc}	175966	161.3	759.5	-	-	175966	296190	8.1	8.1	8.1	8.1	7.6	50.0	5E+02	3E+02	0	28
T^{6-5}_{3tc}	125015	191.7	626.4	-	-	125015	-	14.9	14.9	14.9	14.9	14.6	-	2E+02	1E+02	21	-

Table G.3: Test results for the TE-MSTP-p problem: T_{3tc}.

ins.	B&C-time	IO-time	UB	Gap_{io}	B&C Tree Nodes	# User cuts	# Lazy const.
T_r^{1-1}	6	2	238	8	3E+01	160	3
T_r^{1-2}	26	1	271	100	8E+01	664	28
T_r^{1-3}	-	5	271	100	1E+03	23065	103
T_r^{1-4}	41	1	641	100	2E+02	1124	26
T_r^{1-5}	9	3	116	3	3E+01	462	2
T_r^{2-1}	-	4	2958	100	5E+03	81047	457
T_r^{2-2}	-	4	4425	100	5E+03	83655	385
T_r^{2-3}	-	22	2779	100	1E+03	29034	69
T_r^{2-4}	-	3	5437	100	5E+03	83484	363
T_r^{2-5}	-	4	9183	100	4E+03	81780	398
T_r^{3-1}	-	5	282	100	4E+02	7941	38
T_r^{3-2}	-	1338	302	12	3E+02	5621	28
T_r^{3-3}	-	6	490	100	3E+03	47764	151
T_r^{3-4}	-	816	240	16	5E+02	7817	16
T_r^{3-5}	-	4	327	100	5E+02	9452	41
T_r^{4-1}	-	4	159	100	3E+02	6270	28
T_r^{4-2}	-	7	371	100	4E+02	6937	29
T_r^{4-3}	??	??	??	??	??	??	??
T_r^{4-4}	-	3	234	100	4E+02	6650	32
T_r^{4-5}	-	2	61	100	6E+02	10390	28
T_r^{5-1}	-	7	585	100	3E+02	5066	23
T_r^{5-2}	-	10	313	100	2E+02	4892	28
T_r^{5-3}	-	4	144	100	3E+02	5967	28
T_r^{5-4}	-	6	190	100	2E+02	4249	28
T_r^{5-5}	-	15	369	100	2E+02	4486	18
T_r^{6-1}	-	9	333	100	2E+02	4564	31
T_r^{6-2}	-	14	706	100	3E+02	4798	28
T_r^{6-3}	-	8	808	100	4E+02	7025	45
T_r^{6-4}	-	5	205	100	3E+02	6403	28
T_r^{6-5}	-	10	427	100	3E+02	6103	32
T_r^{7-1}	-	19	1558	100	2E+02	3878	33
T_r^{7-2}	-	48	9863	100	1E+03	17334	125
T_r^{7-3}	-	23	1567	100	2E+02	4107	37
T_r^{7-4}	-	25	5125	100	2E+03	28559	130
T_r^{7-5}	-	28	15881	100	1E+03	22963	139
T_r^{8-1}	-	11	4166	100	5E+02	7988	25
T_r^{8-2}	-	12	4824	100	4E+03	44997	561
T_r^{8-3}	-	95	1925	100	1E+02	2190	19
T_r^{8-4}	2408	8	1155	100	7E+02	8282	30
T_r^{8-5}	-	91	4115	100	2E+02	2807	23
T_r^{9-1}	-	41	262215	100	1E+03	19736	187
T_r^{9-2}	-	67	141607	100	1E+02	2837	45
T_r^{9-3}	-	37	16569	100	2E+02	3072	30
T_r^{9-4}	-	60	32066	100	2E+02	3318	38
T_r^{9-5}	-	24	10922	100	2E+02	3865	42
T_r^{10-1}	-	60	836	100	1E+03	16648	222
T_r^{10-2}	-	131	573	100	1E+02	2259	19
T_r^{10-3}	-	48	11115	100	1E+03	15410	166
T_r^{10-4}	-	86	936	100	2E+02	3024	17
T_r^{10-5}	-	74	2527	100	2E+02	3008	20

Table G.4: Test results for the B&C algorithm: T_{rand}.

ins.	B&C-time	IO-time	UB	Gap_{io}	B&C Tree Nodes	# User cuts	# Lazy const.
T_{3tc}^{1-1}	3	3	21176	0	0E+00	0	1
T_{3tc}^{1-2}	178	22	28604	2	1E+02	926	10
T_{3tc}^{1-3}	-	84	19742	7	7E+02	9214	22
T_{3tc}^{1-4}	317	23	29950	3	2E+02	1586	5
T_{3tc}^{1-5}	-	81	18716	5	6E+02	8654	15
T_{3tc}^{2-1}	-	79	31586	1	6E+02	5502	3
T_{3tc}^{2-2}	24	16	52956	0	9E+00	14	3
T_{3tc}^{2-3}	121	21	66190	1	5E+01	175	2
T_{3tc}^{2-4}	-	119	63167	1	8E+02	14058	17
T_{3tc}^{2-5}	131	73	40886	0	2E+01	220	2
T_{3tc}^{3-1}	-	-	-	-	-	-	-
T_{3tc}^{3-2}	-	821	44506	8	2E+02	3297	32
T_{3tc}^{3-3}	-	-	-	-		-	-
T_{3tc}^{3-4}	-	1568	24216	19	6E+01	1192	22
T_{3tc}^{3-5}	1151	232	52912	1	2E+02	1559	17
T_{3tc}^{4-1}	-	1580	139968	100	6E+00	131	9
T_{3tc}^{4-2}	-	-	-	-	-	-	-
T_{3tc}^{4-3}	-	2332	110785	12	2E+01	387	3
T_{3tc}^{4-4}	-	151	394783	100	3E+02	5008	56
T_{3tc}^{4-5}	-	1012	94771	6	5E+01	1079	11
T_{3tc}^{5-1}	-	-	-	-	-	-	-
T_{3tc}^{5-2}	-	-	-	-	-	-	-
T_{3tc}^{5-3}	-	1549	45104	100	3E+00	73	13
T_{3tc}^{5-4}	-	-	-	-	-	-	-
T_{3tc}^{5-5}	-	-	-	-	-	-	-
T_{3tc}^{6-1}	-	-	-	-	-	-	-
T_{3tc}^{6-2}	-	-	-	-	-	-	-
T_{3tc}^{6-3}	-	-	-	-	-	-	-
T_{3tc}^{6-4}	-	1398	309579	100	6E+00	133	19
T_{3tc}^{6-5}	-	-	-	-	-	-	-

Table G.5: Test results for the B&C algorithm: T_{3tc}.