ALGORITMOS DE OTIMIZAÇÃO PARA O *STEINER TEAM ORIENTEERING PROBLEM*

LUCAS ASSUNÇÃO DE ALMEIDA

ALGORITMOS DE OTIMIZAÇÃO PARA O *STEINER TEAM ORIENTEERING PROBLEM*

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Engenharia de Produção.

Orientador: Geraldo Robson Mateus

Belo Horizonte
Agosto de 2019

LUCAS ASSUNÇÃO DE ALMEIDA

OPTIMIZATION ALGORITHMS FOR THE STEINER TEAM ORIENTEERING PROBLEM

Thesis presented to the Graduate Program in Production Engineering of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Production Engineering.

Advisor: Geraldo Robson Mateus

Belo Horizonte
August 2019

Almeida, Lucas Assunção de

A447o Optimization algorithms for the Steiner team orienteering problem [recurso eletrônico] / Lucas Assunção de Almeida. - 2019.
1 recurso online (xxvi, 142 f. : il., color.) : pdf

Orientador: Geraldo Robson Mateus

Tese (doutorado) – Universidade Federal de Minas Gerais, Escola de Engenharia
Apêndices: f. 85-142.

Bibliografia: f. 81-84.
Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia de produção – Teses. 2. Otimização combinatória - Teses. I. Mateus, Geraldo Robson. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 658.5(043)

Ficha catalográfica: Biblioteca Prof° Mário Werneck, Escola de Engenharia da UFMG

UNIVERSIDADE FEDERAL DE MINAS GERAIS

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

UFMG

FOLHA DE APROVAÇÃO

Optimization algorithms for the Steiner team orienteering problem

LUCAS ASSUNÇÃO DE ALMEIDA

Tese submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA DE PRODUÇÃO, como requisito para obtenção do grau de Doutor em ENGENHARIA DE PRODUÇÃO, área de concentração PESQUISA OPERACIONAL E INTERVENÇÃO EM SISTEMAS SOCIOTÉCNICOS, linha de pesquisa Mod. e Algorit. de Otimiz. para Sistemas em Redes e de Prod..

Aprovada em 19 de agosto de 2019, pela banca constituída pelos membros:

Prof(a). Geraldo Robson Mateus - Orientador
UFMG

Prof(a). Martin Gomez Ravetti
UFMG

Prof(a). Ricardo Saraiva de Camargo
UFMG

Prof(a). Marcus Vinicius Soledade Poggi de Aragao
PUC Rio de Janeiro

Prof(a). Eduardo Uchoa Barboza
Departamento de Engenharia de Produção - UFF

Belo Horizonte, 7 de outubro de 2019.

Ao meu painho (que sempre quis um filho "doutor") e à minha mainha, por toda sorte de suporte até aqui.

"Feche os livros e vá viver."
(Pablo Neruda)

Resumo

O Team Orienteering Problem (TOP) é um problema NP-difícil de roteamento em que uma frota homogênea de veículos tem por objetivo coletar prêmios disponíveis em um determinado número de localidades, enquanto respeitando restrições de tempo de percurso. No TOP, cada localidade pode ser visitada por, no máximo, um veículo, e o objetivo é maximizar o montante de prêmios coletados pelos veículos dentro de um limite de tempo pré-estabelecido. Na nossa pesquisa, propomos uma generalização do TOP, denominada Steiner Team Orienteering Problem (STOP). No STOP, é dado, adicionalmente, um subconjunto de localidades obrigatórias. Assim, o STOP também tem por objetivo maximizar o total de prêmios coletados dentro de um limite de tempo, mas, agora, cada localidade obrigatória deve ser visitada.

Como uma primeira contribuição, propomos para o STOP uma nova formulação baseada em *produto* e a usamos dentro de um esquema de planos de cortes. O algoritmo beneficia-se da compacidade e da força da formulação proposta e funciona separando cinco famílias de desigualdades válidas, a saber: restrições de conectividade, clássicas *lifted cover inequalities* baseadas em limites duais, uma classe de desigualdades denominadas *arc-vertex inference cuts* e duas classes de cortes baseados em vértices conflitantes. Até onde sabemos, as últimas três classes de desigualdades são também inéditas na literatura, sendo aqui introduzidas junto às suas respectivas provas de validade. Um algoritmo *branch-and-cut* que é estado-da-arte na literatura do TOP foi adaptado ao STOP e usado como *baseline* para avaliar a performance do algoritmo de planos de cortes proposto. Extensivos experimentos computacionais mostram a competitividade do novo algoritmo na resolução de instâncias do STOP e do TOP. Em particular, o algoritmo é capaz de resolver, no total, 14 instâncias do TOP a mais do que qualquer outro algoritmo exato na literatura, além de encontrar nove novos certificados de otimalidade. Com relação às novas instâncias do STOP introduzidas neste trabalho, nosso algoritmo resolve 31 instâncias a mais do que o *baseline*.

Neste trabalho, também provamos que encontrar uma solução viável para o STOP é NP-difícil e propomos uma heurística *Large Neighborhood Search* (LNS) para o problema. A heurística é inicializada com soluções obtidas pela matheurística conhecida pelo nome de *Feasibility Pump*, a qual, em nossa implementação, tem por base a formulação compacta que propomos para o STOP. A heurística LNS em si combina buscas locais clássicas da literatura de problemas de roteamento com um componente de memória de longo-prazo baseado em *Path Relinking*. Experimentos computacionais mostram a eficiência e eficácia da heurística proposta quando resolvendo um conjunto de 387 instâncias do STOP. No geral, as soluções heurísticas obtidas implicam um *gap* percentual de apenas 0.54% em relação aos melhores limites conhecidos para

as instâncias. Em particular, a heurística atinge os melhores limites já sabidos em 382 das 387 instâncias utilizadas. Ademais, em 21 desses casos, nossa heurística é ainda capaz de melhorar esses limites.

Por fim, o algoritmo híbrido obtido ao inicializar o algoritmo de planos de cortes com as soluções providas pela heurística LNS é capaz de resolver na otimalidade quatro instâncias do TOP e sete instâncias do STOP a mais do que o algoritmo de planos de cortes sozinho. Além disso, o algoritmo híbrido obtém novos certificados de otimalidade para cinco instâncias do TOP ainda não resolvidas por nenhum outro algoritmo da literatura (incluindo nosso algoritmo de planos de cortes sem a ajuda da heurística).

Abstract

The Team Orienteering Problem (TOP) is an NP-hard routing problem in which a fleet of identical vehicles aims at collecting rewards (prizes) available at given locations, while satisfying restrictions on the travel times. In the TOP, each location can be visited by at most one vehicle, and the goal is to maximize the total sum of rewards collected by the vehicles within a given time limit. In our research, we propose a generalization of the TOP, namely the Steiner Team Orienteering Problem (STOP). In the STOP, we provide, additionally, a subset of mandatory locations. In this sense, the STOP also aims at maximizing the total sum of rewards collected within the time limit, but, now, every mandatory location must be visited.

As a first contribution, we propose a new commodity-based formulation for the STOP and use it within a cutting-plane scheme. The algorithm benefits from the compactness and strength of the proposed formulation and works by separating five families of valid inequalities, which consist of some general connectivity constraints, classical lifted cover inequalities based on dual bounds, a class of the so-called arc-vertex inference cuts and classes of conflict cuts and clique conflict cuts. To our knowledge, the last three classes of inequalities are also introduced in this work, along with the due proofs of validity. A state-of-the-art branch-and-cut algorithm from the literature of the TOP is adapted to the STOP and used as baseline to evaluate the performance of the cutting-plane. Extensive computational experiments show the competitiveness of the new algorithm while solving several STOP and TOP instances. In particular, it is able to solve, in total, 14 more TOP instances than any other previous exact algorithm and finds nine new optimality certificates. With respect to the new STOP instances introduced in this work, our algorithm solves 31 more instances than the baseline.

In this work, we also prove that solely finding a feasible solution for the STOP is NP-hard and propose a Large Neighborhood Search (LNS) heuristic for the problem. The algorithm is provided with initial solutions obtained by means of the matheuristic framework known as Feasibility Pump, which, in our implementation, uses as backbone the commodity-based formulation we propose. The LNS heuristic itself combines classical local searches from the literature of routing problems with a long-term memory component based on Path Relinking. Computational experiments show the efficiency and effectiveness of the proposed heuristic in solving a benchmark of 387 STOP instances. Overall, the heuristic solutions imply an average percentage gap of only 0.54% when compared to the best known bounds. In particular, the heuristic reaches the best previously known bounds on 382 of the 387 instances. Additionally, in 21 of these cases, our heuristic is even able to improve over the best known bounds.

At last, the hybrid algorithm obtained from warm starting our cutting-plane algorithm

with the LNS heuristic is able to solve to optimality four more TOP instances and seven more STOP instances than the cutting-plane algorithm alone. Additionally, it provides the optimality certificates of five previously unsolved (even by our plain cutting-plane algorithm) TOP instances.

List of Figures

List of Tables

List of Acronyms

LP	*Linear Programming*
ILP	*Integer Linear Programming*
MILP	*Mixed Integer Linear Programming*
OP	*Orienteering Problem*
TOP	*Team Orienteering Problem*
STOP	*Steiner Team Orienteering Problem*
CVRP	*Capacitated Vehicle Routing Problem*
HPP	*Hamiltonian Path Problem*
VNS	*Variable Neighborhood Search*
TS	*Tabu Search*
SA	*Simulated Annealing*
LNS	*Large Neighborhood Search*
GRASP	*Greedy Randomized Adaptive Search Procedure*
PR	*Path Relinking*
PSO	*Particle Swarm Optimization*
FP	*Feasibility Pump*
OFP	*Objective Feasibility Pump*
RINS	*Relaxation Induced Neighborhood Search*
B-B&C	*Baseline Branch-and-Cut algorithm*
CPA	*Cutting-Plane Algorithm*

Contents

Chapter 1

Introduction

Orienteering is a sport usually practiced in places with irregular terrains, such as mountains and dense forests. It is given a set of *control points* to be visited, each of them with an associated reward (prize). The competitors are provided a topographical map and compass in order to guide them from an origin point to a destination one, which are the same for all competitors. Their goal is to maximize the total sum of rewards collected from visiting the control points within a previously established time limit. Each reward can be collected by a single competitor, and the winner is the one that reaches the destination point within the time limit with the maximum amount of rewards.

Based on this sport, Tsiligirides introduced the *Orienteering Problem* (OP) (Tsiligirides, 1984). The problem is defined on a graph, usually complete and undirected, where a value of reward is associated with each vertex and a traverse time is associated with each edge (or arc). The OP aims at finding a route from an origin vertex to a destination one (visiting each vertex at most once) that satisfies a total traverse time constraint while maximizing the sum of rewards collected. In the OP, a reward can only be collected once, just like in the original orienteering sport. In fact, an optimal route for the OP corresponds to an optimal one for an orienteering competitor, except for the fact that, in the OP, no vertex can be visited more than once. We also point out that, contrary to the classical *traveling salesman problem* (Dantzig et al., 1954), a solution for the OP does not necessarily visit all the vertices of the graph.

When the origin and destination vertices coincide, the problem is known as the *selective traveling salesman problem* (Laporte and Martello, 1990). Moreover, when we consider a team of competitors working together, the problem becomes the Team Orienteering Problem (TOP) (Chao et al., 1996). In the TOP, all the m members of the team depart from the same vertex at the same time and have to arrive at the destination vertex, also within a same time limit. The goal is to find m routes that, together, maximize the total reward collected by the team. As for the OP, a vertex/reward cannot be visited/collected multiple times, i.e., once a member of the team collects the reward of a vertex, this vertex cannot be visited again.

Both the OP and the TOP are NP-hard (Laporte and Martello, 1990; Poggi et al., 2010) and find applications in transportation and delivery of goods (de Freitas Viana, 2011). With the advent of the *e-commerce*, for instance, several virtual stores assign their delivery requests to different shipping companies. Nevertheless, the fleet available to a given shipping company is not always enough to perform all the deliveries assigned to it in a single working day. In these cases, the company must select only a subset of the total amount of its deliveries. To this end, a value of priority can be associated with each delivery. This value corresponds to the reward achieved by performing the delivery in the current working day and might combine different factors, such as the priority of the client and the urgency of the request.

A similar application arises in the planning of technical visits (Tang and Miller-Hooks, 2005). Also in this case, a reward is associated with visiting each customer and performing a given service. Likewise, the values of the rewards rely on factors such as the urgency of the request and the customer priority. Therefore, the goal is to select a subset of technical visits (to be performed within a working horizon of time) that maximizes the total sum of rewards achieved. Notice that, in both applications, this priority policy is not enough to ensure that deliveries or technical visits with top priority (e.g., those whose deadlines are expiring) will be necessarily selected in the planning. In this study, we propose a variation of the TOP, namely the *Steiner Team Orienteering Problem* (STOP), that addresses this issue.

The STOP is defined on a digraph, where an origin and a destination vertices are given, and the remaining vertices are subdivided into two categories: the mandatory ones, which must necessarily be visited, and the profitable ones, which work as Steiner vertices and, thus, may not be visited. A traverse time is associated with each arc in this digraph, and values of reward are associated with visiting the profitable vertices. In order to represent the team of members, it is also given a homogeneous fleet of vehicles, which can only run for a given time limit. The STOP aims at finding routes (one for each vehicle) from the origin vertex to the destination one such that every mandatory vertex belongs to exactly one route and the total sum of rewards collected on the visited profitable vertices is maximized. Here, each profitable vertex can be visited by at most one vehicle, thus avoiding the multiple collection of a same reward.

Main contributions

The aim of this research is to devise exact and heuristic algorithms for the STOP, which can also be applied to its specific cases, such as the TOP and the OP. Regarding exact approaches, our main contribution consists of introducing a commodity-based compact formulation for the STOP and devising a cutting-plane scheme to solve it. The cutting-plane relies on the separation of five families of inequalities, which consist of some (i) general connectivity constraints, (ii) classical lifted cover inequalities based on dual bounds, classes of (iii) conflict cuts and (iv) clique conflict cuts, as well as the so-called (v) arc-vertex inference cuts. As far as we are aware, the last three families of inequalities are also introduced in this work and can be applied to similar problems in a straightforward manner. The due proofs regarding the validity of these new inequalities are also given, along with examples of the fractional solutions cut off by them.

Our exact algorithm highly benefits from the compactness and the strength of the formulation proposed, which we prove to give the same bounds as the one used within a state-of-the-art branch-and-cut algorithm from the literature of the TOP. In this work, we adapt this branch-and-cut algorithm to the STOP and use it as a baseline to evaluate the performance of the cutting-plane proposed. According to extensive experiments, our algorithm shows to be highly competitive with previous exact approaches in the literature. In fact, it is able to solve, in total, 14 more instances than any other TOP exact algorithm and finds the optimality certificates of nine previously unsolved TOP instances. With respect to the new STOP instances introduced in this work, the new algorithm solves to optimality 31 more instances than the baseline.

As to motivate the research on heuristics for the problem, we first prove that solely finding a feasible solution for STOP is NP-hard, thus formalizing the additional difficulty of the problem when compared to its more specific cases with no mandatory vertices (like TOP). Then, we propose a Large Neighborhood Search (LNS) heuristic for the problem. The algorithm is provided with initial solutions obtained by means of the matheuristic framework known as Feasibility Pump (FP) (Fischetti et al., 2005). In our implementation, FP uses as backbone our commodity-based formulation reinforced by the classes of inequalities separated in our cutting-plane algorithm. The LNS heuristic itself combines classical local searches from the literature of routing problems with a long-term memory component based on Path Relinking (PR). We use the primal bounds provided by our cutting-plane algorithm to evaluate the quality of the solutions obtained by the heuristic. Computational experiments show the efficiency and effectiveness of the proposed heuristic in solving a benchmark of 387 STOP instances. Overall, the heuristic solutions imply an average percentage gap of only 0.54% when compared to the bounds of the cutting-plane baseline. In particular, the heuristic reaches the best previously known bounds on 382 of the 387 instances. Additionally, in 21 of these cases, our heuristic is even able to improve over the best known bounds.

For completeness, we also test the hybrid algorithm obtained from warm starting our cutting-plane algorithm with the solutions provided by our heuristic. Computational experiments show that this approach is able to improve the performance of our plain cutting-plane. In total, the hybrid algorithm is able to solve to optimality four more TOP instances and seven more STOP instances than the plain cutting-plane algorithm. Additionally, it provides the optimality certificates of five previously unsolved (even by our plain cutting-plane algorithm) TOP instances.

Thesis' outline

The remainder of this thesis is organized as follows. Related works are discussed in Section 1.1, and the STOP is formally defined in Section 1.2. Chapter 2 is dedicated to presenting the mathematical foundation used to devise the algorithms proposed. Within this chapter, in Section 2.1, we present two formulations for the STOP and prove that they provide the same bounds. In Section 2.2, we describe five families of inequalities able to reinforce the original formulations, and the procedures used to separate them are presented in Section 2.3.

Chapter 3 is devoted to the exact algorithms. The baseline branch-and-cut algorithm adapted to the STOP is briefly described in Section 3.1, and the cutting-plane scheme proposed is detailed in Section 3.2. Some implementation details are given in Section 3.3, followed by the due computational results (Section 3.4). In Section 3.5 (and Appendix A.6), we discuss improving the convergence of the exact algorithms by warm starting them with primal heuristic solutions. Chapter 4 is dedicated to the heuristic approaches. In Section 4.1, we prove that solely finding an initial feasible solution for the STOP is NP-hard and describe the FP procedure applied to address this issue. Section 4.2 is devoted to detailing the LNS heuristic proposed, and some implementation choices are discussed in Section 4.3, followed by the due computational results (Section 4.4). At last, in Chapter 5, we pinpoint some concluding remarks and briefly discuss future work directions, along with some unsuccessful approaches we tested during the research.

1.1 Related works

Although the STOP has not been addressed in the literature yet, a specific case of the problem that considers a single vehicle, namely the *Steiner orienteering problem*, was already introduced by (Letchford et al., 2013). In the work, the authors propose four Integer Linear Programming (ILP) models for the problem, but no computational experiment is reported. The STOP is also closely related to several routing problems, such as the TOP, the OP and the Capacitated Vehicle Routing Problem (CVRP) (Toth and Vigo, 2001) and its variations. In the remainder of this section, we present a literature review on the main heuristic and exact algorithms to solve the TOP, the problem most closely related to the STOP.

The particular case of the STOP with no mandatory vertices, namely the TOP, was introduced by the name of the *multiple tour maximum collection problem* in the work of Butt and Cavalier (1994). Nevertheless, the problem was only formally defined by Chao et al. (1996). In the latter work, the first benchmark of TOP instances was proposed, along with a heuristic procedure. This benchmark, which consists of complete graphs with 21 to 102 vertices, is adopted in all the TOP works from the literature discussed next.

Throughout the last decade, several heuristics have been proposed for the TOP. For instance, Tang and Miller-Hooks (2005) presented an algorithm that combines a *Tabu Search* (TS) heuristic with an adaptive memory procedure. Archetti et al. (2007) developed two more TS heuristics for the problem, as well as two procedures based on *Variable Neighborhood Search* (VNS). In addition, Ke et al. (2008) proposed *ant colony* based algorithms which presented results comparable to those of Archetti et al. (2007), with less computational time effort. The VNS heuristic of Vansteenwegen et al. (2009) was the first procedure to focus on time efficiency. However, the quality of the solutions obtained by it is slightly worse than that of the solutions obtained by Archetti et al. (2007).

Later on, Souffriau et al. (2010) proposed a *Greedy Randomized Adaptive Search Procedure* (GRASP) metaheuristic with *Path Relinking* (PR) which was able to outperform all the heuristic approaches aforementioned (Chao et al., 1996; Tang and Miller-Hooks, 2005; Archetti et al., 2007; Ke et al., 2008; Vansteenwegen et al., 2009) both in effectiveness (i.e., bounds of the solutions) and time efficiency. More recently, three new approaches were able to outperform the results of Souffriau et al. (2010): the *Simulated Annealing* (SA) heuristic of Lin (2013), the *Large Neighborhood Search* (LNS) based heuristics of Kim et al. (2013) and the evolutionary algorithm of Dang et al. (2013b), which is inspired by *Particle Swarm Optimization* (PSO).

The algorithm of Dang et al. (2013b) showed to be competitive with the ones of Kim et al.

(2013) in terms of the quality of the solutions obtained for complete graph instances with up to 100 vertices. However, according to the results, the latter heuristics (Kim et al., 2013) are more efficient. Dang et al. (2013b) also tested their evolutionary algorithm on larger instances, with up to 400 vertices. Due to the lack of optimality certificates for these instances, the heuristic was only evaluated in terms of stability and time efficiency in these cases. The results obtained by Lin (2013) were not compared to those of Kim et al. (2013) and Dang et al. (2013b).

To our knowledge, the latest heuristic for the TOP was proposed by Ke et al. (2016). Their heuristic, namely *Pareto mimic algorithm*, introduces a so-called *mimic operator* to generate new solutions by imitating incumbent ones. The algorithm also adopts the concept of *Pareto dominance* to update the population of incumbent solutions by considering multiple indicators that measure the quality of each solution. The results indicate that this new algorithm can achieve all the best-known bounds obtained by Lin (2013) and Dang et al. (2013b). In addition, the algorithm of Ke et al. (2016) was even able to find improved bounds for 10 of the larger instances (with up to 400 vertices) introduced by Dang et al. (2013b).

A few works propose exact solution approaches for the TOP. As far as we know, Butt and Ryan (1999) presented the first exact algorithm for the TOP, which is based on column generation. More recently, Boussier et al. (2007) proposed a *set packing* formulation with an exponential number of variables, each of them representing a feasible route. The formulation is solved by means of a branch-and-price algorithm, and the pricing sub-problems are solved through dynamic programming. Poggi et al. (2010) proposed a branch-and-cut-and-price algorithm, along with new min-cut and triangle clique inequalities. The algorithm solves a Dantzig-Wolfe reformulation of a pseudo-polynomial compact formulation where edges are indexed by the time they are placed in a route.

Later on, Dang et al. (2013a) developed a branch-and-cut algorithm that relies on a set of dominance properties and valid inequalities, such as symmetry breaking, generalized sub-tour eliminations and clique cuts based on graphs of incompatibilities. The algorithms of Poggi et al. (2010) and Dang et al. (2013a) were both able to obtain new optimality certificates. Moreover, the branch-and-cut algorithm of Dang et al. (2013a) was able to outperform the branch-and-price algorithm of Boussier et al. (2007) in terms of the total number of instances solved to optimality. Since Poggi et al. (2010) do not report the experimental results for the whole benchmark of TOP instances in the literature, the performance of their algorithm could not be properly compared with other approaches.

Recently, Keshtkaran et al. (2016) proposed a branch-and-price algorithm where the pricing sub-problems are solved be means of a dynamic programming algorithm with decremental state space relaxation featuring a two-phase dominance rule relaxation. The authors also presented a branch-and-cut-and-price algorithm that incorporates a family of subset-row inequalities to the branch-and-price scheme. The two algorithms showed to be competitive with the previous exact methods in the literature. In fact, they both were able to outperform the algorithms of Boussier et al. (2007) and Dang et al. (2013a) in terms of the total number of instances solved to proven optimality within the same execution time limit of two hours. More recently, the work of Dang et al. (2013a) was extended by El-Hajj et al. (2016), where the authors attempt to solve the same formulation proposed by the former work via a *cutting-plane* algorithm. The algorithm explores intermediate models obtained by considering only a subset of the vehicles and uses the information iteratively retrieved to solve the original problem. Here, the promising inequalities introduced by Dang et al. (2013a) are also used to accelerate the convergence of the algorithm.

Overall, the branch-and-price of Keshtkaran et al. (2016) and the *cutting-plane* algorithm of El-Hajj et al. (2016) outperform the other exact algorithms previously discussed. In fact, they present a complementary behaviour when solving the hardest instance sets, i.e., on some instances, one is better than the other and vice-versa. As pointed out in both works, such behaviour constitutes a pattern between branch-and-cut and branch-and-price algorithms previously presented in the literature of the TOP.

A more recent work of Bianchessi et al. (2018) introduced a two-index compact (with a polynomial number of variables and constraints) formulation inspired by the one of Maffioli and Sciomachen (1997) for the *sequential ordering problem*, a scheduling problem where jobs have to be processed on a single machine and are subject to time windows and precedence relations. Bianchessi et al. (2018) reinforced this compact formulation for the TOP with connectivity constraints and solved it via a branch-and-cut algorithm developed with the *callback* mechanism of the optimization solver CPLEX[1]. This simple approach showed to be very effective in practice. In fact, the algorithm was able to solve at optimality 26 more instances than any other exact algorithm aforementioned when enabling multi-threading, and 10 more instances when not. All experiments used the CPLEX built-in cuts.

The branch-and-cut algorithm of Bianchessi et al. (2018) is adopted in this thesis as the baseline to evaluate the performance of the cutting-plane algorithms we propose. In particular, our new exact algorithms arise as the new state-of-the-art for the TOP, once they beat the results of Bianchessi et al. (2018). For detailed surveys on exact and heuristic resolution approaches for the TOP and its variants, we refer to the works of Vansteenwegen et al. (2011) and Gunawan et al. (2016).

1.2 Problem definition and notation

The STOP is defined on a digraph $G = (N, A)$, where N is the vertex set, and A is the arc set. Let $s, t \in N$ be the origin and destination vertices, respectively, with $s \neq t$. Moreover, let $S \subseteq N \backslash \{s, t\}$ be the subset of *mandatory* vertices, and $P \subseteq N \backslash \{s, t\}$ be the set of *profitable* vertices, such that $S \cap P = \emptyset$ and $N = S \cup P \cup \{s, t\}$. A reward $p_i \in \mathbb{Z}^+$ is associated with each vertex $i \in P$, and a traverse time $d_{ij} \in \mathbb{R}^+$ is associated with each arc $(i, j) \in A$. Each vehicle of the homogeneous fleet M can run for no more than a time limit T.

The STOP aims at finding up to $m = |M|$ routes from s to t such that every mandatory vertex in S belongs to exactly one route and the total sum of rewards collected by visiting profitable vertices is maximized. Here, each profitable vertex in P can be visited by at most one vehicle, thus avoiding the multiple collection of a same reward. Likewise, each mandatory vertex in S must be visited only once.

[1]http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

In the remainder of this thesis, we also consider the notation described as follows. Given a subset $V \subset N$, we define the sets of arcs leaving and entering V as $\delta^+(V) = \{(i,j) \in A : i \in V, j \in N \backslash V\}$ and $\delta^-(V) = \{(i,j) \in A : i \in N \backslash V, j \in V\}$, respectively. Similarly, given a vertex $i \in N$, we define the sets of vertices $\delta^+(i) = \{j \in N : (i,j) \in A\}$ and $\delta^-(i) = \{j \in N : (j,i) \in A\}$. Moreover, given two arbitrary vertices $i, j \in N$ and a path p from i to j in G, we define $A_p \subseteq A$ as the arc set of p.

Let R_{ij} denote the minimum time needed to reach a vertex j when departing from a vertex i in the graph G, i.e., $R_{ij} = \min\{\sum\limits_{a \in A_p} d_a : p \text{ is a path from } i \text{ to } j \text{ in } G\}$. Accordingly, $R_{ii} = 0$ for all $i \in N$, and, if no path exists from a vertex to another, the corresponding entry of R is set to infinity. One may observe that R is not necessarily symmetric, since G is directed. Moreover, considering that the traverse times associated with the arcs of G are non-negative (and, thus, no negative cycle exists), this R matrix can be computed *a priori* (for each instance) by means of the classical dynamic programming algorithm of Floyd-Warshall (Cormen et al., 2001), for instance.

Chapter 2

Mathematical formulations and valid inequalities

In this chapter, we detail all the mathematical foundation used to devise the algorithms (both exact and heuristic) developed in this study. In summary, we describe the mathematical formulations used as backbone for the algorithms, along with some classes of valid inequalities able to reinforce these formulations. The due separation procedures are also detailed.

2.1 Mathematical formulations

In this section, we present two compact Mixed Integer Linear Programming (MILP) formulations for the STOP. The first one, denoted by $\mathcal{F}_1$, directly extends the TOP formulation of Bianchessi et al. (2018) through the addition of constraints that impose the selection of mandatory vertices. The second one, denoted by $\mathcal{F}_2$, is a commodity-based formulation which, to the best of our knowledge, is also introduced in this work. In particular, $\mathcal{F}_1$ and $\mathcal{F}_2$ constitute, respectively, the backbone of the branch-and-cut baseline algorithm (discussed in Section 3.1) and of our cutting-plane algorithm (presented in Section 3.2). By the end of this section, we also give a formal proof of the equivalence of these formulations and discuss how we take advantage of a specific characteristic of $\mathcal{F}_2$ in the cutting-plane algorithm we propose. Moreover, we shortly describe some of the formulations that performed poorly in pilot experiments and were, thus, discarded from this study.

Now, consider the decision variables y on the choice of vertices belonging or not to the solution routes, such that $y_i = 1$ if the vertex $i \in N$ is visited by a vehicle of the fleet, and $y_i = 0$, otherwise. Likewise, let the binary variables x identify the solution routes themselves: $x_{ij} = 1$ if the arc $(i, j) \in A$ is traversed in the solution; $x_{ij} = 0$, otherwise. In addition, let the continuous variables z_{ij}, for all $(i, j) \in A$, represent the arrival time at vertex j of a vehicle directly coming from vertex i. The slack variable φ represents the number of vehicles that are not used in the solution. $\mathcal{F}_1$ is defined from (2.1) to (2.16).

$$(\mathcal{F}_1) \quad \max \quad \sum_{i \in P} p_i y_i, \tag{2.1}$$

$$
\begin{aligned}
s.t. \quad & y_i = 1 \qquad \forall\, i \in S \cup \{s,t\}, && (2.2)\\
& \sum_{j \in \delta^+(i)} x_{ij} = y_i \qquad \forall\, i \in S \cup P, && (2.3)\\
& \sum_{j \in \delta^+(s)} x_{sj} + \varphi = m, && (2.4)\\
& \sum_{i \in \delta^-(t)} x_{it} + \varphi = m, && (2.5)\\
& \sum_{i \in \delta^-(s)} x_{is} = 0, && (2.6)\\
& \sum_{j \in \delta^+(t)} x_{tj} = 0, && (2.7)\\
& \sum_{j \in \delta^+(i)} x_{ij} - \sum_{j \in \delta^-(i)} x_{ji} = 0 \qquad \forall\, i \in S \cup P, && (2.8)\\
& z_{sj} = d_{sj} x_{sj} \qquad \forall\, j \in \delta^+(s), && (2.9)\\
& \sum_{j \in \delta^+(i)} z_{ij} - \sum_{j \in \delta^-(i)} z_{ji} = \sum_{j \in \delta^+(i)} d_{ij} x_{ij} \qquad \forall\, i \in S \cup P, && (2.10)\\
& z_{ij} \le (T - R_{jt}) x_{ij} \qquad \forall (i,j) \in A, && (2.11)\\
& z_{ij} \ge (R_{si} + d_{ij}) x_{ij} \qquad \forall (i,j) \in A, && (2.12)\\
& x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A, && (2.13)\\
& y_i \in \{0,1\} \qquad \forall i \in N, && (2.14)\\
& z_{ij} \ge 0 \qquad \forall (i,j) \in A, && (2.15)\\
& 0 \le \varphi \le m. && (2.16)
\end{aligned}
$$

The objective function in (2.1) gives the total reward collected by visiting profitable vertices. Constraints (2.2) impose that all mandatory vertices (as well as s and t) are selected, while constraints (2.3) ensure that each vertex in $S \cup P$ is visited at most once. Restrictions (2.4) and (2.5) ensure that at most m vehicles leave the origin s and arrive at the destination t, whereas constraints (2.6) and (2.7) impose that vehicles cannot arrive at s nor leave t. Moreover, constraints (2.8), along with constraints (2.2) and (2.3), guarantee that, if a vehicle visits a vertex $i \in S \cup P$, then it must enter and leave this vertex exactly once.

Constraints (2.9)-(2.11) ensure that each of the solution routes from s to t has a total traverse time of at most T. In particular, constraints (2.9) implicitly set the depart time from vertex s to be zero, while constraints (2.10) manage the subsequent arrival times according to the vertices previously visited in each route. Constraints (2.11) impose that an arc $(i,j) \in A$ can only be traversed if the minimum extra time needed to reach t from j does not unfeasible the route it belongs. Restrictions (2.12) are, in fact, valid inequalities that provide lower bounds on the arrival times represented by the z variables, and restrictions (2.13)-(2.16) set the domain of the variables. Notice that, in $\mathcal{F}_1$, the continuous variables z work as flow variables, thus preventing the existence of sub-tours in the solutions.

Formulation $\mathcal{F}_1$ was originally stated by Bianchessi et al. (2018) with the additional inequality

$$
\sum_{(i,j) \in A} d_{ij} x_{ij} \le mT, \tag{2.17}
$$

which the authors claimed to strengthen the formulation. By the end of this section (see Corollary 1), we prove that such inequality is, in fact, redundant.

The commodity-based formulation we introduce in this work, namely $\mathcal{F}_2$, also considers the y and x decision variables as defined above, and the intuition behind it is also similar to that of $\mathcal{F}_1$. Precisely, in $\mathcal{F}_2$, time is treated as a commodity to be spent by the vehicles when traversing each arc in their routes, such that every vehicle departs from s with an initial amount of T units of commodity, the time limit. Accordingly, the z variables of $\mathcal{F}_1$ (related to the arrival times at vertices) are replaced in $\mathcal{F}_2$ by the flow variables f_{ij}, for all $(i,j) \in A$, which represent the amount of time still available for a vehicle after traversing the arc (i,j) as not to exceed T. As in $\mathcal{F}_1$, the slack variable φ represents the number of vehicles that are not used in the solution. $\mathcal{F}_2$ is defined as follows.

$$
\begin{aligned}
(\mathcal{F}_2) \quad \max \quad & \sum_{i \in P} p_i y_i, && (2.18)\\
s.t. \quad & \text{Constraints (2.2)-(2.8)} && \\
& f_{sj} = (T - d_{sj}) x_{sj} \qquad \forall\, j \in \delta^+(s), && (2.19)\\
& \sum_{j \in \delta^-(i)} f_{ji} - \sum_{j \in \delta^+(i)} f_{ij} = \sum_{j \in \delta^+(i)} d_{ij} x_{ij} \qquad \forall\, i \in S \cup P, && (2.20)\\
& f_{ij} \leq (T - R_{si} - d_{ij}) x_{ij} \qquad \forall (i,j) \in A,\ i \neq s, && (2.21)\\
& f_{ij} \geq R_{jt} x_{ij} \qquad \forall (i,j) \in A, && (2.22)\\
& x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A, && (2.23)\\
& y_i \in \{0,1\} \qquad \forall i \in N, && (2.24)\\
& f_{ij} \geq 0 \qquad \forall (i,j) \in A, && (2.25)\\
& 0 \leq \varphi \leq m. && (2.26)
\end{aligned}
$$

The objective function in (2.18) gives the total reward collected by visiting profitable vertices. Constraints (2.19)-(2.21) ensure that each of the solution routes has a total traverse time of at most T. Precisely, restrictions (2.19) implicitly state, along with (2.4), that the total flow available at the origin s is $(m - \varphi)T$, and, in particular, each vehicle (used) has an initial amount of T units of flow. Constraints (2.20) manage the flow consumption incurred from traversing the arcs selected, whereas constraints (2.21) impose that an arc $(i,j) \in A$ can only be traversed if the minimum time of a route from s to j through (i,j) does not exceed T. In (2.21), we do not consider the arcs leaving the origin, as they are already addressed by (2.19). Similarly to (2.12), the valid inequalities (2.22) give lower bounds on the flow passing through each arc, and constraints (2.23)-(2.26) define the domain of the variables. Here, the management of the flow associated with the variables f also avoids the existence of sub-tours in the solutions.

Notice that, in both formulations, the y variables can be easily discarded, as they solely aggregate specific subsets of the x variables. Nevertheless, they enable us to represent some families of valid inequalities (as detailed in Section 2.2) by means of less dense cuts, which can noticeably benefit the performance of cutting-plane algorithms.

Now, let $\mathcal{L}_1$ and $\mathcal{L}_2$ be the linearly relaxed versions of $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively.

Theorem 1. *$\mathcal{L}_1$ and $\mathcal{L}_2$ are equivalent.*

Proof. We show that, for every solution in $\mathcal{L}_1$, there is a corresponding one in $\mathcal{L}_2$ (and vice-versa), with a same objective function value associated. First, consider x and y as defined in $\mathcal{F}_1$ and $\mathcal{F}_2$, but without the integrality. Also recall that both formulations have the same objective function and that constraints (2.2)-(2.8) belong to $\mathcal{F}_1$ and to $\mathcal{F}_2$. Then, we only have to show that there also exists a correspondence between the remaining linear constraints which define $\mathcal{L}_1$ and $\mathcal{L}_2$. To this end, let us establish the following relation between z and f variables:

$$z_{ij} = Tx_{ij} - f_{ij} \qquad \forall (i,j) \in A. \tag{2.27}$$

From (2.27), it holds that

1. (2.9) $\iff$ (2.19)

$$\begin{aligned} z_{sj} = d_{sj}x_{sj} \qquad \forall\, j \in \delta^+(s) &\iff Tx_{sj} - f_{sj} = d_{sj}x_{sj} \qquad \forall\, j \in \delta^+(s) \\ &\iff f_{sj} = (T - d_{sj})x_{sj} \qquad \forall\, j \in \delta^+(s). \end{aligned}$$

2. (2.10) $\iff$ (2.20)

$$\begin{aligned} \sum_{j\in\delta^+(i)} z_{ij} - \sum_{j\in\delta^-(i)} z_{ji} &= \sum_{j\in\delta^+(i)} d_{ij}x_{ij} \qquad \forall\, i \in S \cup P \iff \\ \sum_{j\in\delta^+(i)} \Big(Tx_{ij} - f_{ij}\Big) - \sum_{j\in\delta^-(i)} \Big(Tx_{ji} - f_{ji}\Big) &= \sum_{j\in\delta^+(i)} d_{ij}x_{ij} \qquad \forall\, i \in S \cup P \iff \\ \sum_{j\in\delta^+(i)} Tx_{ij} - \sum_{j\in\delta^+(i)} f_{ij} - \sum_{j\in\delta^-(i)} Tx_{ji} + \sum_{j\in\delta^-(i)} f_{ji} &= \sum_{j\in\delta^+(i)} d_{ij}x_{ij} \qquad \forall\, i \in S \cup P \iff \\ \sum_{j\in\delta^-(i)} f_{ji} - \sum_{j\in\delta^+(i)} f_{ij} + T\Bigg(\sum_{j\in\delta^+(i)} x_{ij} - \sum_{j\in\delta^-(i)} x_{ji}\Bigg) &= \sum_{j\in\delta^+(i)} d_{ij}x_{ij} \qquad \forall\, i \in S \cup P, \end{aligned}$$

which, from (2.8), implies

$$\sum_{j\in\delta^-(i)} f_{ji} - \sum_{j\in\delta^+(i)} f_{ij} = \sum_{j\in\delta^+(i)} d_{ij}x_{ij} \qquad \forall\, i \in S \cup P.$$

3. (2.11) $\iff$ (2.22)

$$\begin{aligned} z_{ij} \le (T - R_{jt})x_{ij} \qquad \forall (i,j) \in A &\iff Tx_{ij} - f_{ij} \le (T - R_{jt})x_{ij} \qquad \forall (i,j) \in A \\ &\iff -f_{ij} \le (T - R_{jt})x_{ij} - Tx_{ij} \qquad \forall (i,j) \in A \\ &\iff f_{ij} \ge R_{jt}x_{ij} \qquad \forall (i,j) \in A. \end{aligned}$$

4. (2.12) $\implies$ (2.21)

$$\begin{aligned} z_{ij} \ge (R_{si} + d_{ij})x_{ij} \qquad \forall (i,j) \in A &\implies Tx_{ij} - f_{ij} \ge (R_{si} + d_{ij})x_{ij} \qquad \forall (i,j) \in A \\ &\implies f_{ij} \le (T - R_{si} - d_{ij})x_{ij} \qquad \forall (i,j) \in A \end{aligned}$$

$$\implies f_{ij} \leq (T - R_{si} - d_{ij})x_{ij} \qquad \forall (i,j) \in A,\, i \neq s.$$

5. (2.21) and (2.19) $\implies$ (2.12)

   From (2.21) and (2.19), we have that $f_{ij} \leq (T - R_{si} - d_{ij})x_{ij}$ for all $(i,j) \in A$, which implies

$$Tx_{ij} - f_{ij} \geq (R_{si} + d_{ij})x_{ij} \qquad \forall (i,j) \in A \implies z_{ij} \geq (R_{si} + d_{ij})x_{ij} \qquad \forall (i,j) \in A.$$

□

Proposition 1. *The inequality (2.17) does not cut off any solution from the polyhedron $\mathcal{L}_2$, the linear relaxation of $\mathcal{F}_2$.*

Proof. We prove this result by showing that (2.17) can be implied by linearly combining some of the linear constraints of $\mathcal{F}_2$. First, by aggregating all the constraints (2.20), we obtain

$$\overbrace{\sum_{i\in S\cup P}\left(\sum_{j\in\delta^-(i)} f_{ji} - \sum_{j\in\delta^+(i)} f_{ij}\right)}^{(a)} = \sum_{i\in S\cup P}\sum_{j\in\delta^+(i)} d_{ij}x_{ij}. \tag{2.28}$$

Now, let us define the set $\bar{A} = \{(i,j) \in A : i,j \in S \cup P\}$ composed of the arcs whose corresponding vertices are neither s nor t. Then, we can rewrite (a) as

$$\begin{aligned}\overbrace{\sum_{i\in S\cup P}\left(\sum_{j\in\delta^-(i)} f_{ji} - \sum_{j\in\delta^+(i)} f_{ij}\right)}^{(a)} &= \overbrace{\sum_{i\in S\cup P}\sum_{j\in\delta^-(i)} f_{ji}}^{(b)} - \overbrace{\sum_{i\in S\cup P}\sum_{j\in\delta^+(i)} f_{ij}}^{(c)} \\ &= \overbrace{\sum_{(j,i)\in\bar{A}} f_{ji} + \sum_{i\in\delta^+(s)\cap(S\cup P)} f_{si} + \sum_{i\in\delta^+(t)\cap(S\cup P)} f_{ti}}^{=(b)} \\ &\quad - \overbrace{\left(\sum_{(i,j)\in\bar{A}} f_{ij} + \sum_{i\in\delta^-(s)\cap(S\cup P)} f_{is} + \sum_{i\in\delta^-(t)\cap(S\cup P)} f_{it}\right)}^{=(c)}.\end{aligned} \tag{2.29}$$

From (2.6), (2.7), (2.19) and (2.21), we have that $\sum_{i\in\delta^+(t)\cap(S\cup P)} f_{ti} = \sum_{i\in\delta^-(s)\cap(S\cup P)} f_{is} = 0$. Also notice that $\sum_{(j,i)\in\bar{A}} f_{ji}$ turns into $\sum_{(i,j)\in\bar{A}} f_{ij}$ (and vice-versa) by simply reordering the notation. Thus, (2.29) can be rewritten as

$$\overbrace{\sum_{i\in S\cup P}\left(\sum_{j\in\delta^-(i)} f_{ji} - \sum_{j\in\delta^+(i)} f_{ij}\right)}^{(a)} = \sum_{i\in\delta^+(s)\cap(S\cup P)} f_{si} - \sum_{i\in\delta^-(t)\cap(S\cup P)} f_{it}. \tag{2.30}$$

Directly from (2.28) and (2.30), it follows that

$$\begin{aligned}\sum_{i\in S\cup P}\sum_{j\in\delta^+(i)} d_{ij}x_{ij} &= \sum_{i\in\delta^+(s)\cap(S\cup P)} f_{si} - \sum_{i\in\delta^-(t)\cap(S\cup P)} f_{it}\\ &\leq \sum_{i\in\delta^+(s)\cap(S\cup P)} f_{si}\\ &\leq \sum_{i\in\delta^+(s)\cap(S\cup P)} f_{si} + \sum_{i\in\delta^+(s)\setminus(S\cup P)} f_{si} = \sum_{i\in\delta^+(s)} f_{si}. \end{aligned}\tag{2.31}$$

From (2.19) and (2.31), we have that

$$\begin{aligned}\sum_{i\in S\cup P}\sum_{j\in\delta^+(i)} d_{ij}x_{ij} &\leq \sum_{i\in\delta^+(s)} f_{si}\\ &= \sum_{i\in\delta^+(s)} (T-d_{si})x_{si}\\ &= T\Bigg(\sum_{i\in\delta^+(s)} x_{si}\Bigg) - \sum_{i\in\delta^+(s)} d_{si}x_{si},\end{aligned}\tag{2.32}$$

which implies

$$\overbrace{\sum_{i\in S\cup P}\sum_{j\in\delta^+(i)} d_{ij}x_{ij} + \sum_{i\in\delta^+(s)} d_{si}x_{si}}^{(d)} \leq T\Bigg(\sum_{i\in\delta^+(s)} x_{si}\Bigg).\tag{2.33}$$

Notice that (d) corresponds to

$$\begin{aligned}\overbrace{\sum_{i\in S\cup P}\sum_{j\in\delta^+(i)} d_{ij}x_{ij} + \sum_{i\in\delta^+(s)} d_{si}x_{si}}^{(d)} &= \sum_{i\in S\cup P\cup\{s\}}\sum_{j\in\delta^+(i)} d_{ij}x_{ij}\\ &= \sum_{i\in N\setminus\{t\}}\sum_{j\in\delta^+(i)} d_{ij}x_{ij} = \sum_{(i,j)\in A} d_{ij}x_{ij},\end{aligned}\tag{2.34}$$

since, from (2.7), no arc leaving t can be selected. Then, from (2.4), (2.26), (2.33) and (2.34), it follows that

$$\overbrace{\sum_{(i,j)\in A} d_{ij}x_{ij}}^{=(d)} \leq T\Bigg(\sum_{j\in\delta^+(s)} x_{sj}\Bigg) = (m-\varphi)T \leq mT. \qquad \square$$

Corollary 1. *The inequality (2.17) does not cut off any solution from the polyhedron $\mathcal{L}_1$, the linear relaxation of $\mathcal{F}_1$.*

Proof. Directly from Theorem 1 and Proposition 1. □

As already mentioned, constraints (2.12) and (2.22) are, in fact, valid inequalities in formulations $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively. Accordingly, one can take advantage from this fact when solving these formulations by means of branch-and-cut schemes that, like CPLEX, have cut management mechanisms. Precisely, in such schemes, valid inequalities — usually referred to as *user cuts* — are treated differently from actual model restrictions, as they are stored in *pools* of cuts and only added to the models whenever they are violated.

In the case of $\mathcal{F}_2$, one can particularly benefit from cut management mechanisms. Precisely, we experimentally observed that, on average, the impact of the valid inequalities (2.22) on the strength of $\mathcal{F}_2$ is not as expressive as that of inequalities (2.12) on the strength of $\mathcal{F}_1$ (we refer to A.1 for the summary of the results). In practice, this behaviour suggests that (2.22) are less likely to be active at optimal solutions for $\mathcal{F}_2$. Then, instead of treating inequalities (2.22) as constraints of $\mathcal{F}_2$, we can explicitly define them as user cuts as an attempt to make the corresponding models lighter while not losing the strength of the original formulation. This simple idea was originally proposed by Fischetti et al. (1998) and is our main motivation for solving $\mathcal{F}_2$ — and not $\mathcal{F}_1$ — within our cutting-plane algorithm.

Originally, we also proposed and tested several other compact formulations for the STOP. Precisely, we tested two-commodity and multi-commodity versions of the formulations $\mathcal{F}_1$ and $\mathcal{F}_2$, as well as variations in which the x variables are indexed by the vehicles of the fleet M. Likewise, we also considered classical commodity-based formulations in which the consumption of each unit of commodity is linked to visiting a single vertex. In addition, we adapted to the STOP the formulation for the CVRP proposed by Kulkarni and Bhave (1985), which uses the reinforced Miller-Tucker-Zemlin (Miller et al., 1960) subtour elimination constraints of Kara et al. (2004). Nevertheless, since all of these additional formulations performed poorly (when solved directly with CPLEX) in comparison with $\mathcal{F}_1$ and $\mathcal{F}_2$, they were omitted from this study. In fact, we conjecture that the superiority of $\mathcal{F}_1$ and $\mathcal{F}_2$ is partially due to the way they implicitly handle the limit imposed on the total traverse times of the routes.

2.2 Families of valid inequalities

In this section, we discuss five families of valid inequalities to be separated in the cutting-plane scheme we propose. They consist of (i) some general connectivity constraints, (ii) a class of conflict cuts, (iii) clique conflict cuts, (iv) arc-vertex inference cuts and (v) classical lifted cover inequalities based on dual bounds. As far as we are aware, the classes of inequalities (ii), (iii) and (iv) are also introduced in this work.

2.2.1 General Connectivity Constraints (GCCs)

The GCC inequalities were originally devised to ensure connectivity and prevent sub-tours in solution routes (Toth and Vigo, 2001). Although these properties are already guaranteed in formulations $\mathcal{F}_1$ and $\mathcal{F}_2$, the GCCs presented below are able to further strengthen both formulations (Bianchessi et al., 2018).

$$\sum_{(i,j)\in\delta^+(V)} x_{ij} \geq y_k \qquad \forall\, V \subseteq N\backslash\{t\},\, |V| \geq 2,\, \forall\, k \in V. \tag{2.35}$$

2.2.2 Conflict Cuts (CCs)

Consider the set $\mathcal{K}$ of vertex pairs which cannot be simultaneously in a same valid route. Precisely, for every pair of *conflicting vertices* $\langle i, j \rangle \in \mathcal{K}$, with $i, j \in N \backslash \{s, t\}$, we have that any route from s to t that visits i and j (in any order) has a total traverse time that exceeds the limit T. Then, CCs are defined as follows.

$$\sum_{e \in \delta^-(V)} x_e \geq y_i + y_j \qquad \forall \langle i, j \rangle \in \mathcal{K},\ \forall\, V \subseteq N \backslash \{s\},\ \{i, j\} \subseteq V, \tag{2.36}$$

$$\sum_{e \in \delta^+(V)} x_e \geq y_i + y_j \qquad \forall \langle i, j \rangle \in \mathcal{K},\ \forall\, V \subseteq N \backslash \{t\},\ \{i, j\} \subseteq V. \tag{2.37}$$

Proposition 2. *Inequalities (2.36) do not cut off any feasible solution of $\mathcal{F}_2$.*

Proof. Consider an arbitrary feasible solution $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$ for $\mathcal{F}_2$, a pair of conflicting vertices $\langle i, j \rangle \in \mathcal{K}$ and a subset $V \subseteq N \backslash \{s\}$. Then, we have four possibilities:

1. if $\bar{y}_i = \bar{y}_j = 0$, then $\overbrace{\sum_{e \in \delta^-(V)} x_e}^{\geq 0} \geq \overbrace{y_i + y_j}^{=0}$.

2. if $\bar{y}_i = 1$ and $\bar{y}_j = 0$, then, since $s \notin V$, there must be an arc $e' \in \delta^-(V)$, with $\bar{x}_{e'} = 1$, so that the vertex i is traversed in a route from s. Thus, $\overbrace{\sum_{e \in \delta^-(V)} x_e}^{\geq 1} \geq \overbrace{y_i + y_j}^{=1}$.

3. if $\bar{y}_i = 0$ and $\bar{y}_j = 1$, then, since $s \notin V$, there must be an arc $e' \in \delta^-(V)$, with $\bar{x}_{e'} = 1$, so that the vertex j is traversed in a route from s. Thus, $\overbrace{\sum_{e \in \delta^-(V)} x_e}^{\geq 1} \geq \overbrace{y_i + y_j}^{=1}$.

4. if $\bar{y}_i = \bar{y}_j = 1$, then, since i and j are conflicting, there must be at least two disjoint routes from s to t, one that visits i, and other that visits j. Since $s \notin V$, we must also have at least two arcs $e', e'' \in \delta^-(V)$, such that $x_{e'} = x_{e''} = 1$. Therefore, $\overbrace{\sum_{e \in \delta^-(V)} x_e}^{\geq 2} \geq \overbrace{y_i + y_j}^{=2}$.

□

Corollary 2. *Inequalities (2.37) do not cut off any feasible solution of $\mathcal{F}_2$.*

Proof. The same mathematical argumentation of Proposition 2 can be used to prove the validity of inequalities (2.37) by simply replacing s and $\delta^-(V)$ with t and $\delta^+(V)$, respectively. □

Corollary 3. *Inequalities (2.36) and (2.37) do not cut off any feasible solution of $\mathcal{F}_1$.*

Proof. Directly from Theorem 1, Proposition 2 and Corollary 2. □

Notice that, from (2.8), we have that $\sum_{e \in \delta^-(V)} x_e = \sum_{e \in \delta^+(V)} x_e$ for all $V \subseteq S \cup P = N \backslash \{s, t\}$. Therefore, for all $V \subseteq N \backslash \{s, t\}$, inequalities (2.36) and (2.37) cut off the exact same regions of

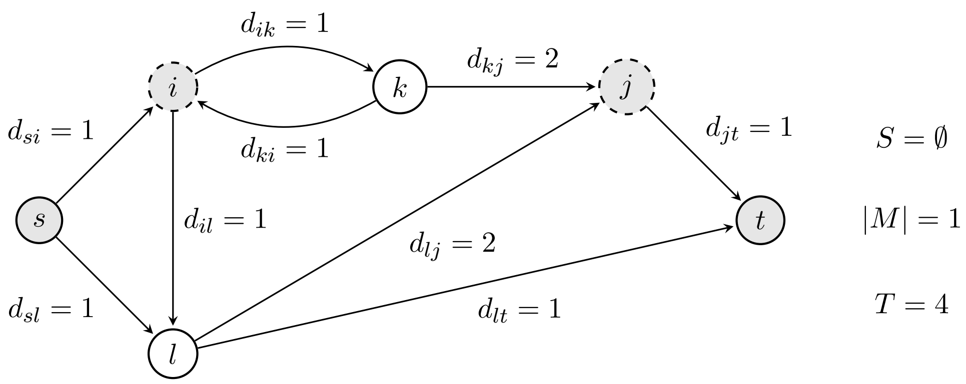


Figure 2.1: An example of an STOP instance. Profit values are omitted. Here, the pair $\langle i, j \rangle$ gives an example of conflicting vertices.

the polyhedrons $\mathcal{L}_1$ and $\mathcal{L}_2$. In this sense, the whole set of CCs can be represented in a more compact manner by (2.36) and

$$\sum_{e \in \delta^+(V)} x_e \geq y_i + y_j \qquad \forall \langle i, j \rangle \in \mathcal{K},\ \forall V \subseteq N \backslash \{t\},\ \{s, i, j\} \subseteq V. \tag{2.38}$$

Notice that inequalities (2.38) only consider the subsets $V \subseteq N \backslash \{t\}$ which necessarily contain s.

Intuitively speaking, CCs forbid the simultaneous selection (in a same route) of any pair of conflicting vertices of $\mathcal{K}$. In particular, CCs (2.36) work similarly to the classical *capacity-cut constraints* of Toth and Vigo (2001), but in a more flexible manner. Precisely, given a pair of conflicting vertices $\langle i, j \rangle \in \mathcal{K}$ and a subset $V \subseteq N \backslash \{s\}$, $\{i, j\} \subseteq V$, the corresponding CC of type (2.36) states that the minimum number of vehicles needed to visit i and j is exactly $y_i + y_j$ (which assumes, at most, value 2). Alternatively, $y_i + y_j$ can be seen as a lower bound on the number of vehicles needed to visit all the vertices in V.

For instance, consider the digraph shown in Figure 2.1, and let the traverse times of its arcs be $d_{si} = d_{sl} = d_{il} = d_{ik} = d_{ki} = d_{lt} = d_{jt} = 1$ and $d_{kj} = d_{lj} = 2$. Also consider $S = \emptyset$ and a single vehicle to move from s to t, with $T = 4$. In this case, i and j are an example of conflicting vertices, since the only possible route from s to t visiting both vertices exceeds the time limit $T = 4$. Figures 2.2 and 2.3 show typical fractional solutions that are cut off by CCs (2.36), while Figure 2.4 gives a solution that is cut off by (2.38). Notice that the solution of Figure 2.2 also violates CCs (2.37).

2.2.3 Clique Conflict Cuts (CCCs)

Now, consider the conflict graph $G_c = (N \backslash \{s, t\}, E_c)$ representing the pairs of conflicting vertices in $\mathcal{K}$, such that $E_c = \{\{i, j\} : \langle i, j \rangle \in \mathcal{K}\}$ (Figure 2.5 shows the conflict graph related to the STOP instance of Figure 2.1). Also let Σ be the set of all the cliques of G_c, which are referred to as *conflict cliques*. Then, CCCs are defined as

$$\sum_{e \in \delta^-(V)} x_e \geq \sum_{i \in \sigma} y_i \qquad \forall \sigma \in \Sigma,\ \forall V \subseteq N \backslash \{s\},\ \sigma \subseteq V, \tag{2.39}$$

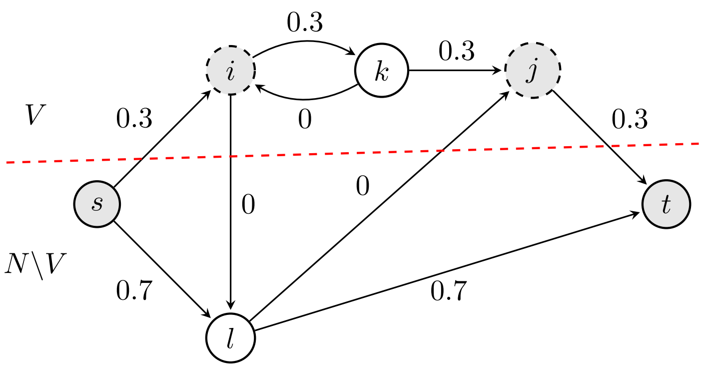


Figure 2.2: Representation of a fractional solution that is cut off from $\mathcal{L}_2$ by (2.36) when considering the polyhedron $\mathcal{L}_2$ and the STOP instance of Figure 2.1. Here, we have $x_{si} = x_{ik} = x_{kj} = x_{jt} = 0.3$, $x_{sl} = x_{lt} = 0.7$ and $x_{il} = x_{ki} = x_{lj} = 0$. Accordingly, $y_s = y_t = 1$, $y_i = y_k = y_j = 0.3$ and $y_l = 0.7$. The violated inequality has $V = \{i, k, j\}$.

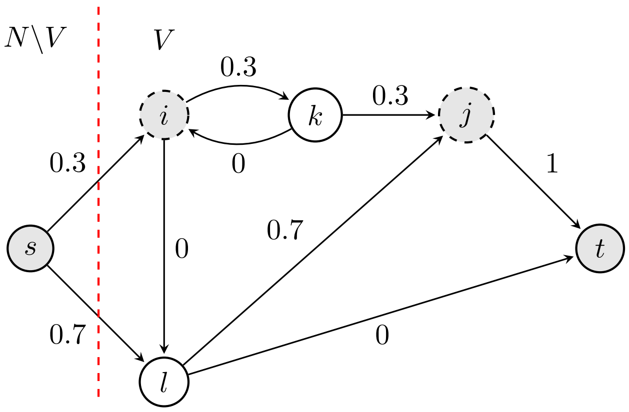


Figure 2.3: Representation of a fractional solution that is cut off from $\mathcal{L}_2$ by (2.36) when considering the polyhedron $\mathcal{L}_2$ and the STOP instance of Figure 2.1. Here, we have $x_{si} = x_{ik} = x_{kj} = 0.3$, $x_{sl} = x_{lj} = 0.7$, $x_{il} = x_{ki} = x_{lt} = 0$ and $x_{jt} = 1$. Accordingly, $y_s = y_j = y_t = 1$, $y_i = y_k = 0.3$ and $y_l = 0.7$. The violated inequality has $V = \{i, k, j, l, t\}$.

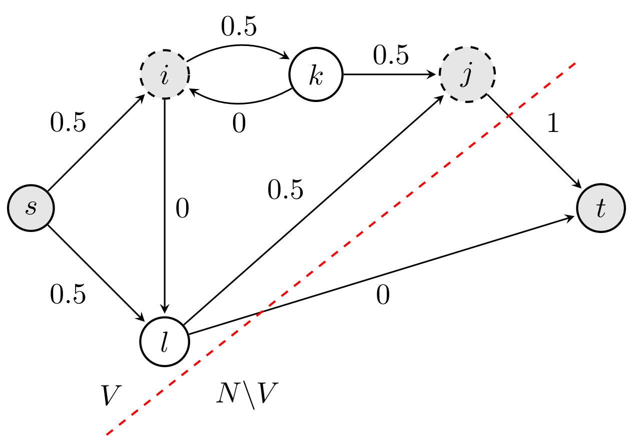


Figure 2.4: Example of a fractional solution that is cut off by CCs (2.38) when considering the polyhedron $\mathcal{L}_2$ and the STOP instance of Figure 2.1. Here, we have $x_{si} = x_{ik} = x_{kj} = x_{sl} = x_{lj} = 0.5$, $x_{il} = x_{ki} = x_{lt} = 0$ and $x_{jt} = 1$. Accordingly, $y_s = y_j = y_t = 1$, $y_i = y_k = y_l = 0.5$. The violated inequality has $V = \{s, l, i, k, j\}$.

$$\sum_{e \in \delta^+(V)} x_e \geq \sum_{i \in \sigma} y_i \qquad \forall\, \sigma \in \Sigma,\ \forall\, V \subseteq N \backslash \{t\},\ \sigma \subseteq V. \tag{2.40}$$

Their validity follows from the fact that all the vertices of a conflict clique must necessarily belong to different routes in any feasible solution for $\mathcal{F}_1$ and $\mathcal{F}_2$. The formal proof can be easily devised through the same argumentation of Proposition 2. Notice that CCCs are a natural extension of GCCs and CCs. In particular, GCCs and CCs are CCCs based on cliques of sizes one and two, respectively.

Remark 1. *The set of CCCs defined by (2.39) and (2.40) contains all GCCs and CCs.*

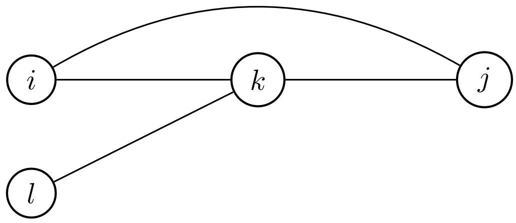


Figure 2.5: Conflict graph related to the STOP instance of Figure 2.1.

Also in this case, from (2.8), we have that $\sum_{e \in \delta^-(V)} x_e = \sum_{e \in \delta^+(V)} x_e$ for all $V \subseteq S \cup P = N \backslash \{s, t\}$. Therefore, for all $V \subseteq N \backslash \{s, t\}$, inequalities (2.39) and (2.40) cut off the exact same regions of the polyhedrons $\mathcal{L}_1$ and $\mathcal{L}_2$. In this sense, the whole set of CCCs can be represented in a more compact manner by (2.39) and

$$\sum_{e \in \delta^+(V)} x_e \geq \sum_{i \in \sigma} y_i \qquad \forall\, \sigma \in \Sigma,\ \forall\, V \subseteq N \backslash \{t\},\ \sigma \cup \{s\} \subseteq V. \tag{2.41}$$

Theorem 2. *For every non-maximal conflict clique $\sigma_1 \in \Sigma$, there is a clique $\sigma_2 \in \Sigma$, with $\sigma_1 \subset \sigma_2$, such that the CCCs referring to σ_1 are dominated by the ones referring to σ_2, when considering the polyhedron $\mathcal{L}_2$.*

Proof. Consider an arbitrary non-maximal conflict clique $\sigma_1 \in \Sigma$ and, without loss of generality, a solution $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$ for $\mathcal{L}_2$ that is violated by at least one of the CCCs referring to σ_1. We have to show that there exists a clique $\sigma_2 \in \Sigma$, with $\sigma_1 \subset \sigma_2$, whose corresponding CCCs also cut off $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$ from $\mathcal{L}_2$. To this end, consider a vertex $k \in (S \cup P) \backslash \sigma_1$, such that $\{k, j\} \in E_c$ for all $j \in \sigma_1$. Such vertex exists, since σ_1 is supposed to be non-maximal. Then, define the conflict clique $\sigma_2 = \sigma_1 \cup \{k\}$.

By assumption, $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$ violates at least one of the CCCs (2.39) and (2.40) referring to σ_1. Then, we must have at least one of these two possibilities:

1. $\exists\, V' \subseteq N \backslash \{s\}$, with $\sigma_1 \subseteq V'$, such that $\sum_{e \in \delta^-(V')} \bar{x}_e < \sum_{i \in \sigma_1} \bar{y}_i$.

   Define the set $V'' = V' \cup \{k\}$. Notice that, if we originally have $k \in V'$, then the CCC (2.39) considering σ_2 and $V = V'(= V''$ in this case) is also violated by $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$.

Otherwise, if $k \notin V'$, then

$$\sum_{e \in \delta^-(V'')} \bar{x}_e = \sum_{e \in \delta^-(V')} \bar{x}_e + \overbrace{\sum_{i \in \delta^-(k) \setminus V'} \bar{x}_{ik}}^{(g)} - \overbrace{\sum_{j \in \delta^+(k) \cap V'} \bar{x}_{kj}}^{(h)}, \tag{2.42}$$

which follows from the fact that the difference between the summation of arcs entering V'' and that of arcs entering V' corresponds to the difference between (g) the sum of arcs arriving at k that do not leave a vertex of V' and (h) the sum of arcs leaving k that arrive at a vertex of V'. In other words, (g) considers the arcs that traverse the cut $(N \setminus V'', V'')$ but do not traverse $(N \setminus V', V')$, while (h) considers the arcs that do not traverse the cut $(N \setminus V'', V'')$ but traverse $(N \setminus V', V')$.

From (2.42) and the hypothesis that $\sum_{e \in \delta^-(V')} \bar{x}_e < \sum_{i \in \sigma_1} \bar{y}_i$, it follows that

$$\sum_{e \in \delta^-(V'')} \bar{x}_e < \sum_{i \in \sigma_1} \bar{y}_i + \overbrace{\sum_{i \in \delta^-(k) \setminus V'} \bar{x}_{ik}}^{(g)} - \overbrace{\sum_{j \in \delta^+(k) \cap V'} \bar{x}_{kj}}^{(h)}. \tag{2.43}$$

From (2.3), (2.8), (2.23) and (2.24), we have that

$$\overbrace{\sum_{i \in \delta^-(k) \setminus V'} \bar{x}_{ik}}^{(g)} - \overbrace{\sum_{j \in \delta^+(k) \cap V'} \bar{x}_{kj}}^{(h)} \leq \bar{y}_k. \tag{2.44}$$

Then, from (2.43), it follows that

$$\sum_{e \in \delta^-(V'')} \bar{x}_e < \sum_{i \in \sigma_1} \bar{y}_i + \bar{y}_k \tag{2.45}$$

$$< \sum_{i \in \sigma_2} \bar{y}_i, \tag{2.46}$$

Since $\sigma_2 = \sigma_1 \cup \{k\}$ and $\sigma_1 \subseteq V'$, we have that $\sigma_2 \subseteq V''$. Then, also in this case, there exists a CCC (2.39) referring to σ_2 (in particular, with $V = V''$) that is also violated by $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$.

2. $\exists\, V' \subseteq N \setminus \{t\}$, with $\sigma_1 \subseteq V'$, such that $\sum_{e \in \delta^+(V')} \bar{x}_e < \sum_{i \in \sigma_1} \bar{y}_i$.

   Through the same idea of the previous case, we can also show that the CCC (2.40) referring to σ_2 that considers $V = V' \cup \{k\}$ is also violated by $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$.

□

From Theorem 2, we may discard several CCCs (2.39) and (2.40), as we only need to consider the ones related to maximal conflict cliques. We highlight that, in the conflict graph, there might be maximal cliques of size one, which correspond to isolated vertices.

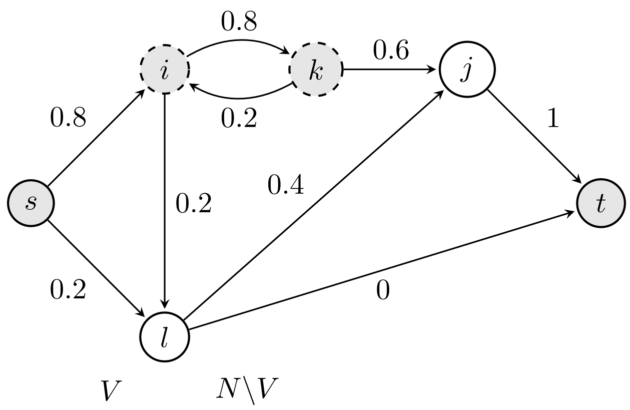


Figure 2.6: Example of a fractional solution that is cut off by AVICs (2.48) and (2.49) when considering the polyhedron $\mathcal{L}_2$ and the STOP instance of Figure 2.1. Here, we have $x_{si} = x_{ik} = 0.8$, $x_{sl} = x_{il} = x_{ki} = 0.2$, $x_{lj} = 0.4$, $x_{kj} = 0.6$, $x_{lt} = 0$ and $x_{jt} = 1$. Accordingly, $y_s = y_i = y_j = y_t = 1$, $y_k = 0.8$ and $y_l = 0.4$. The violated AVIC is $y_i - y_k \leq 1 - (x_{ik} + x_{ki})$.

2.2.4 Arc-Vertex Inference Cuts (AVICs)

Consider the set $E = \{\{i, j\} : (i, j) \in A \text{ and } (j, i) \in A\}$. AVICs are defined as

$$|y_i - y_j| \leq 1 - (x_{ij} + x_{ji}) \qquad \forall \{i, j\} \in E, \tag{2.47}$$

which can be linearized as

$$y_i - y_j \leq 1 - (x_{ij} + x_{ji}) \qquad \forall \{i, j\} \in E, \text{ and} \tag{2.48}$$

$$y_j - y_i \leq 1 - (x_{ij} + x_{ji}) \qquad \forall \{i, j\} \in E. \tag{2.49}$$

In logic terms, these inequalities correspond to the boolean expressions

$$(x_{ij} = 1 \oplus x_{ji} = 1) \rightarrow y_i - y_j = 0 \qquad \forall \{i, j\} \in E,$$

where $\oplus$ stands for the *exclusive disjunction* operator.

The validity of AVICs relies on two trivial properties that are inherent to feasible solutions for $\mathcal{F}_1$ and $\mathcal{F}_2$: (i) for all $\{i, j\} \in E$, arcs (i, j) and (j, i) cannot be simultaneously selected (i.e. $x_{ij} + x_{ji} \leq 1$), and (ii) once an arc $(i, j) \in A$ is selected in a solution (i.e., $x_{ij} = 1$), we must have $y_i = y_j$ (which, more precisely, are also equal to one). Figure 2.6 gives an example of a fractional solution that is cut off from $\mathcal{L}_2$ (and $\mathcal{L}_1$) by AVICs.

One may notice that simpler valid inequalities can be devised by considering arcs separately, as follows

$$|y_i - y_j| \leq 1 - x_{ij} \qquad \forall (i, j) \in A. \tag{2.50}$$

However, these inequalities are not only weaker, but redundant for formulations $\mathcal{F}_1$ and $\mathcal{F}_2$, as proven next.

Proposition 3. *Inequalities (2.50) do not cut off any solution from the polyhedrons $\mathcal{L}_1$ and $\mathcal{L}_2$.*

Proof. From (2.3), (2.8) and the domain of the binary variables x and y, we have that $0 \leq x_{ij} \leq y_i \leq 1$ and $0 \leq x_{ij} \leq y_j \leq 1$ for all $(i,j) \in A$. Accordingly, the maximum possible value assumed by $|y_i - y_j|$ occurs when one of the corresponding y variables assumes its minimum value (i.e., x_{ij}), and the other assumes its maximum (i.e., 1). □

2.2.5 Lifted Cover Inequalities (LCIs)

Let τ be a dual (upper) bound on the optimal solution value of $\mathcal{F}_2$. Then, consider the inequality

$$\sum_{i \in P} p_i y_i \leq \lfloor \tau \rfloor, \tag{2.51}$$

where $P \subseteq N \backslash \{s,t\}$ is the set of profitable vertices, with $p_i \in \mathbb{Z}^+$ for all $i \in P$, as defined in Section 1.2. By definition, (2.51) is valid for $\mathcal{F}_2$ (and $\mathcal{F}_1$), once its left-hand side corresponds to the objective function of this formulation.

Notice that (2.51) consists of a *knapsack* constraint, and, thus, it can be strengthened by means of classical cover inequalities and *lifting*. In this sense, a set $C \subseteq P$ is called a *cover* for inequality (2.51) if $\sum\limits_{i \in C} p_i > \lfloor \tau \rfloor$. Moreover, this cover is said to be *minimal* if it no longer covers (2.51) once any of its elements is removed, i.e., $\sum\limits_{i \in C \backslash \{j\}} p_i \leq \lfloor \tau \rfloor$ for all $j \in C$.

Consider the set Φ of the y solution values that satisfy (2.51). Precisely,

$$\Phi = \Big\{ y \in \{0,1\}^{|P|} : \sum_{i \in P} p_i y_i \leq \lfloor \tau \rfloor \Big\}. \tag{2.52}$$

For any cover $C \subseteq P$, the inequality

$$\sum_{i \in C} y_i \leq |C| - 1 \tag{2.53}$$

is called a *cover inequality* and is valid for Φ. In this work, we apply a variation of these inequalities, namely LCIs, which are facet-inducing for Φ and can be devised from (2.53) through lifting (see, e.g., Balas (1975); Wolsey (1975); Gu et al. (1998); Kaparis and Letchford (2008)). Let two disjoint sets C_1 and C_2 define a partition of a given minimal cover C, with $C_1 \neq \emptyset$. LCIs are defined as

$$\sum_{i \in C_1} y_i + \sum_{j \in C_2} \pi_j y_j + \sum_{j \in P \backslash C} \mu_j y_j \leq |C_1| + \sum_{j \in C_2} \pi_j - 1, \tag{2.54}$$

where $\pi_i \in \mathbb{Z}$, $\pi_i \geq 1$ for all $i \in C_2$, and $\mu_i \in \mathbb{Z}$, $\mu_i \geq 0$ for all $i \in P \backslash C$, are the lifted coefficients. We detail the way these coefficients are computed in the separation procedure of Section 2.3.5. For now, only notice that setting $\pi = \mathbf{1}$ and $\mu = \mathbf{0}$ suffices for the validity of (2.54), as it leads to the classical cover inequality (2.53).

2.3 Separation of valid inequalities

Let $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$ (or $(\bar{x}, \bar{y}, \bar{z}, \bar{\varphi})$) be a given fractional solution referring to the linear relaxation of $\mathcal{F}_2$ (or $\mathcal{F}_1$). Also consider the residual graph $\tilde{G} = (N, \tilde{A})$ induced by $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$, such that each arc $(i, j) \in A$ belongs to $\tilde{A}$ if, and only if, $\bar{x}_{ij} > 0$. Moreover, a capacity $c[i, j] = \bar{x}_{ij}$ is associated with each arc $(i, j) \in \tilde{A}$.

As detailed in the sequel, the separation of GCCs and CCs involves solving maximum flow problems. Then, for clarity, consider the following notation. Given an arbitrary digraph G_a with capacitated arcs, and two vertices i and j of G_a, let $\textit{max-flow}_{i\to j}(G_a)$ denote the problem of finding the maximum flow (and, thus, a minimum cut) from i to j on G_a. Moreover, let $\langle F_{i\to j}, \theta_{i\to j}\rangle$ denote an optimal solution of such problem, where $F_{i\to j}$ is the value of the maximum flow, and $\theta_{i\to j}$ defines a corresponding minimum cut, with $i \in \theta_{i\to j}$.

2.3.1 GCCs

The separation of violated GCCs is done by means of the same algorithm adopted by Bianchessi et al. (2018) and described as follows. For each pair of vertices $\langle v, t\rangle$, $v \in N\backslash\{t\}$, a maximum flow from v to t on $\tilde{G}$ is computed, i.e., the problem $\textit{max-flow}_{v\to t}(\tilde{G})$ is solved, obtaining a solution $\langle F_{v\to t}, \theta_{v\to t}\rangle$. Let $v^* = \arg \max_{j\in\theta_{v\to t}} \{\bar{y}_j\}$. If $t \notin \theta_{v\to t}$, $|\theta_{v\to t}| \geq 2$ and $\bar{y}_{v^*}$ is greater than the value of the maximum flow $F_{v\to t}$, then a violated GCC is found, which corresponds to

$$\sum_{(i,j)\in\delta^+(\theta_{v\to t})} x_{ij} \geq y_{v^*}. \tag{2.55}$$

Notice that, in this case, the GCCs found (if any) are the most violated ones, one for each pair $\langle v, t\rangle$, $v \in N\backslash\{t\}$.

2.3.2 CCs

Considering the shortest (minimum time) paths matrix R defined by the end of Section 1.2, we first compute the set $\mathcal{K}$ of conflicting vertices by checking, for all pairs $\langle i, j\rangle$, $i, j \in N$, $i \neq j$, if there exists a path from s to t on G that traverses both i and j (in any order) and that satisfies the total time limit T. If no such path exists, then $\langle i, j\rangle$ belongs to $\mathcal{K}$. For simplicity, in this work, we only consider a subset $\tilde{\mathcal{K}} \subseteq \mathcal{K}$ of conflicting vertex pairs, such that

$$\langle i, j\rangle \in \tilde{\mathcal{K}} \text{ iff } \begin{cases} (i)\ R_{si} + R_{ij} + R_{jt} > T, \text{ and} \\ (ii)\ R_{sj} + R_{ji} + R_{it} > T, \end{cases} \qquad \forall\, i, j \in N, i \neq j,$$

where (i) is satisfied if a minimum traverse time route from s to t that visits i before j exceeds the time limit. Likewise, (ii) considers a minimum time route that visits j before i. Since the routes from s to t considered in (i) and (ii) are composed by simply aggregating entries of $\mathcal{M}$, they may not be elementary, i.e., they might visit a same vertex more than once. Then, $\tilde{\mathcal{K}}$ is not necessarily equal to $\mathcal{K}$. Also observe that we only have to compute $\tilde{\mathcal{K}}$ a single time for a given STOP instance, as it is completely based on the original graph G.

Once $\tilde{\mathcal{K}}$ is computed, we look for violated CCs of types (2.36) and (2.37) separately, as described in the algorithm of Figure 2.7. Let the set $\mathcal{X}$ keep the CCs found during the separation procedure. Initially, $\mathcal{X}$ is empty (line 1, Figure 2.7). Then, for all pairs of conflicting vertices $\langle i, j\rangle \in \tilde{\mathcal{K}}$, we build two auxiliary graphs, one for each type of CC. The first graph, denoted by $\tilde{G}_1$, is built by adding to the residual graph $\tilde{G}$ an artificial vertex β_1 and two arcs: (i, β_1) and (j, β_1) (see line 4, Figure 2.7). The second one, denoted by $\tilde{G}_2$, is built by reversing all the arcs of $\tilde{G}$ and, then, adding an artificial vertex β_2, as well as the arcs (i, β_2) and (j, β_2) (see line 5, Figure 2.7).

The capacities of the arcs of $\tilde{G}_1$ and $\tilde{G}_2$ are kept in the data structures c_1 and c_2, respectively. In both graphs, the capacities of the original arcs in $\tilde{G}$ are preserved (see lines 6-8, Figure 2.7). Moreover, all the additional arcs have a same capacity value, which is equal to a sufficiently large number. Here, we adopted the value of $|M|$, the number of vehicles (see line 9, Figure 2.7).

Figures 2.8 and 2.9 illustrate the construction of the auxiliary graphs described above. In these examples, we consider the STOP instance of Figure 2.1 and assume that the current fractional solution $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$ has $\bar{x}_{si} = \bar{x}_{ik} = \bar{x}_{kj} = \bar{x}_{sl} = \bar{x}_{lj} = 0.5$, $\bar{x}_{ki} = \bar{x}_{il} = \bar{x}_{lt} = 0$ and $\bar{x}_{jt} = 1$.

Input: A fractional solution $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$, its corresponding residual graph $\tilde{G} = (N, \tilde{A})$ and the subset $\tilde{\mathcal{K}}$ of conflicting vertex pairs.
Output: A set $\mathcal{X}$ of CCs violated by $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$.
1. $\mathcal{X} \leftarrow \emptyset$;
2. **for all** $(\langle i, j\rangle \in \tilde{\mathcal{K}})$ **do**
3. *Step I. Building the auxiliary graphs*
4. Build $\tilde{G}_1 = (\tilde{N}_1, \tilde{A}_1)$, with $\tilde{N}_1 = N \cup \{\beta_1\}$ and $\tilde{A}_1 = \tilde{A} \cup \{(i, \beta_1), (j, \beta_1)\}$;
5. Build $\tilde{G}_2 = (\tilde{N}_2, \tilde{A}_2)$, with $\tilde{N}_2 = N \cup \{\beta_2\}$ and $\tilde{A}_2 = \{(v, u) : (u, v) \in \tilde{A}\} \cup \{(i, \beta_2), (j, \beta_2)\}$;
6. **for all** $((u, v) \in \tilde{A})$ **do**
7. $c_1[u, v] \leftarrow c_2[v, u] \leftarrow c[u, v]$;
8. **end-for**;
9. $c_1[i, \beta_1] \leftarrow c_1[j, \beta_1] \leftarrow c_2[i, \beta_2] \leftarrow c_2[j, \beta_2] \leftarrow |M|$;
10. *Step II. Looking for a violated CC (2.36)*
11. $\langle F_{s\to\beta_1}, \theta_{s\to\beta_1}\rangle \leftarrow$ *max-flow*$_{s\to\beta_1}(\tilde{G}_1)$;
12. **if** $(F_{s\to\beta_1} < \bar{y}_i + \bar{y}_j)$ **then**
13. $\mathcal{X} \leftarrow \mathcal{X} \cup \{\langle N\backslash\theta_{s\to\beta_1}, \langle i, j\rangle\rangle\}$;
14. **end-if**;
15. *Step III. Looking for a violated CC (2.37)*
16. $\langle F_{t\to\beta_2}, \theta_{t\to\beta_2}\rangle \leftarrow$ *max-flow*$_{t\to\beta_2}(\tilde{G}_2)$;
17. **if** $(F_{t\to\beta_2} < \bar{y}_i + \bar{y}_j)$ **then**
18. $\mathcal{X} \leftarrow \mathcal{X} \cup \{\langle N\backslash\theta_{t\to\beta_2}, \langle i, j\rangle\rangle\}$;
19. **end-if**;
20. **end-for**;
21. **return** $\mathcal{X}$;

Figure 2.7: Algorithm used to separate violated CCs.

Once the auxiliary graphs are built for a given $\langle i, j\rangle \in \tilde{\mathcal{K}}$, the algorithm looks for violated CCs by solving two maximum flow problems: one from s to β_1 on $\tilde{G}_1$ and other from t to β_2 on $\tilde{G}_2$. Let $\langle F_{s\to\beta_1}, \theta_{s\to\beta_1}\rangle$ be the solution of the first maximum flow problem, i.e., $\langle F_{s\to\beta_1}, \theta_{s\to\beta_1}\rangle =$ *max-flow*$_{s\to\beta_1}(\tilde{G}_1)$. Recall that $F_{s\to\beta_1}$ gives the value of the resulting maximum flow, and $\theta_{s\to\beta_1}$

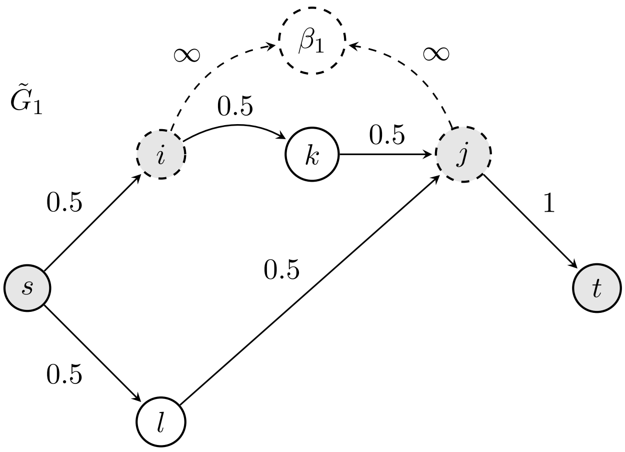


Figure 2.8: Example of an auxiliary graph $\tilde{G}_1$ used in the separation of CCs (2.36). This graph considers the STOP instance of Figure 2.1, the pair of conflicting vertices $\langle i, j \rangle$ and a fractional solution $\bar{x}$, with $\bar{x}_{si} = \bar{x}_{ik} = \bar{x}_{kj} = \bar{x}_{sl} = \bar{x}_{lj} = 0.5$, $\bar{x}_{ki} = \bar{x}_{il} = \bar{x}_{lt} = 0$ and $\bar{x}_{jt} = 1$. Here, the values associated with the arcs are their corresponding capacities, and the infinity symbol stands for a sufficiently large value.

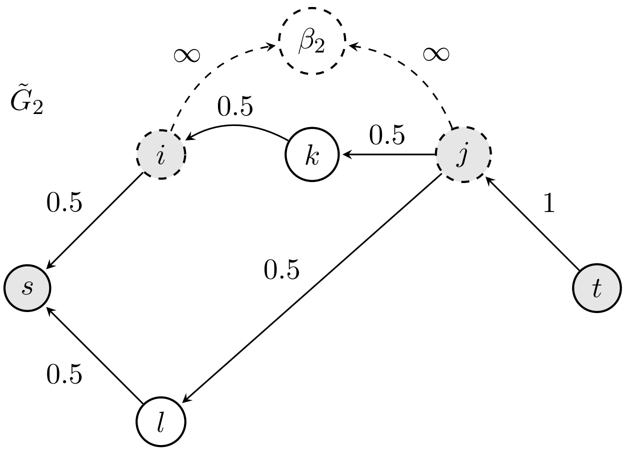


Figure 2.9: Example of an auxiliary graph $\tilde{G}_2$ used in the separation of CCs (2.37). This graph considers the STOP instance of Figure 2.1, the pair of conflicting vertices $\langle i, j \rangle$ and a fractional solution $\bar{x}$, with $\bar{x}_{si} = \bar{x}_{ik} = \bar{x}_{kj} = \bar{x}_{sl} = \bar{x}_{lj} = 0.5$, $\bar{x}_{ki} = \bar{x}_{il} = \bar{x}_{lt} = 0$ and $\bar{x}_{jt} = 1$. Here, the values associated with the arcs are their corresponding capacities, and the infinity symbol stands for a sufficiently large value.

defines a corresponding minimum cut, with $s \in \theta_{s\to\beta_1}$. Then, the algorithm checks if $F_{s\to\beta_1}$ is smaller than $\bar{y}_i + \bar{y}_j$. If that is the case, a violated CC (2.36) is identified and added to $\mathcal{X}$ (see lines 10-14, Figure 2.7). Precisely, this inequality is denoted by $\langle N\backslash\theta_{s\to\beta_1}, \langle i,j\rangle\rangle$ and defined as

$$\sum_{e\in\delta^-(N\backslash\theta_{s\to\beta_1})} x_e \geq y_i + y_j, \tag{2.56}$$

where $N\backslash\theta_{s\to\beta_1}$ corresponds to the subset $V \subseteq N\backslash\{s\}$ of (2.36).

Likewise, let $\langle F_{t\to\beta_2}, \theta_{t\to\beta_2}\rangle$ be the solution of the second maximum flow problem, i.e., $\langle F_{t\to\beta_2}, \theta_{t\to\beta_2}\rangle = \textit{max-flow}_{t\to\beta_2}(\tilde{G}_2)$. If $F_{t\to\beta_2}$ is smaller than $\bar{y}_i + \bar{y}_j$, then a violated CC (2.37) is identified and added to $\mathcal{X}$ (see lines 15-19, Figure 2.7). This CC is denoted by $\langle N\backslash\theta_{t\to\beta_2}, \langle i,j\rangle\rangle$ and defined as

$$\sum_{e\in\delta^+(N\backslash\theta_{t\to\beta_2})} x_e \geq y_i + y_j. \tag{2.57}$$

Here, $N\backslash\theta_{t\to\beta_2}$ corresponds to the subset $V \subseteq N\backslash\{t\}$ of (2.37).

2.3.3 CCCs

First, consider the subset of conflicting vertex pairs $\tilde{\mathcal{K}} \subseteq \mathcal{K}$, which is heuristically computed as described at the beginning of Section 2.3.2. Then, we build the corresponding conflict graph $\tilde{G}_c = (N, \tilde{E}_c)$ representing the pairs of conflicting vertices in $\tilde{\mathcal{K}}$, such that $\tilde{E}_c = \{\{i,j\} : \langle i,j\rangle \in \tilde{\mathcal{K}}\}$. Thereafter, we compute a subset $\tilde{\Sigma} \subseteq \Sigma$ of conflict cliques by finding all the maximal cliques of $\tilde{G}_c$, including the ones of size one. To this end, we apply the depth-first search algorithm of Tomita et al. (2006), which runs with worst-case time complexity of $O(3^{\frac{|N|}{3}})$.

Once $\tilde{\Sigma}$ is computed, we look for violated CCCs of types (2.39) and (2.40) in a similar way as in the separation of CCs described in Section 2.3.2. The algorithm is detailed in Figure 2.10. Let the set $\mathcal{X}'$ keep the CCCs found during the separation procedure. Initially, $\mathcal{X}'$ is empty (line 1, Figure 2.10). Due to the possibly large number of maximal cliques, we adopt a simple filtering mechanism to discard cliques from further search whenever convenient. Precisely, the cliques, which are initially marked as *active* (line 2, Figure 2.10), are disabled if any of its vertices belongs to a previously separated CCC. Then, for every conflict clique $\sigma \in \tilde{\Sigma}$, we check if it is currently *active* (lines 3 and 4, Figure 2.10) and, if so, we build two auxiliary graphs, one for each type of CCC. The first graph, denoted by $\tilde{G}'_1$, is built by adding to the residual graph $\tilde{G}$ an artificial vertex β_1 and $|\sigma|$ arcs (i, β_1), one for each $i \in \sigma$ (see line 6, Figure 2.10). The second one, denoted by $\tilde{G}'_2$, is built by reversing all the arcs of $\tilde{G}$ and, then, adding an artificial vertex β_2, as well as an arc (i, β_2) for each $i \in \sigma$ (see line 7, Figure 2.10).

The capacities of the arcs of $\tilde{G}'_1$ and $\tilde{G}'_2$ are kept in the data structures c'_1 and c'_2, respectively. In both graphs, the capacities of the original arcs in $\tilde{G}$ are preserved (see lines 8-10, Figure 2.10). Moreover, all the additional arcs have a same capacity value, which is equal to a sufficiently large number. Here, we adopted the value of $|M|$, the number of vehicles (see lines 11-13, Figure 2.10).

Once the auxiliary graphs are built for a given $\sigma \in \tilde{\Sigma}$, the algorithm first looks for a violated CCC (2.39) by computing the maximum flow from s to β_1 on $\tilde{G}'_1$. Accordingly, let

Input: A fractional solution $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$, its corresponding residual graph $\tilde{G} = (N, \tilde{A})$ and the subset $\tilde{\Sigma} \subseteq \Sigma$ of maximal conflict cliques.
Output: A set $\mathcal{X}'$ of CCCs violated by $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$.
1. $\mathcal{X}' \leftarrow \emptyset$;
2. Set all cliques in $\tilde{\Sigma}$ as *active*;
3. **for all** $(\sigma \in \tilde{\Sigma})$ **do**
4.   **if** (σ is *active*) **then**
5.   *Step I. Building the auxiliary graphs*
6.   Build $\tilde{G}'_1 = (\tilde{N}'_1, \tilde{A}'_1)$, $\tilde{N}'_1 = N \cup \{\beta_1\}$, $\tilde{A}'_1 = \tilde{A} \cup \{(i, \beta_1) : i \in \sigma\}$;
7.   Build $\tilde{G}'_2 = (\tilde{N}'_2, \tilde{A}'_2)$, $\tilde{N}'_2 = N \cup \{\beta_2\}$, $\tilde{A}'_2 = \{(v, u) : (u, v) \in \tilde{A}\} \cup \{(i, \beta_2) : i \in \sigma\}$;
8.   **for all** $((u, v) \in \tilde{A})$ **do**
9. $c'_1[u, v] \leftarrow c'_2[v, u] \leftarrow c[u, v]$;
10.   **end-for**;
11.   **for all** $(i \in \sigma)$ **do**
12. $c'_1[i, \beta_1] \leftarrow c'_2[i, \beta_2] \leftarrow |M|$;
13.   **end-for**;
14.   *Step II. Looking for a violated CCC (2.39)*
15. $\langle F_{s \to \beta_1}, \theta_{s \to \beta_1} \rangle \leftarrow$ *max-flow*$_{s \to \beta_1}(\tilde{G}'_1)$;
16.   **if** $(F_{s \to \beta_1} < \sum_{i \in \sigma} \bar{y}_i)$ **then**
17. $\mathcal{X}' \leftarrow \mathcal{X}' \cup \{\langle N \backslash \theta_{s \to \beta_1}, \sigma \rangle\}$;
18.   Call *update-active-cliques*($\tilde{\Sigma}$, σ, $\bar{y}$);
19.   **else**
20.   *Step III. Looking for a violated CCC (2.40)*
21. $\langle F_{t \to \beta_2}, \theta_{t \to \beta_2} \rangle \leftarrow$ *max-flow*$_{t \to \beta_2}(\tilde{G}'_2)$;
22.   **if** $(F_{t \to \beta_2} < \sum_{i \in \sigma} \bar{y}_i)$ **then**
23. $\mathcal{X}' \leftarrow \mathcal{X}' \cup \{\langle N \backslash \theta_{t \to \beta_2}, \sigma \rangle\}$;
24.   Call *update-active-cliques*($\tilde{\Sigma}$, σ, $\bar{y}$);
25.   **end-if**;
26.   **end-if-else**
27.   **end-if**;
28. **end-for**;
29. **return** $\mathcal{X}'$;

Figure 2.10: Algorithm used to separate violated CCCs.

$\langle F_{s \to \beta_1}, \theta_{s \to \beta_1} \rangle$ be the solution of *max-flow*$_{s \to \beta_1}(\tilde{G}'_1)$. Recall that $F_{s \to \beta_1}$ gives the value of the resulting maximum flow, and $\theta_{s \to \beta_1}$ defines a corresponding minimum cut, with $s \in \theta_{s \to \beta_1}$. Then, the algorithm checks if $F_{s \to \beta_1}$ is smaller than $\sum_{i \in \sigma} \bar{y}_i$. If that is the case, a violated CCC (2.39) is identified and added to $\mathcal{X}'$. Precisely, this inequality is denoted by $\langle N \backslash \theta_{s \to \beta_1}, \sigma \rangle$ and defined as

$$\sum_{e \in \delta^-(N \backslash \theta_{s \to \beta_1})} x_e \geq \sum_{i \in \sigma} y_i, \tag{2.58}$$

where $N \backslash \theta_{s \to \beta_1}$ corresponds to the subset $V \subseteq N \backslash \{s\}$ of (2.39). If a violated CCC (2.39) is identified, we also disable the cliques that are no longer *active* by calling the procedure *update-active-cliques*, which is described in Figure 2.11. The separation of CCCs (2.39) is summarized at lines 14-19, Figure 2.10.

Notice that, if the separation algorithm does not find a violated CCC (2.39) for the current clique, then this clique remains *active*. In this case, the algorithm looks for a violated CCC of

```
Input: A set of conflict cliques Σ̃, a conflict clique σ ∈ Σ̃ and a fractional solution ȳ.
for all (vertex i ∈ σ) do
    if (ȳ_i > 0) then
        Deactivate every clique in Σ̃ containing i;
    end-if;
end-for;
```

Figure 2.11: Procedure *update-active-cliques*, which manages the currently *active* conflict cliques during the separation of CCCs.

type (2.40) by computing $\langle F_{t\to\beta_2}, \theta_{t\to\beta_2}\rangle = \textit{max-flow}_{t\to\beta_2}(\tilde{G}'_2)$. If $F_{t\to\beta_2}$ is smaller than $\sum_{i\in\sigma} \bar{y}_i$, then a violated CCC (2.40) is identified and added to $\mathcal{X}'$. This CCC is denoted by $\langle N\backslash\theta_{t\to\beta_2}, \sigma\rangle$ and defined as

$$\sum_{e\in\delta^+(N\backslash\theta_{t\to\beta_2})} x_e \geq \sum_{i\in\sigma} y_i. \tag{2.59}$$

Here, $N\backslash\theta_{t\to\beta_2}$ corresponds to the subset $V \subseteq N\backslash\{t\}$ of (2.40). Also in this case, we disable the cliques that are no longer *active* by calling the procedure *update-active-cliques*. The separation of CCCs (2.40) is summarized at lines 20-25, Figure 2.10.

2.3.4 AVICs

Notice that the total number of AVICs (2.48) and (2.49) is equal to $2 \times |E|$, which is at most $|A|$. Then, this family of inequalities can be separated by complete enumeration with $O(|A|)$ time complexity.

2.3.5 LCIs

Here, we describe the separation procedure we adopt to obtain LCIs, which is based on the algorithmic framework of Gu et al. (1998). Since the peculiarities behind lifting might be tricky, we first elucidate in details the concepts of *down-lifting* and *up-lifting*, which we apply throughout this section.

Consider a (not necessarily minimal) cover $C \subseteq P$ for the knapsack constraint (2.51) and the corresponding cover inequality (2.53). Moreover, let $\omega \in \mathbb{Z}^{|P|}$ be the coefficient vector of the y variables in (2.53), such that $\omega_i = 1$ for all $i \in C$, and $\omega_i = 0$ for all $i \in P\backslash C$. Accordingly, (2.53) can be alternatively stated as

$$\sum_{i\in P} \omega_i y_i \leq |C| - 1. \tag{2.60}$$

The attempt to increase the coefficient values of the variables y_i in (2.60) such that $i \in C$ (and, thus, $\omega_i = 1$) is called down-lifting. Likewise, the process of computing new coefficients for the variables y_i such that $i \notin C$ (and, thus, $\omega_i = 0$) is called up-lifting. In both cases, the aim is to strengthen the original cover inequality by replacing the initial coefficients with possibly greater positive integer values. Naturally, these lifted coefficients must be computed in a way that preserves the validity of the cover inequality with respect to the original knapsack constraint (2.51).

In this work, both liftings are done sequentially (i.e., one variable at a time), and the lifted coefficients are computed by solving auxiliary knapsack problems to optimality. In order to describe how down-lifting works, consider a partition of the cover C into two disjoint sets C_1 and C_2, with $C_1 \neq \emptyset$. In this partition, C_1 keeps the indexes of the variables whose coefficients will not be updated (i.e., will remain equal to one), while C_2 identifies the variables to be down-lifted. Moreover, let C' keep the indexes of the variables whose lifted coefficients were already computed, such that, initially, $C' = \emptyset$ and, by the end of all down-liftings, $C' = C_2$. During the down-lifting process, consider the following auxiliary inequality

$$\sum_{i \in C_1} y_i + \sum_{i \in C'} \pi_i y_i \leq |C_1| + \sum_{i \in C'} \pi_i - 1, \tag{2.61}$$

where $\pi_i \in \mathbb{Z}$, $\pi_i \geq 1$ for all $i \in C'$, are the currently available lifted coefficients. Without loss of generality, to down-lift a variable y_j, $j \in C_2 \backslash C'$, the following auxiliary knapsack problem is solved

$$(\mathcal{A}_1) \quad \phi_1 = \max \left\{ \sum_{i \in C_1} y_i + \sum_{i \in C'} \pi_i y_i : \text{(2.51)},\ y_j = 0,\ y_i = 1\ \forall\, i \in C_2 \backslash (C' \cup \{j\}),\ y \in \{0,1\}^{|P|} \right\},$$

which consists of determining the maximum value ϕ_1 that the left-hand side of (2.61) can assume while satisfying the original knapsack constraint (2.51) and fixing $y_j = 0$ and $y_i = 1$ for all $i \in C_2 \backslash (C' \cup \{j\})$. The resulting lifted coefficient of y_j is given by the gap between ϕ_1 and the current right-hand side value of (2.61), i.e., $\pi_j = \phi_1 - (|C_1| + \sum_{i \in C'} \pi_i - 1)$. Naturally, we update $C' \leftarrow C' \cup \{j\}$ after y_j is lifted.

Intuitively speaking, the process described above can be seen as removing j from the cover (by setting $y_j = 0$) and, then, computing the maximum value π_j can assume as to bring j back to the cover. Recall that simply setting $\pi = \mathbf{1}$ and considering $C' = C_2$ ensures the validity of (2.61). Accordingly, keeping the remaining un-lifted variables y_i, for all $i \in C_2 \backslash (C' \cup \{j\})$, fixed at one while down-lifting a variable y_j yields the validity of the LCI obtained after all down-liftings are done (precisely, when $C' = C_2$). In particular, this resulting LCI takes the form

$$\sum_{i \in C_1} y_i + \sum_{i \in C_2} \pi_i y_i \leq |C_1| + \sum_{i \in C_2} \pi_i - 1, \tag{2.62}$$

and is valid for the region Φ determined by the original knapsack constraint (2.51), as defined by (2.52). In order to avoid unnecessary liftings, a word of caution must be given about down-lifting.

Proposition 4. *If a cover C is minimal, then performing down-lifting on any of the variables y_j, $j \in C$, of the original cover inequality (2.60) is ineffective, as it always leads to lifted coefficients equal to one.*

Since we are not sure if this result is already proven in the literature, we devise a formal proof for it in Appendix A.2.

Now, let the set $D \subseteq P \backslash C$ identify the indexes of the y variables to be up-lifted. Moreover, let C'' be the index set of the variables in $P \backslash C$ whose coefficients are already established, such that, initially, $C'' = P \backslash (C \cup D)$ and, by the end of the up-liftings, $C'' = P \backslash C$. Notice that, if $D = P \backslash C$, then C'' is empty before the up-liftings take place. During the whole up-lifting process, consider the following inequality

$$\sum_{i \in C_1} y_i + \sum_{i \in C_2} \pi_i y_i + \sum_{i \in C''} \mu_i y_i \leq |C_1| + \sum_{i \in C_2} \pi_i - 1, \tag{2.63}$$

where $\mu_i \in \mathbb{Z}$, $\mu_i \geq 0$ for all $i \in C''$, are the currently available coefficients. One may note that simply setting $\mu = \mathbf{0}$, i.e., removing all the variables y_i, $i \in P \backslash C$, from (2.63), turns this inequality into (2.62). Moreover, considering $\mu = \mathbf{0}$ and $\pi = \mathbf{1}$ reduces (2.63) to the original cover inequality (2.60).

Without loss of generality, to up-lift a variable y_k, $k \in D \backslash C''$, the following knapsack problem is solved

$$(\mathcal{A}_2) \quad \phi_2 = \max \left\{ \sum_{i \in C_1} y_i + \sum_{i \in C_2} \pi_i y_i + \sum_{i \in C''} \mu_i y_i : (2.51),\, y_k = 1,\, y \in \{0,1\}^{|P|} \right\}.$$

In practice, problem $\mathcal{A}_2$ can be seen as forcefully adding k to the cover (by setting $y_k = 1$) and, then, computing the maximum value ϕ_2 that the left-hand side of the current inequality (2.63) can assume while satisfying the original knapsack constraint (2.51). Notice that, as k does not belong to the cover originally, satisfying (2.51) and $y_k = 1$ can only lead to a value ϕ_2 inferior or equal to the current right-hand side of (2.63). Thus, the resulting lifted coefficient of y_k is given by the gap between the right-hand side value of (2.63) and ϕ_2. Under the current circumstances, we would have $\mu_k = (|C_1| + \sum_{i \in C_2} \pi_i - 1) - \phi_2$. Also notice that, if $p_k > \lfloor \tau \rfloor$, problem $\mathcal{A}_2$ becomes infeasible. In this case, the original coefficient is preserved, i.e., μ_k is set to $\omega_k = 0$. Here, we update $C'' \leftarrow C'' \cup \{k\}$ once y_k is lifted.

In this work, up-lifting is also applied to lift inequalities that are only valid for a restricted region of Φ — see (2.52). In particular, consider an inequality of the form

$$\sum_{i \in C_1} y_i + \sum_{i \in C''} \mu_i y_i \leq |C_1| - 1 \tag{2.64}$$

that is valid for the restricted polyhedron

$$\Phi' = \left\{ y \in \{0,1\}^{|P \backslash C_2|} : \sum_{i \in P \backslash C_2} p_i y_i \leq \lfloor \tau \rfloor - \sum_{i \in C_2} p_i \right\}. \tag{2.65}$$

Notice that Φ' corresponds to $\{y \in \Phi : y_i = 1 \, \forall \, i \in C_2\}$. Then, in this case, to up-lift a variable y_k, $k \in D \backslash C''$, the following knapsack problem is solved

$$\phi_2' = \max \left\{ \sum_{i \in C_1} y_i + \sum_{i \in C''} \mu_i y_i : (2.51),\, y_k = 1,\, y_i = 1 \, \forall i \in C_2,\, y \in \{0,1\}^{|P|} \right\},$$

and the resulting lifted coefficient assumes the value $\mu_k = (|C_1| - 1) - \phi_2'$. Here, setting $y_i = 1$

for all $i \in C_2$ yields the validity of the resulting LCI with respect to the unrestricted region Φ. In fact, in this specific case of up-lifting, the inequality

$$\sum_{i \in C_1} y_i + \sum_{i \in C_2} y_i + \sum_{i \in C''} \mu_i y_i \leq |C_1| + |C_2| - 1 \tag{2.66}$$

is always valid for Φ and corresponds to (2.63) when $\pi = \mathbf{1}$.

For simplicity, we described the process of down-lifting by considering a classical cover inequality of type (2.53). However, this kind of lifting is usually applied only after some or all the variables identified in $P \backslash C$ have already been up-lifted (see again Proposition 4). In this sense, consider the following inequality obtained from up-lifting the variables y_i, for all $i \in D \subseteq P \backslash C$, of either (2.60) or (2.64)

$$\sum_{i \in C_1} y_i + \sum_{i \in C_2} y_i + \sum_{i \in D} \mu_i y_i \leq |C_1| + |C_2| - 1. \tag{2.67}$$

Recall that C_1 and C_2 compose a partition of the cover C, with $C_1 \neq \emptyset$, and $C' \subseteq C_2$ keeps the indexes of the variables whose down-lifted coefficients were already computed. In this case, the auxiliary inequality used during the down-lifting of a variable y_j, $j \in C_2 \backslash C'$, would take the form

$$\sum_{i \in C_1} y_i + \sum_{i \in C'} \pi_i y_i + \sum_{i \in D} \mu_i y_i \leq |C_1| + \sum_{i \in C'} \pi_i - 1, \tag{2.68}$$

while the auxiliary knapsack problem solved would be

$$\phi_1' = \max \left\{ \sum_{i \in C_1} y_i + \sum_{i \in C'} \pi_i y_i + \sum_{i \in D} \mu_i y_i : \text{(2.51)},\ y_j = 0,\ y_i = 1\ \forall\, i \in C_2 \backslash (C' \cup \{j\}),\ y \in \{0,1\}^{|P|} \right\}.$$

Furthermore, the lifted coefficient would be set to $\pi_j = \phi_1' - (|C_1| + \sum_{i \in C'} \pi_i - 1)$.

Taking into account the liftings detailed above, we describe in Figure 2.12 the algorithm used to separate LCIs from a fractional solution $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$. At the first step of the algorithm, we look for a cover C for the corresponding knapsack constraint (2.51). Starting from $C = \emptyset$, we sequentially add to C elements of the subset $\{i \in P : \bar{y}_i > 0\}$ in non-increasing order of the corresponding values in $\bar{y}$. This is done until C covers (2.51), i.e., $\sum_{i \in C} p_i > \lfloor \tau \rfloor$, or until there are no more elements left to be added (see lines 1-8, Figure 2.12). If no cover is found, the algorithm terminates (line 9, Figure 2.12). Otherwise, the cover found is converted into a minimal one by deleting elements from it. This conversion prioritizes the deletion of elements with the smallest relaxation values (see lines 10-15, Figure 2.12), as a way to increase the chances of devising a violated LCI.

In order to guide the lifting process, we define three subsets of indexes, denoted by $Q = \{i \in C : \bar{y}_i = 1\}$, $\bar{C}_1 = \{i \in P \backslash C : \bar{y}_i > 0\}$ and $\bar{C}_2 = \{i \in P \backslash C : \bar{y}_i = 0\}$. Note that $\bar{C}_1 \cup \bar{C}_2 = P \backslash C$ and $\bar{C}_1 \cap \bar{C}_2 = \emptyset$. The following result guarantees that $C \backslash Q \neq \emptyset$ at this point of the separation procedure, i.e., there is at least one element in C whose corresponding value in $\bar{y}$ is fractional.

Proposition 5. *Given a fractional solution $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$ for $\mathcal{L}_2$ and a bound $\tau = \sum_{i \in P} p_i \bar{y}_i$, then any*

Input: A fractional solution $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$ and its corresponding bound $\tau = \sum_{i \in P} p_i \bar{y}_i$.
Output: An LCI of type (2.54), if any.
Step I. Finding an initial cover
Set the initial cover $C \leftarrow \emptyset$ and the iterator $j \leftarrow 1$;
Create a vector H with the indexes in $\{i \in P : \bar{y}_i > 0\}$;
Sort H in non-increasing order of $\bar{y}$;
while $(j \leq |H|)$ **and** $(\sum_{i \in C} p_i \leq \lfloor \tau \rfloor)$ **do**
  $C \leftarrow C \cup \{H[j]\}$;
  $j \leftarrow j + 1$;
end-while;
if $(\sum_{i \in C} p_i \leq \lfloor \tau \rfloor)$ **then** halt with no resulting LCI;
Step II. Converting the cover into a minimal one
Create a vector $\tilde{C}$ with the elements in C;
Sort $\tilde{C}$ in non-decreasing order of $\bar{y}$;
for $(k \leftarrow 1, \ldots, |\tilde{C}|)$ **do**
  if $(\sum_{i \in C \backslash \{\tilde{C}[k]\}} p_i > \lfloor \tau \rfloor)$ **then** $C \leftarrow C \backslash \{\tilde{C}[k]\}$;
end-for;
Step III. Lifting
Define $Q = \{i \in C : \bar{y}_i = 1\}$, $\bar{C}_1 = \{i \in P \backslash C : \bar{y}_i > 0\}$ and $\bar{C}_2 = \{i \in P \backslash C : \bar{y}_i = 0\}$;
Define the initial inequality $\sum_{i \in C \backslash Q} y_i \leq |C \backslash Q| - 1$;
Up-lift y_i, for all $i \in \bar{C}_1$, obtaining $\sum_{i \in C \backslash Q} y_i + \sum_{i \in \bar{C}_1} \mu_i y_i \leq |C \backslash Q| - 1$;
Down-lift y_i, for all $i \in Q$, obtaining $\sum_{i \in C \backslash Q} y_i + \sum_{i \in Q} \pi_i y_i + \sum_{i \in \bar{C}_1} \mu_i y_i \leq |C \backslash Q| + \sum_{i \in Q} \pi_i - 1$;
Up-lift y_i, for all $i \in \bar{C}_2$, obtaining $\sum_{i \in C \backslash Q} y_i + \sum_{i \in Q} \pi_i y_i + \sum_{i \in P \backslash C} \mu_i y_i \leq |C \backslash Q| + \sum_{i \in Q} \pi_i - 1$;
return the resulting LCI;

Figure 2.12: Algorithm used to separate possibly violated LCIs.

cover C for the corresponding knapsack constraint (2.51), with $\bar{y}_i > 0$ for all $i \in C$, necessarily has an element $i \in C$ such that $0 < \bar{y}_i < 1$.

Proof. Consider a cover C for the knapsack constraint (2.51), with $\bar{y}_i > 0$ for all $i \in C$, and suppose, by contradiction, that $\bar{y}_i = 1$ for all $i \in C$. In this case, since $p_i \in \mathbb{Z}^+$ for all $i \in P$ (by definition), and C covers (2.51), we would have

$$\lfloor \tau \rfloor = \lfloor \sum_{i \in P} p_i \bar{y}_i \rfloor \geq \lfloor \sum_{i \in C} p_i \bar{y}_i \rfloor = \sum_{i \in C} p_i > \lfloor \tau \rfloor, \tag{2.69}$$

which is a contradiction. □

Then, starting from the inequality

$$\sum_{i \in C \backslash Q} y_i \leq |C \backslash Q| - 1, \tag{2.70}$$

we up-lift the variables y_i, for all $i \in \bar{C}_1$, which yields the inequality

$$\sum_{i \in C \setminus Q} y_i + \sum_{i \in \bar{C}_1} \mu_i y_i \leq |C \setminus Q| - 1. \tag{2.71}$$

Since (2.70) only considers a restricted region of the original polyhedron Φ, neither (2.70) nor (2.71) are valid LCIs. Nevertheless, the inequality

$$\sum_{i \in C} y_i + \sum_{i \in \bar{C}_1} \mu_i y_i \leq |C| - 1 \tag{2.72}$$

is valid at this point and can be strengthened by down-lifting the variables y_i, for all $i \in Q$. Once these down-liftings are performed, the LCI takes the form

$$\sum_{i \in C \setminus Q} y_i + \sum_{i \in Q} \pi_i y_i + \sum_{i \in \bar{C}_1} \mu_i y_i \leq |C \setminus Q| + \sum_{i \in Q} \pi_i - 1. \tag{2.73}$$

At last, we up-lift the remaining variables y_i, for all $i \in \bar{C}_2$, obtaining

$$\sum_{i \in C \setminus Q} y_i + \sum_{i \in Q} \pi_i y_i + \sum_{i \in P \setminus C} \mu_i y_i \leq |C \setminus Q| + \sum_{i \in Q} \pi_i - 1, \tag{2.74}$$

with $\mu \geq \mathbf{0}$ and $\pi \geq \mathbf{1}$. The sequence of liftings detailed above is summarized at the last step of the algorithm in Figure 2.12 (see lines 16-21). Naturally, the LCI returned by the separation algorithm (line 22, Figure 2.12) is only used in the cutting-plane scheme if it is violated by $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$, i.e., if $\sum_{i \in C \setminus Q} \bar{y}_i + \sum_{i \in Q} \pi_i \bar{y}_i + \sum_{i \in P \setminus C} \mu_i \bar{y}_i > |C \setminus Q| + \sum_{i \in Q} \pi_i - 1$.

Example. To illustrate the execution of the separation algorithm of Figure 2.12 and the liftings described above, suppose that, based on an arbitrary STOP instance, we obtained the following inequation

$$y_1 + 2y_2 + 2y_3 + 3y_4 + y_5 \leq 5, \tag{2.75}$$

which corresponds to a valid inequality of type (2.51). In addition, let $(\bar{x}, \bar{y}, \bar{f}, \bar{\varphi})$, with $\bar{y} = (0.9, 0.2, 0.1, 1, 0.7)$, be the fractional solution given as input to the separation algorithm. Sorting the elements of $\{1, \ldots, 5\}$ in non-increasing order of $\bar{y}$, we obtain $\{4, 1, 5, 2, 3\}$. Then, following the two first steps of the algorithm in Figure 2.12, the initial cover is $\{1, 2, 4, 5\}$, and the minimal cover is obtained from deleting the element 5, which has the second smallest relaxation value ($\bar{y}_5 = 0.7$) among the elements in the initial cover. Notice that we cannot delete 2 in this case, since the resulting set would no longer be a cover. Accordingly, the minimal cover obtained is given by $C = \{1, 2, 4\}$.

Now, we define the sets Q, $\bar{C}_1$ and $\bar{C}_2$ used to guide the liftings, as detailed at the third step of the algorithm in Figure 2.12. In this case, $Q = \{4\}$, $\bar{C}_1 = \{3, 5\}$ and $\bar{C}_2 = \emptyset$. Naturally, $C \setminus Q = \{1, 2\}$. Then, starting from the inequality $y_1 + y_2 \leq 1$, we up-lift the variables whose indexes are in $\bar{C}_1$. Starting from y_3, we solve

$$\max \Big\{ y_1 + y_2 : \text{(2.75)}, \, y_3 = 1, \, y_4 = 1, \, y \in \{0, 1\}^5 \Big\},$$

which corresponds to

$$\max\left\{y_1 + y_2 :\, y_1 + 2y_2 + y_5 \leq 0,\, y \in \{0,1\}^5\right\}.$$

Notice that we could have set $y_5 = 0$, as it does not belong to the cover inequality at this point. In either way, the optimal solution value of this problem is 0, and, thus, the lifted coefficient is $\mu_3 = 1 - 0 = 1$, which yields the inequality $y_1 + y_2 + y_3 \leq 1$. Then, to up-lift y_5, we solve

$$\max\left\{y_1 + y_2 + y_3 :\, (2.75),\, y_5 = 1,\, y_4 = 1,\, y \in \{0,1\}^5\right\},$$

which corresponds to

$$\max\left\{y_1 + y_2 + y_3 :\, y_1 + 2y_2 + 2y_3 \leq 1,\, y \in \{0,1\}^5\right\}$$

and has 1 as optimal value. In this case, we have $\mu_5 = 1-1 = 0$, and the current inequality $y_1 + y_2 + y_3 \leq 1$ remains the same. In order to turn this inequality into a valid LCI for the original unrestricted region Φ, we down-lift the variables whose indexes are in Q, which, in this case, is only y_4. Solving

$$\max\left\{y_1 + y_2 + y_3 :\, (2.75),\, y_4 = 0,\, y \in \{0,1\}^5\right\},$$

which corresponds to

$$\max\left\{y_1 + y_2 + y_3 :\, y_1 + 2y_2 + 2y_3 + y_5 \leq 5,\, y \in \{0,1\}^5\right\},$$

we obtain the optimal value 3. Then, the lifted coefficient assumes $\pi_4 = 3 - 1 = 2$, and the resulting LCI takes the form $y_1 + y_2 + y_3 + 2y_4 \leq 3$. Since $\bar{C}_2 = \emptyset$, no more liftings are done. Notice that, in this case, the resulting LCI is violated by the current fractional solution, as $\bar{y}_1 + \bar{y}_2 + \bar{y}_3 + 2\bar{y}_4 = 0.9 + 0.2 + 0.1 + 2 = 3.2 > 3$.

Chapter 3

Exact algorithms

This chapter is devoted to the exact algorithms developed in this thesis. They consist of the state-of-the-art branch-and-cut algorithm from the literature of the TOP that we adapted to the general case of the STOP and our new cutting-plane algorithms. In this chapter, we also provide some implementation details and a computational study.

3.1 Baseline branch-and-cut algorithm

The exact algorithm of Bianchessi et al. (2018) for the TOP is used as a baseline to evaluate the performance of the cutting-plane scheme here proposed. In the case of the STOP, the compact formulation $\mathcal{F}_1$, defined by (2.1)-(2.16), with the addition of (2.17), is solved by means of an optimization solver (in this case, CPLEX) in a way that GCCs are separated on the fly at each node of the branch-and-bound tree. In order to avoid a tailing-off phenomenon, the separation of cuts at each node of the tree is iterated until the bound improvement is no greater than a pre-established tolerance ϵ_1. In this branch-and-cut algorithm, the same procedure described in Section 2.3.1 is applied to separate GCCs for each fractional solution, and all the violated cuts found are added to the model.

We remark that, although (2.17) is redundant for $\mathcal{F}_1$ (see, again, Corollary 1), this inequality was preserved in our experiments as a way to properly reproduce the original algorithm of Bianchessi et al. (2018). In fact, we conjecture that CPLEX does benefit from this inequality when the separation of built-in cuts is enabled. This is quite intuitive, since (2.17) takes the form of a knapsack constraint, and cover cuts are among the several classical valid inequalities that compose the built-in cuts.

3.2 New cutting-plane scheme

When solving (mixed) integer problems, cutting-plane algorithms work by iteratively reinforcing an initial Linear Programming (LP) model, which usually corresponds to the linearly relaxed version of the original problem. Precisely, at each iteration, the algorithm seeks linear inequalities that are violated by the solution of the current LP model. These inequalities are referred to as *cuts* and are added to the model on the fly until a stopping condition is met or the current solution is feasible (and, thus, optimal) for the original integer problem.

The algorithm here proposed solves the compact formulation $\mathcal{F}_2$ within a cutting-plane scheme that starts from the LP model $\mathcal{L}_2$, the linear relaxation of $\mathcal{F}_2$. Accordingly, the polyhedron defined by $\mathcal{L}_2$ is gradually restricted by the addition of new linear inequalities, which, in this case, correspond to the valid inequalities detailed in Section 2.2. This initial step, called *cutting-plane phase*, was devised in this work according to two configurations: (i) a preliminary one, only considering GCCs, CCs and LCIs, and (ii) an extended one, which separates AVICs and LCIs and replaces GCCs and CCs with CCCs (see, again, Remark 1 and Theorem 2). The existence of these two configurations reflects chronologically the evolution of this research, as GCCs, CCs and LCIs were developed prior to CCCs and AVICs. In the sequel, we describe the algorithm in details for the first configuration (as summarized in Figure 3.1) and highlight the minor modifications incurred by the second configuration (algorithm in Figure 3.2).

Consider the polyhedron Ω defined by the feasible region of $\mathcal{L}_2$. Precisely, $\Omega = \{(x, y, f, \varphi) \in \mathbb{R}^{|A|} \times \mathbb{R}^{|N|} \times \mathbb{R}^{|A|} \times \mathbb{R} :$ (2.2)-(2.8), (2.19)-(2.22), $\mathbf{0} \leq x \leq \mathbf{1}$, $\mathbf{0} \leq y \leq \mathbf{1}$, $f \geq \mathbf{0}$ and $0 \leq \varphi \leq m\}$. For simplicity, assume that $\mathcal{F}_2$ is feasible, and, thus, $\Omega \neq \emptyset$. Let LP^{ψ} and UB^{ψ} keep, respectively, the LP model and the dual (upper) bound on the optimal solution of $\mathcal{F}_2$ available by the end of an iteration ψ of the cutting-plane phase. Likewise, the set Γ^{ψ} keeps all the linear inequalities added to the original LP $\mathcal{L}_2$ until the iteration ψ. In addition, $opt(\mathrm{LP}^{\psi})$ denotes the optimal solution value of a model LP^{ψ}. Here, the iteration $\psi = 0$ stands for the initialization of the cutting-plane phase. Accordingly, the initial model and its corresponding bound are denoted by LP^0 and UB^0, respectively, and $\Gamma^0 = \emptyset$.

Input: The initial LP model $\mathcal{L}_2$ and a tolerance value ϵ_2.
Output: A reinforced LP model and a dual bound on the optimal solution of $\mathcal{F}_2$.
1. Initialize the iterator $\psi \leftarrow 0$;
2. Define $\mathrm{LP}^0 = \mathcal{L}_2$ and set $\Gamma^0 \leftarrow \emptyset$;
3. $\mathrm{UB}^0 \leftarrow opt(\mathrm{LP}^0)$;
4. **do**
5. Update $\psi \leftarrow \psi + 1$;
6. Set $\Gamma^{\psi} \leftarrow \Gamma^{\psi-1}$;
7. Separate and add to Γ^{ψ} *some* of the violated GCCs, if any;
8. Separate and add to Γ^{ψ} *some* of the violated CCs, if any;
9. Separate and add to Γ^{ψ} a violated LCI, if any;
10. Define $\mathrm{LP}^{\psi} = \Big\{ \max \sum_{j \in P} p_j y_j :$ inequality i is satisfied for all $i \in \Gamma^{\psi}$, $(x, y, f, \varphi) \in \Omega \Big\}$;
11. $\mathrm{UB}^{\psi} \leftarrow opt(\mathrm{LP}^{\psi})$;
12. **while** $(\Gamma^{\psi} \neq \Gamma^{\psi-1})$ **and** $(\mathrm{UB}^{\psi-1}$- $\mathrm{UB}^{\psi} > \epsilon_2)$;
13. **return** $(\mathrm{LP}^{\psi}, \mathrm{UB}^{\psi})$;

Figure 3.1: Description of the cutting-plane phase of the algorithm proposed, when considering the first configuration of cuts (GCCs, CCs and LCIs).

After the initialization (lines 1-3, Figure 3.1), the iterative procedure takes place. At each loop of the cutting-plane phase, the iterator ψ is updated, and the set Γ^{ψ} is initialized with the cuts found so far (see lines 5 and 6, Figure 3.1). Then, the algorithm looks for linear inequalities violated by the solution of the current model, which, at this point, corresponds to

$\text{LP}^{\psi-1}$. These cuts are found by means of the separation procedures described in Section 2.3. Instead of selecting all the violated cuts found, we only add to Γ^{ψ} the most violated cut (if any) and the ones that are sufficiently orthogonal to it. As verified in several works (see, e.g., Wesselmann and Suhl (2012); Samer and Urrutia (2015); Bicalho et al. (2016)), this strategy is able to balance the strength and diversity of the cuts separated, while limiting the model size. Here, this strategy is applied to select both GCCs and CCs, but separately (lines 7 and 8, Figure 3.1). Naturally, this filtering procedure does not apply to LCIs, since at most a single LCI is separated per iteration (line 9, Figure 3.1). Details on how these cuts are selected are given in Section 3.3.

After looking for violated inequalities (cuts), we define an updated model LP^{ψ}, which corresponds to adding to $\mathcal{L}_2$ all the cuts selected so far (see line 10, Figure 3.1). Accordingly, the current bound is set to the optimal solution value of LP^{ψ} (line 11, Figure 3.1). The algorithm iterates until either no more violated cuts are found or the bound improvement of the current model is inferior or equal to the tolerance ϵ_2 (see line 12, Figure 3.1). We highlight that the order in which the three types of inequalities are separated is not relevant in this case (lines 7-9, Figure 3.1), as the updated LP model is not solved until all separation procedures are done.

Input: The initial LP model $\mathcal{L}_2$ and a tolerance value ϵ_2.
Output: A reinforced LP model and a dual bound on the optimal solution of $\mathcal{F}_2$.
1. Initialize the iterator $\psi \leftarrow 0$;
2. Set $\Gamma^0 \leftarrow \emptyset$;
3. Separate and add AVICs to Γ^0 by complete enumeration;
4. Define $\text{LP}^0 = \left\{ \max \sum_{j \in P} p_j y_j : \text{inequality } i \text{ is satisfied for all } i \in \Gamma^0, (x, y, f, \varphi) \in \Omega \right\}$;
5. $\text{UB}^0 \leftarrow opt(\text{LP}^0)$;
6. **do**
7. Update $\psi \leftarrow \psi + 1$;
8. Set $\Gamma^{\psi} \leftarrow \Gamma^{\psi-1}$;
9. Separate and add to Γ^{ψ} *some* of the violated CCCs, if any;
10. Separate and add to Γ^{ψ} a violated LCI, if any;
11. Define $\text{LP}^{\psi} = \left\{ \max \sum_{j \in P} p_j y_j : \text{inequality } i \text{ is satisfied for all } i \in \Gamma^{\psi}, (x, y, f, \varphi) \in \Omega \right\}$;
12. $\text{UB}^{\psi} \leftarrow opt(\text{LP}^{\psi})$;
13. **while** $(\Gamma^{\psi} \neq \Gamma^{\psi-1})$ **and** $(\text{UB}^{\psi-1}\text{- UB}^{\psi} > \epsilon_2)$;
14. **return** $(\text{LP}^{\psi}, \text{UB}^{\psi})$;

Figure 3.2: Description of the cutting-plane phase of the algorithm proposed, when considering the second configuration of cuts (AVICs, CCCs and LCIs).

The cutting-plane phase algorithm considering the second configuration of cuts (summarized in Figure 3.2) differs slightly from the algorithm previously described. First, instead of considering $\text{LP}^0 = \mathcal{L}_2$, the algorithm takes as initial model $\mathcal{L}_2$ with the addition of all the AVICs (see lines 2-4, Figure 3.2). Additionally, taking into account Remark 1 and Theorem 2, the separation of GCCs and CCs is replaced by solely separating CCCs (line 9, Figure 3.2).

Once the cutting-plane phase is over, the general cutting-plane algorithm follows the same execution for both configurations of cuts. Precisely, the integrality of the variables x and y is restored, and the resulting reinforced model is solved to optimality by an optimization solver. At this point, inequalities (2.22) from $\mathcal{F}_2$ are turned into cuts (instead of actual restrictions), just like the ones selected in the cutting-plane phase. As discussed by the end of Section 2.1, the aim is to take advantage from cut management mechanisms within the optimization solver's branch-and-cut scheme.

In practical terms, the algorithm described above is a branch-and-cut in which the inequalities proposed in Section 2.2 are only separated at the root node of the branch-and-bound tree. In fact, the cutting-plane proposed can be easily extended by performing the cutting-plane phase of Figure 3.1 (or that of Figure 3.2) at each node of the branch-and-bound tree. Pilot experiments suggested that such approach is not worthy in this case, as the additional strengthening of the model does not pay off the loss in compactness.

3.3 Implementation details

All the codes were developed in C++, along with the optimization solver ILOG CPLEX 12.6. The baseline branch-and-cut algorithm described in Section 3.1 was implemented using the callback mechanism of CPLEX. Moreover, CPLEX was used to solve the LP models within the cutting-plane phase of the cutting-plane algorithm proposed and to close the integrality gap of the reinforced MILP model obtained from the addition of cuts. We kept the default configurations of CPLEX in our implementations, since all the previous works in the literature of TOP that used CPLEX make the same choice.

Regarding the separation of cuts, we solved maximum flow sub-problems with the implementation of the preflow push-relabel algorithm of Goldberg and Tarjan (1988) provided by the open-source Library for Efficient Modeling and Optimization in Networks — LEMON (Dezsõ et al., 2011). The knapsack sub-problems that arise during the separation of LCIs were solved through classical dynamic programming based on Bellman recursion (Bellman, 1957).

In the selection of cuts, we adopted the *absolute violation* criterion to determine which inequalities are violated by a given solution. In turn, the so-called *distance* criterion was used to properly compare two cuts, i.e., to determine which one is most violated by a solution. Given an n-dimensional column vector w of binary variables, a point $\bar{w} \in \mathbb{R}^n$ and an inequality of the general form $a^T w \leq b$, with $a \in \mathbb{R}^n$, $b \in \mathbb{R}$, the absolute violation of this inequality with respect to $\bar{w}$ is simply given by $a^T \bar{w} - b$. Moreover, the distance from $a^T w \leq b$ to $\bar{w}$ corresponds to the Euclidean distance between the hyperplane $a^T w = b$ and $\bar{w}$, which is equal to $\frac{(a^T \bar{w} - b)}{\|a\|}$, where $\|a\|$ is the Euclidean norm of a.

In our implementation of the cutting-plane algorithm, we set two parameters for each type of inequality separated: a precision one used to classify the inequalities into violated or not (namely *absolute violation precision*), and other one to discard cuts that are not sufficiently orthogonal to the most violated ones. The latter parameter determines the minimum angle that an arbitrary cut must form with the most violated cut, as not to be discarded. In practice, this parameter establishes the maximum acceptable inner product between the arbitrary cut and the most violated one. Accordingly, we call it the *maximum inner product*. In the case of two inequalities $a_1^T w \leq b_1$ and $a_2^T w \leq b_2$, with $a_1, a_2 \in \mathbb{R}^n$ and $b_1, b_2 \in \mathbb{R}$, the inner product between them is given by $\frac{(a_1^T a_2)}{\|a_1\| \|a_2\|}$ and corresponds to the cosine of the angle defined by them. The values adopted for these parameters are shown in Table 3.1.

Table 3.1: Parameter configuration adopted in the separation and selection of valid inequalities in the cutting-plane algorithm.

	Parameter	
Inequalities	Absolute violation precision	Maximum inner product
GCCs	0.05	0.03
CCs	0.3	0.03
LCIs	10^{-5}	–

The tolerance input value ϵ_2 of the cutting-plane algorithm (see Figures 3.1 and 3.2) was set to 10^{-3}. In the case of the baseline branch-and-cut, the absolute violation precision regarding the separation of GCCs was also set to 0.05. In addition, the tailing-off tolerance ϵ_1 was set to 10^{-3}, the same value adopted to ϵ_2.

We remark that all the parameter configurations described above were established according to pilot tests on a control set of 10 TOP instances, composed of both challenging instances and some of the smallest ones. This control set is detailed in Table 3.2, where we report, for each instance, the number of vertices ($|N|$), the number of vehicles ($|M|$) and the route duration limit (T). The reduced number of instances was chosen as a way to avoid overfitting. The whole benchmarks of instances adopted in this study are detailed in Section 3.4.

Table 3.2: Control set of TOP instances used to tune the algorithms' parameters.

Instance	$\lvert N\rvert$	$\lvert M\rvert$	T
p3.3.r	33	3	33.3
p4.3.j	100	3	46.7
p4.3.n	100	3	60.0
p5.3.m	66	3	21.7
p5.3.r	66	3	30.0
p6.2.k	64	2	32.5
p6.3.m	64	3	25.0
p6.3.n	64	3	26.7
p7.3.o	102	3	100.0
p7.3.p	102	3	106.7

We also highlight that, despite of their exact-like form, the algorithms adopted to separate GCCs, CCs and CCCs are heuristics. In particular, notice that the simple fact that they only consider the cuts that are violated by at least a constant factor makes them heuristics in practice. Moreover, these algorithms adopt a stopping condition based on bound improvement of subsequent iterations, which might halt the separation before all the violated cuts are found. Then, in practical terms, the separation algorithms adopted do not necessarily give the actual theoretical bounds obtained from the addition of the inequalities proposed.

3.4 Computational experiments

The computational experiments were performed on a 64 bits Intel Core i7-4790K machine with 4.0 GHz and 15.0 GB of RAM, under Linux operating system. The machine has four physical cores, each one running at most two threads in hyper-threading mode. Here, the Baseline Branch-and-Cut and the Cutting-Plane Algorithm are referred to as B-B&C and CPA, respectively. In particular, the implementations of CPA under the first and the second configurations of cuts (as detailed in Section 3.2) are referred to as CPA_1 and CPA_2, respectively. All the exact algorithms were set to run for up to 7200s, the same time limit established in previous works concerning the TOP (Boussier et al., 2007; Poggi et al., 2010; Dang et al., 2013a; Keshtkaran et al., 2016; El-Hajj et al., 2016; Bianchessi et al., 2018).

In our experiments, we used the benchmark of TOP instances introduced by Chao et al. (1996), which consists of complete graphs with up to 102 vertices. In this case, no mandatory vertices are considered. Based on this benchmark, we also generated new instances by randomly setting a percentage of the vertices as mandatory. Here, this percentage was set to only 5%, as greater values led to the generation of too many infeasible instances.

The original benchmark of Chao et al. (1996) is composed of 387 instances, which are sub-divided into seven data sets, according to the number of vertices of their graphs. Within a given data set, the instances only differ by the time limit imposed on the route duration and the number of vehicles, which varies from 2 to 4. The characteristics of these data sets are detailed in Table 3.3. For each set, it is reported the number of instances (#), the number of vertices in the graphs ($|N|$) and the range of values that the route duration limit T assumes.

Table 3.3: Description of the original benchmark of TOP instances.

Set	1	2	3	4	5	6	7
#	54	33	60	60	78	42	60
$\|N\|$	32	21	33	100	66	64	102
T	3.8–22.5	1.2–42.5	3.8–55	3.8–40	1.2–65	5–200	12.5–120

As done in previous works (Dang et al., 2013a; Bianchessi et al., 2018), we pre-processed all the instances used in our experiments by removing vertices and arcs that are inaccessible with respect to the limit T imposed on the total traverse times of the routes. To this end, we considered the R matrix defined by the end of Section 1.2, which keeps, for each pair of vertices, the time duration of a minimum time path between them. Moreover, in the specific case of the cutting-plane algorithm, constraints (2.6) and (2.7) are implicitly satisfied by deleting all the arcs that either enter the origin s or leave the destination t. Naturally, the time spent in these pre-processings are included in the execution times of the algorithms tested.

In Section 3.4.1, we compare the performance of CPA with B-B&C and other exact algorithms in the literature of the TOP at solving the original benchmark of Chao et al. (1996). In turn, the results obtained by our implementations of B-B&C and CPA while solving the new instances (with a non-empty set of mandatory vertices) are discussed in Section 3.4.2.

3.4.1 Results for the TOP instances

Here, we study the behaviour of CPA at solving the TOP benchmark of Chao et al. (1996). In this sense, we first analyzed the impact of the inequalities discussed in Section 2.2 on the strength of the formulation $\mathcal{F}_2$. To this end, we computed the dual (upper) bounds obtained from adding these inequalities to $\mathcal{L}_2$ (the linear relaxation of $\mathcal{F}_2$) according to 10 different configurations, as described in Table 3.4. Precisely, for each instance and configuration, we solved the cutting-plane phase described in Figures 3.1 and 3.2 while considering only the types of inequalities of the corresponding configuration. We remark that these first experiments do not take into account the CPLEX built-in cuts, since only the inequalities described in Section 2.2 are separated at the cutting-plane phase.

Table 3.4: Configurations of valid inequalities.

	Inequalities				
Configuration	GCCs	CCs	CCCs	LCIs	AVICs
1	×				
2		×			
3			×		
4				×	
5					×
6	×	×			
7	×	×		×	
8	×	×		×	×
9			×	×	
10			×	×	×

The results are detailed in Table 3.5. The first column displays the name of each instance set. Then, for each configuration of inequalities, we give the average and the standard deviation (over all the instances in each set) of the percentage bound improvements obtained from the addition of the corresponding inequalities. Without loss of generality, given an instance, its percentage improvement in a configuration $i \in \{1, \dots, 10\}$ is given by $100 \cdot \frac{UB_{LP} - UB_i}{UB_{LP}}$, where UB_{LP} denotes the bound provided by $\mathcal{L}_2$, and UB_i stands for the bound obtained from solving the cutting-plane phase in the configuration i. The last row displays the numerical results while considering the complete benchmark of instances.

The results exposed in Table 3.5 indicate that, on average, CCCs are the inequalities that strengthen formulation $\mathcal{F}_2$ the most, followed by CCs, GCCs, AVICs and LCIs. The results also show that coupling CCCs with LCIs and AVICs gives the best average bound improvement (4.03%) among all the configurations of inequalities tested. We also point out that, although LCIs alone only provide marginal average improvements on the bounds, coupling them with the other inequalities is still effective. Such behaviour is somehow expected, as the separation of LCIs relies on the quality of the currently available bounds. Then, these LCIs tend to work better once the bounds are already strengthened by other inequalities.

One may notice that, for some instance sets, coupling different types of inequalities gives worse average bound improvements than considering only a subset of them (see, e.g., set 3

Table 3.5: Percentage dual (upper) bound improvements obtained from adding to $\mathcal{L}_2$ the inequalities of Section 2.2 according to the 10 configurations in Table 3.4. Results for the original benchmark of TOP instances. Recall that CPA_1 and CPA_2 adopt configurations 7 and 10, respectively.

	Configuration of inequalities									
	1 — GCCs		2 — CCs		3 — CCCs		4 — LCIs		5 — AVICs	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1	3.61	3.09	4.46	3.87	5.33	4.24	0.83	2.03	2.34	1.88
2	0.14	0.44	0.42	1.38	0.87	2.54	0.77	2.38	0.16	0.50
3	1.12	1.08	2.08	1.64	3.77	2.78	0.62	1.01	0.99	0.89
4	4.01	3.53	3.51	3.70	6.38	4.80	0.01	0.02	3.20	2.56
5	0.51	1.65	0.86	1.81	1.68	2.96	0.18	0.65	0.41	1.13
6	0.00	0.00	0.00	0.00	0.01	0.04	0.04	0.10	0.00	0.00
7	3.67	2.27	5.40	4.25	6.45	3.80	0.32	1.10	2.64	1.51
Total	1.98	2.73	2.54	3.43	3.88	4.16	0.37	1.25	1.48	1.91

	Configuration of inequalities									
	6 — GCCs & CCs		7 — 6 & LCIs		8 — 7 & AVICs		9 — CCCs & LCIs		10 — 9 & AVICs	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1	4.77	3.79	5.12	3.89	5.09	4.00	5.21	4.20	5.37	4.29
2	0.44	1.40	1.08	2.88	1.09	2.88	1.16	3.03	1.15	3.03
3	2.12	1.65	3.01	2.16	2.94	1.99	3.65	2.56	3.79	2.66
4	4.90	4.07	4.92	4.06	5.05	4.01	6.38	4.80	6.48	4.76
5	0.89	1.80	1.08	1.91	1.08	1.82	1.84	3.02	1.94	3.09
6	0.00	0.00	0.04	0.10	0.04	0.10	0.07	0.13	0.07	0.13
7	6.00	3.80	6.16	3.72	6.13	3.67	6.61	3.90	6.65	3.89
Total	2.90	3.59	3.21	3.68	3.21	3.67	3.94	4.16	4.03	4.19

under configurations 7 and 8). This behaviour is not inconsistent, since these inequalities are separated heuristically, as discussed by the end of Section 3.3.

In a second experiment, we evaluated the performance of CPA by comparing the results obtained by the algorithm with the ones of B-B&C reported by Bianchessi et al. (2018). To make a fair comparison, we also report the results of our implementation of B-B&C running within our experimental environment. The results are shown in Table 3.6. The first column displays the name of each instance set, and, for each algorithm, we give four result values described as follows. The first value corresponds to the number of instances solved at optimality out of the complete instance set. The second one is the average wall-clock processing time (in seconds) spent in solving these instances. Note that this entry only takes into account the instances solved at optimality. The last couple of result values provides the average and the standard deviation (only over the unsolved instances in each set) of the relative optimality gaps obtained by the algorithm. These gaps are given by $\frac{UB-LB}{UB}$, where LB and UB are, respectively, the best lower and upper bounds obtained by the corresponding algorithm for a given instance. Whenever $LB = UB = 0$, the corresponding optimality gap is set to 0%. The last row gives the overall results considering the complete benchmark of instances.

From the results, one may note that the average optimality gaps (concerning unsolved instances) of the solutions obtained by our implementation of B-B&C are extremely close to those presented in the original report. On the other hand, our implementation of B-B&C solved to optimality significantly less instances than the original report. Particularly, it finds difficulty in closing the gaps of the largest instances (sets 4 and 7). We believe that such behaviour is not only due to the differences in hardware, but also to some specific implementation choices, such as the algorithm adopted to solve the maximum flow problems, the CPLEX solver version and, in special, the values of the parameters discussed in Section 3.3. Since the overall performance of our implementation is in accordance with the original report and the latter does not provide all of the implementation details — in particular, the tolerance and precision values adopted in the separation of GCCs —, we chose not to address this issue in this study.

In any case, the results clearly indicate the superiority of our algorithm (CPA) in solving the original benchmark of TOP instances, even when compared to the original report of B-B&C. In particular, CPA_1 was able to solve to optimality 31 and 14 more instances than B-B&C when considering our implementation and the original report by Bianchessi et al. (2018), respectively. By its turn, CPA_2 was able to solve to optimality 28 and 11 more instances than B-B&C when considering our implementation and the original report by Bianchessi et al. (2018), respectively. We also remark that CPA and B-B&C present comparable average execution times.

Comparing CPA_1 and CPA_2, we notice that the former implementation solves to optimality three more instances than the latter. Additionally, the average optimality gap (concerning all unsolved instances) of the solutions obtained by CPA_1 (2.71%) is almost as tight as that of CPA_2 (2.5%). The results indicate that, although CPA_2 has stronger average dual bounds at the root nodes (see, again, Table 3.5), CPA_1 performs better on average for the instances tested.

As expected, instances whose graphs have greater dimensions are the hardest (sets 4, 5 and 7). We also noticed that instances with greater route duration limits (given by T) tend to

Table 3.6: Comparison between B-B&C and CPA at solving the original benchmark of TOP instances. Bold entries highlight, for each instance set, the best algorithm(s) in terms of number of instances solved to optimality.

	B-B&C (Bianchessi et al., 2018)								CPA (Ours)							
	Original report				Our implementation				CPA_1				CPA_2			
	solved		*unsolved*		*solved*		*unsolved*		*solved*		*unsolved*		*solved*		*unsolved*	
			Gap (%)				Gap (%)				Gap (%)				Gap (%)	
Set	#opt/total	Time (s)	Avg	StDev	#opt/total	Time (s)	Avg	StDev	#opt/total	Time (s)	Avg	StDev	#opt/total	Time (s)	Avg	StDev
1	**54/54**	1.10	–	–	**54/54**	0.70	–	–	**54/54**	1.91	–	–	**54/54**	1.80	–	–
2	**33/33**	0.20	–	–	**33/33**	0.07	–	–	**33/33**	0.13	–	–	**33/33**	0.12	–	–
3	**60/60**	184.90	–	–	**60/60**	109.63	–	–	**60/60**	106.33	–	–	**60/60**	95.80	–	–
4	39/60	870.40	2.29	–	32/60	985.92	2.59	1.69	**43/60**	1286.50	3.03	2.57	42/60	1052.02	2.25	1.78
5	60/78	517.90	3.49	–	60/78	291.58	2.95	1.45	**62/78**	395.45	3.01	1.78	60/78	400.77	3.00	1.82
6	36/42	22.10	1.92	–	39/42	183.95	1.95	0.57	**42/42**	262.58	–	–	41/42	118.76	1.80	–
7	45/60	992.80	2.53	–	32/60	446.84	2.71	1.26	47/60	626.11	1.94	0.75	**48/60**	834.88	2.19	0.68
Total	327/387	424.49	2.67	–	310/387	248.82	2.69	1.45	**341/387**	371.79	2.71	1.95	338/387	352.14	2.50	1.60

be more difficult to solve. On the other hand, the number of vehicles available does not seem to interfere with the difficulty in solving the instances. We believe this is in accordance with the way we model the problem in this work. Precisely, one may note that the size of formulation $\mathcal{F}_2$ (as well as $\mathcal{F}_1$) does not depend on the number of vehicles, as all the routes are implicitly modeled by means of a single commodity.

In Table 3.7, we summarize, for each instance set, the total number of instances solved to optimality by each exact algorithm in the literature, including our CPA. This table is displayed for completeness purposes, as the differences in hardware and experimental environments are not taken into account. The results for the branch-and-price (B&P) and the branch-and-cut-and-price (B&C&P) algorithms of Keshtkaran et al. (2016) are presented separately. Moreover, the algorithm of Poggi et al. (2010) was omitted due to the lack of complete results in the original report.

In total, CPA_1 and CPA_2 solved, respectively, 14 and 11 more instances than any previous exact algorithm in the literature. Together, they were able to prove the optimality of nine TOP instances previously unsolved. Among these instances, six were solved by both algorithms, two were solved exclusively by CPA_1, and one was solved by CPA_2. A detailed per-instance report of the results obtained by our implementations of B-B&C and CPA is presented in Appendix A.3. There, we also point out the new instances solved, as well as their corresponding optimal bounds. In Appendix A.3, we also expose extra information of interest, such as the average optimality gaps at the root nodes of the algorithms proposed, the strength of the initial dual and primal bounds (when compared to the best known ones), the number of nodes explored at the branch-and-bound tree, as well as the average number of cuts separated for each family of inequalities.

3.4.2 Results for the new STOP instances

Now, we study the performance of B-B&C and CPA at solving the new benchmark of STOP instances. The name of each new STOP instance (set) corresponds to the original name of the TOP instance (set) from which it was generated, followed by the percentage of vertices selected as mandatory (in this case, 5%).

As for the TOP instances, we first analyzed the impact of the inequalities discussed in Section 2.2 on the strength of formulation $\mathcal{F}_2$. To this end, we computed the dual (upper) bounds obtained from adding these inequalities to $\mathcal{L}_2$ according to the 10 configurations described in Table 3.4. Precisely, for each instance and configuration, we solved the cutting-plane phase described in Figure 3.1 and 3.2 while considering only the types of inequalities of the corresponding configuration.

The results are detailed in Table 3.8. The first column displays the name of each instance set. Then, for each configuration of inequalities, we give the average and the standard deviation (over all the instances in each set) of the percentage bound improvements obtained from the addition of the corresponding inequalities. Without loss of generality, given an instance, its percentage improvement in a configuration $i \in \{1, \dots, 10\}$ is given by $100 \cdot \frac{UB_{LP} - UB_i}{UB_{LP}}$, where UB_{LP} denotes the bound provided by $\mathcal{L}_2$, and UB_i stands for the bound obtained from solving

Table 3.7: Total number of instances solved by each exact algorithm in the literature of the TOP. Bold entries highlight, for each instance set, the best algorithm(s) in terms of number of instances solved to optimality.

			Keshtkaran et al. (2016)			B-B&C (Bianchessi et al., 2018)		CPA (Ours)	
	Boussier et al. (2007)	Dang et al. (2013a)	B&P	B&C&P	El-Hajj et al. (2016)	Original report	Our implementation	CPA_1	CPA_2
Set	#opt/total	#opt/total	#opt/total	#opt/total	#opt/total	#opt/total	#opt/total	#opt/total	#opt/total
1	51/54	**54/54**	**54/54**	**54/54**	**54/54**	**54/54**	**54/54**	**54/54**	**54/54**
2	**33/33**	**33/33**	**33/33**	**33/33**	**33/33**	**33/33**	**33/33**	**33/33**	**33/33**
3	50/60	**60/60**	**60/60**	51/60	**60/60**	**60/60**	**60/60**	**60/60**	**60/60**
4	25/60	22/60	20/60	22/60	30/60	39/60	30/60	**43/60**	42/60
5	48/78	44/78	60/78	59/78	54/78	60/78	59/78	**62/78**	60/78
6	36/42	**42/42**	36/42	38/42	**42/42**	36/42	37/42	**42/42**	41/42
7	27/60	23/60	38/60	34/60	27/60	45/60	33/60	47/60	**48/60**
Total	270/387	278/387	301/387	291/387	300/387	327/387	306/387	**341/387**	338/347

Table 3.8: Percentage dual (upper) bound improvements obtained from adding to $\mathcal{L}_2$ the inequalities of Section 2.2 according to the 10 configurations in Table 3.4. Results for the new STOP instances. Recall that CPA_1 and CPA_2 adopt configurations 7 and 10, respectively.

	Configuration of inequalities									
	1 — GCCs		2 — CCs		3 — CCCs		4 — LCIs		5 — AVICs	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1_5%	7.32	4.67	8.83	5.98	8.88	4.09	1.00	3.93	5.05	3.41
2_5%	0.32	0.90	0.86	2.44	0.64	1.81	0.51	1.43	0.22	0.61
3_5%	1.48	1.35	2.71	1.90	3.79	2.60	0.59	0.99	1.39	1.14
4_5%	5.94	5.62	5.15	6.46	8.80	7.33	0.00	0.01	5.52	5.01
5_5%	0.18	0.65	0.86	1.46	1.10	2.01	0.02	0.10	0.22	0.43
6_5%	0.06	0.17	0.69	2.24	0.18	0.45	0.03	0.07	0.18	0.53
7_5%	5.96	4.24	8.95	9.58	9.69	4.77	0.00	0.00	7.30	5.97
Total	3.18	4.46	4.03	5.95	5.01	5.59	0.26	1.52	3.00	4.31

	Configuration of inequalities									
	6 — GCCs & CCs		7 — 6 & LCIs		8 — 7 & AVICs		9 — CCCs & LCIs		10 — 9 & AVICs	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1_5%	9.26	6.01	9.46	6.00	9.49	6.05	9.77	6.65	9.13	4.01
2_5%	0.85	2.41	0.97	2.75	1.00	2.82	1.12	3.18	1.21	3.41
3_5%	2.71	1.87	3.22	2.05	3.28	2.04	4.00	2.73	4.07	2.80
4_5%	7.27	6.62	7.27	6.63	7.52	6.58	8.83	7.31	10.00	9.77
5_5%	0.84	1.32	0.87	1.40	0.88	1.39	1.08	1.97	1.25	2.08
6_5%	0.70	2.24	0.73	2.23	0.74	2.24	0.21	0.45	0.28	0.55
7_5%	10.16	8.85	10.08	8.59	10.43	8.60	9.55	4.70	10.63	5.41
Total	4.63	6.14	4.75	6.09	4.86	6.14	5.19	5.97	5.52	6.56

the cutting-plane phase in the configuration i. The last row displays the numerical results while taking into account the complete benchmark of instances.

The results exposed in Table 3.8 follow the same pattern we observed during the resolution of the original TOP instances. Precisely, they indicate that, on average, CCCs are the inequalities that strengthen formulation $\mathcal{F}_2$ the most, followed by CCs, GCCs, AVICs and LCIs. Moreover, coupling CCCs with LCIs and AVICs gives the best average bound improvement (5.52%) among all the configurations of inequalities tested. We also point out that, curiously, the new STOP instances seem to benefit more than the original TOP instances from the addition of the proposed inequalities.

As already discussed in Section 3.4.1, coupling different types of inequalities may lead to worse average bound improvements than considering them separately (see, e.g., set 5_5% under configuration 2 and 6). Also notice that, although CCCs dominate GCCs and CCs (Remark 1 and Theorem 2), there are cases where CCCs give worse average bound improvements than CCs (see set 2_5%, Table 3.8). Both behaviours can be explained by the fact that these inequalities are separated heuristically, as discussed by the end of Section 3.3.

Then, we compared the performance of our implementations of B-B&C and CPA. The results are shown in Table 3.9. The first column displays the name of each instance set, and, for each algorithm, we give four result values described as follows. The first value corresponds to the number of instances solved at optimality (or to proven infeasibility) out of the complete instance set. The second one is the average wall-clock processing time (in seconds) spent in solving these instances. Note that this entry only takes into account the instances solved at optimality. The last couple of result values provides the average and the standard deviation (only over the unsolved instances in each set) of the relative optimality gaps obtained by the algorithm. Recall that these gaps are given by $\frac{UB-LB}{UB}$, where LB and UB are, respectively, the best lower and upper bounds obtained by the corresponding algorithm for a given instance. If, for a given instance, no feasible solution is found within the time limit and its infeasibility is also not proven, the corresponding optimality gap is assumed to be 100%. Likewise, this gap is set to 0% whenever the instance is proven to be infeasible. The last row gives the overall results considering the complete benchmark of instances.

The results indicate that CPA_2 outperforms B-B&C in terms of the quality of the solutions obtained when solving the new benchmark of STOP instances. In particular, the average gaps (concerning the unsolved instances) of the solutions provided by CPA_2 are smaller than or equal to those of B-B&C for all instance sets. Moreover, both implementations of CPA were able solve to proven optimality more instances than B-B&C. In total, CPA_1 and CPA_2 solved to optimality 30 and 31 more instances than B-B&C, respectively. Although B-B&C presents, in general, smaller average execution times, these values are still close enough to the ones obtained by CPA, as they have a same order of magnitude for most of the instance sets.

From the results, we also conclude that CPA_1 and CPA_2 have comparable behaviours when solving the new STOP instances. Nevertheless, we observed some sort of complementarity between them: while CPA_1 solves more instances from sets 5_5% and 6_5%, CPA_2 is more successful in solving the sets 4_5% and 7_5% (the ones with greatest dimension instances).

Notice that, for the instance set with the greatest dimensions (set 7_5%), the standard deviation of the optimality gaps obtained by the three implementations were particularly high. This is partially due to the fact that, for a few instances in this set, the algorithms could neither find feasible solutions nor prove their infeasibility within the time limit, thus implying optimality gaps of 100% in these cases. In fact, by analyzing the results in a per-instance basis, we observed that the three algorithms had difficulty in proving the infeasibility of the new STOP instances when that was the case. Such behaviour is in accordance with the fact that, differently from the TOP, solely finding a feasible solution for an STOP instance (or proving its infeasibility) is NP-hard in the general case, as it will be detailed in Chapter 4 (see Theorem 3 and Corollary 4). From the experiments, we could not conclude whether fixing vertices as mandatory (in the new STOP instances) complicates or favors the solvability of the feasible instances.

In general, the results for the new STOP instances indicate a similar behaviour as the one observed when solving the original TOP instances. Precisely, instances with greater route duration limits tend to be more difficult to be solved by B-B&C and CPA, and the number of vehicles available does not seem to interfere with the difficulty in solving the instances.

A detailed per-instance report of the results obtained by our implementations of B-B&C and CPA is presented in Appendix A.4. There, we also expose extra information of interest, such as the average optimality gaps at the root nodes of the algorithms proposed, the strength of the initial dual and primal bounds (when compared to the best known ones), the number of nodes explored at the branch-and-bound tree, as well as the average number of cuts separated for each family of inequalities.

3.4.3 Summary of the results

In this section, we summarize the main conclusions inferred from the results exposed in Sections 3.4.1 and 3.4.2 and Appendixes A.3 and A.4.

1. From Tables 3.5 and 3.8, we conclude that:

    a) The inequalities proposed in Section 2.2 do not seem to dominate each other, except for CCCs, which were proven to dominate GCCs and CCs (Remark 1 and Theorem 2);

    b) CCs (in CPA_1) and CCCs (in CPA_2) are the inequalities that strengthen formulation $\mathcal{F}_2$ the most;

    c) On average, the new STOP instances seem to benefit more than the original TOP instances from the addition of the proposed inequalities;

2. In total, CPA_1 and CPA_2 solved, respectively, 14 and 11 more instances than any previous exact algorithm in the literature (see Table 3.7). Together, they were able to prove the optimality of nine TOP instances previously unsolved. Regarding the new STOP instances, CPA_1 and CPA_2 solved to optimality 30 and 31 more instances than B-B&C, respectively (see Table 3.9);

3. Overall, CPA_1 and CPA_2 have comparable performances. Although CPA_2 has stronger average dual bounds at the root nodes (see, again, Tables 3.5 and 3.8), the additional strength does not always lead to a faster convergence to optimal solutions. In total, CPA_1

Table 3.9: Comparison between B-B&C and CPA at solving the new benchmark of STOP instances. Bold entries highlight, for each instance set, the best algorithm(s) in terms of number of instances solved to optimality.

	B-B&C (Our implementation)				CPA_1 (ours)				CPA_2 (ours)			
	solved		*unsolved*		*solved*		*unsolved*		*solved*		*unsolved*	
			Gap (%)				Gap (%)				Gap (%)	
Set	#opt/total	Time (s)	Avg	StDev	#opt/total	Time (s)	Avg	StDev	#opt/total	Time (s)	Avg	StDev
1_5%	**54/54**	1.03	–	–	**54/54**	1.62	–	–	**54/54**	2.37	–	–
2_5%	**33/33**	0.03	–	–	**33/33**	0.03	–	–	**33/33**	0.03	–	–
3_5%	**60/60**	133.68	–	–	**60/60**	130.31	–	–	**60/60**	109.11	–	–
4_5%	30/60	639.86	3.15	2.10	41/60	1085.46	3.58	2.85	**43/60**	714.01	3.11	2.53
5_5%	62/78	170.10	2.45	0.94	**65/78**	353.58	2.70	1.10	64/78	297.00	2.25	0.98
6_5%	39/42	174.27	2.20	0.08	**42/42**	219.66	–	–	41/42	132.06	1.68	–
7_5%	38/60	145.89	18.61	33.28	51/60	790.22	13.04	32.63	**52/60**	884.81	14.91	34.40
Total	316/387	158.73	7.75	19.70	346/387	361.04	5.38	15.30	**347/387**	310.69	5.13	15.49

solves three more TOP instances than CPA_2 within two hours of execution, while CPA_2 solves one more STOP instance than CPA_1. They are also comparable in terms of average execution times and average optimality relative gaps of the solutions obtained for both benchmarks of instances (see Tables A.5 and A.9);

4. Regarding the instance characteristics that interfere with the solvability, we noticed that greater route duration limits tend to make instances harder to solve. This is possibly due to the fact that greater limits imply more feasible routes, thus increasing the search space. On the other hand, the number of vehicles available does not seem to interfere with the difficulty in solving the instances, which is in accordance with the way we model the problem in this work;

5. From the experiments, we could not conclude if the fixation of vertices in the STOP makes the problem harder to solve than the TOP. Nevertheless, we observed that the three algorithms implemented had difficulty in proving the infeasibility of the new STOP instances when that was the case, as well as in finding an initial feasible solution. Such behaviour is in accordance with the fact that, differently from the TOP, solely finding a feasible solution for an STOP instance is NP-hard in the general case, as it will be detailed in Chapter 4 (see Theorem 3 and Corollary 4);

6. From Tables A.3, A.4, A.7 and A.8, which expose the average optimality gaps at the root nodes of B-B&C and CPA, we take two main conclusions:

    a) The inequalities proposed in Section 2.2 improve the bounds provided by formulations $\mathcal{F}_1$ and $\mathcal{F}_2$, even when the CPLEX built-in cuts are considered;

    b) On average, the contribution of the primal bounds to the values of the root optimality gaps is greater than that of the dual bounds. In practical terms, the root optimality gaps could be significantly decreased by solely improving the primal bounds, especially for the largest instances — see sets 4(_5%), 5(_5%) and 7(_5%) in the aforementioned tables.

3.5 Warm starting the cutting-plane algorithms with primal heuristics

The results exposed in the previous section indicate that we can obtain tighter root optimality gaps for the cutting-plane algorithms developed by improving the quality of the initial primal bounds. Then, we tested the hybrid algorithm obtained from warm starting CPA with the solutions provided by the heuristic discussed in the next chapter (Chapter 4) as an attempt to improve the overall convergence of the exact algorithms. In total, the hybrid algorithm was able to solve to optimality four more TOP instances and seven more STOP instances than the cutting-plane algorithms alone. Additionally, it provides the optimality certificates of five previously unsolved (even by the cutting-plane algorithms) TOP instances. The results obtained are shown in Appendix A.6.

Chapter 4

Heuristics

This chapter is dedicated to the heuristics we propose for the STOP, which consist of an FP based matheuristic (applied to find initial feasible solutions) and an LNS heuristic. As a theoretical contribution, we prove that finding an initial feasible solution for the STOP is NP-hard. We also discuss some implementation details and present a robust computational study on the performance of the algorithms proposed.

4.1 Finding an initial solution

In the TOP, a trivial feasible solution consists of a set of empty routes, one for each vehicle. This solution can be easily improved by inserting vertices while not exceeding the routes' duration limit. In fact, the heuristics in the literature of the TOP make use of this simple procedure, usually adopting greedy criteria to iteratively add vertices to the empty routes (Chao et al., 1996; Archetti et al., 2007; Vansteenwegen et al., 2009; Souffriau et al., 2010; Lin, 2013; Kim et al., 2013). Once we consider mandatory vertices, a trivial solution with empty routes is no longer feasible. In fact, we can formally prove that finding an initial feasible solution for the STOP is an NP-hard problem. To this end, consider the following definition of the Hamiltonian Path Problem (HPP), which is kwown to be NP-complete (Garey and Johnson, 1979).

Input: A digraph $G = (N, A)$, where N is the vertex set, and A is the arc set. An origin vertex $s \in N$ and a destination vertex $t \in N$. For short, $\langle G, s, t\rangle$.

Question: Is there an Hamiltonian path from s to t in G?

Also consider the problem of determining, given an STOP instance, if there exists a feasible solution for it. This decision problem is referred to as Feasibility STOP (F-STOP) hereafter. We show that HPP is polynomial-time reducible to F-STOP.

Theorem 3. *HPP* $\leq_p$ *F-STOP.*

Proof. First, we propose a straightforward polynomial time reduction of HPP instances to F-STOP ones. Given an HPP instance $\langle G, s, t\rangle$, we build a corresponding F-STOP instance by considering the same digraph G, where s and t are its origin and destination vertices as well. A single vehicle is considered, and all the vertices of G are mandatory (except for s and t).

Accordingly, the set of profitable vertices is empty, and the traverse time vector is set to $d = \mathbf{1}$. The time limit T is a sufficiently large number (say, $T \geq |N| - 1$), as to allow the selection of all mandatory vertices in a route.

Now, we only have to show that an HPP instance has an Hamiltonian path from s to t if, and only if, the corresponding F-STOP instance (built as described above) is feasible. The validity of this proposition is quite intuitive, as (i) the pair of HPP and F-STOP instances shares the same graph, (ii) the time limit T allows that all vertices belong to an STOP solution, and (iii) by definition, the single STOP route must visit each vertex exactly once, which defines an Hamiltonian path. □

Corollary 4. *Finding a feasible solution for an STOP instance (or proving its infeasibility) is NP-hard in the general case.*

Proof. Naturally, finding a feasible solution for STOP is at least as hard as determining if there exists such solution, i.e., solving F-STOP. Then, the result follows directly from Theorem 3 and the NP-completeness of HPP (Garey and Johnson, 1979). □

In this work, we find initial solutions by means of the FP matheuristic framework, which is described next.

4.1.1 Feasibility Pump (FP)

The FP matheuristic was proposed by Fischetti et al. (2005) as an alternative to solve the NP-hard problem of finding feasible solutions for generic MILP problems of the form $\min\{c^T x : Ax \geq b,\ x_i \text{ integer } \forall i \in \mathcal{I}\}$. Here, x and c are column vectors of, respectively, variables and their corresponding costs, A is the restriction matrix, b is a column vector, and $\mathcal{I}$ is the set of integer variables.

At each iteration, also called *pumping cycle* (or, simply, *pump*) of the FP, an integer (infeasible) solution $\tilde{x}$ is used to build an auxiliary LP problem based on the linear relaxation of the original MILP problem. Precisely, the auxiliary problem aims at finding a solution x^* with minimum distance from $\tilde{x}$ in the search space defined by $\{x : Ax \geq b\}$. Each new x^* is rounded and used as the integer solution of the next iteration. The algorithm ideally stops when the current solution of the auxiliary problem is also integer (i.e., $[x^*] = x^*$, where $[x^*] = \lfloor x^* + 0.5 \rfloor$) and, thus, feasible for the original problem. Notice that the FP only works in the continuous space of solutions that satisfy all the linear constraints of the original problem, and the objective function of the auxiliary problems is the one element that guides the fractional solutions into integer feasibility. As accurately summarized by Fischetti et al. (2005), "*the FP generates two (hopefully convergent) trajectories of points x^* and $\tilde{x}$ that satisfy feasibility in a complementary but partial way — one satisfies the linear constraints, the other the integer requirement*".

The original FP framework pays little attention to the quality of the solutions. In fact, the objective function of the original MILP problem is only taken into account to generate an initial fractional solution to be rounded and used in the first iteration. In all the subsequent iterations, the auxiliary LPs aim at minimizing distance functions that do not explore the original objective, which explains the poor quality of the solutions obtained (Fischetti et al., 2005; Achterberg and

Berthold, 2007). Some variations of the framework address this issue by combining, in the auxiliary problems, the original objective function with the distance metric. That is the case, for instance, of the Objective Feasibility Pump (OFP) (Achterberg and Berthold, 2007), in which the transition from the original objective function to the distance-based auxiliary one is done gradually with the progress of the pumps. We refer to Berthold et al. (2019) for a detailed survey on the several FP variations that have been proposed throughout the years to address possible drawbacks and convergence issues of the original framework.

In this work, we adopt both the original FP framework and the OFP to find feasible solutions for $\mathcal{F}_2$. In the sequel, we only describe in details the OFP, as it naturally generalizes the FP. For simplicity, formulation $\mathcal{F}_2$ is used throughout the explanation, instead of a generic MILP.

Consider the vector x of decision variables (one for each arc in A), as defined in Section 2.1, and let $\tilde{x} \in \{0,1\}^{|A|}$ be a binary vector defining a not necessarily feasible solution for $\mathcal{F}_2$. The distance function used to guide the OFP framework into integer feasibility is defined as

$$\Delta(x, \tilde{x}) = \sum_{(i,j) \in A} \mid x_{ij} - \tilde{x}_{ij} \mid, \tag{4.1}$$

which can be rewritten in a linear manner as

$$\Delta(x, \tilde{x}) = \sum_{(i,j) \in A:\, \tilde{x}_{ij}=0} x_{ij} + \sum_{(i,j) \in A:\, \tilde{x}_{ij}=1} (1 - x_{ij}). \tag{4.2}$$

Considering the y decision variables on the selection of vertices in the solution routes (as defined in Section 2.1), the objective function of the auxiliary problems solved at each iteration of the OFP consists of a convex combination of the distance function $\Delta(x, \tilde{x})$ and the original objective function of $\mathcal{F}_2$. Precisely,

$$\Delta_\gamma(x, y, \tilde{x}) = \frac{(1-\gamma)}{\|\Delta(x, \tilde{x})\|}\Delta(x, \tilde{x}) + \frac{\gamma}{\|\sum\limits_{i \in P} p_i y_i\|} \overbrace{(-\sum_{i \in P} p_i y_i)}^{\text{minus (2.18)}}, \tag{4.3}$$

with $\gamma \in [0,1]$, and $\|\cdot\|$ being the Euclidean norm of a vector. Notice that, in (4.3), we consider an alternative definition of $\mathcal{F}_2$ as a minimization problem, in which $\max \sum\limits_{i \in P} p_i y_i = \min(-\sum\limits_{i \in P} p_i y_i)$. Moreover, both $\Delta(x, \tilde{x})$ and the original objective function are normalized in order to avoid scaling issues. Also notice that, since $\tilde{x}$ is a constant binary vector, $\|\Delta(x, \tilde{x})\| = \sqrt{|A|}$ in this case.

Now, consider the polyhedron Ω defined by the feasible region of $\mathcal{L}_2$, the linear relaxation of $\mathcal{F}_2$. Precisely, $\Omega = \{(x, y, f, \varphi) \in \mathbb{R}^{|A|} \times \mathbb{R}^{|N|} \times \mathbb{R}^{|A|} \times \mathbb{R} :$ (2.2)-(2.8), (2.19)-(2.22), $\mathbf{0} \leq x \leq \mathbf{1}$, $\mathbf{0} \leq y \leq \mathbf{1}$, $f \geq \mathbf{0}$ and $0 \leq \varphi \leq m\}$ The OFP works by iteratively solving auxiliary problems defined as

$$D(x, y, \tilde{x}, \gamma) :\ \min\{\Delta_\gamma(x, y, \tilde{x}) :\ (x, y, f, \varphi) \in \Omega\}, \tag{4.4}$$

where γ balances the influence of the distance function and the original objective, i.e., the integer feasibility and the quality of the solution. Considering $\mathcal{F}_2$, the OFP algorithm is described in Figure 4.1.

Input: The model $\mathcal{F}_2$, *max_iter* $\in \mathbb{Z}^+$, *max_iter* ≥ 1, $\lambda \in [0,1]$ and $K \in \mathbb{Z}^+$.
Output: Ideally, a feasible solution for $\mathcal{F}_2$.
1. Initialize $\gamma \leftarrow 1$ and *iter_counter* $\leftarrow 1$;
2. Solve $D(x, y, \mathbf{0}, \gamma)$, obtaining a solution $(x^*, y^*, f^*, \varphi^*)$;
3. **if** x^* is integer **then return** x^*;
4. $\tilde{x} \leftarrow [x^*]$ (= rounding of x^*);
5. **while** (*iter_counter* $\leq$ *max_iter*);
6. Update $\gamma \leftarrow \lambda\gamma$ and *iter_counter* $\leftarrow$ *iter_counter* + 1;
7. Solve $D(x, y, \tilde{x}, \gamma)$, obtaining a solution $(x^*, y^*, f^*, \varphi^*)$;
8. **if** (x^* is integer) **then return** x^*;
9. **if** ($\tilde{x} \neq [x^*]$) **then** $\tilde{x} \leftarrow [x^*]$;
10. **else** flip *rand*$(K/2,\ 3K/2)$ entries $\tilde{x}_{ij}$, $(i,j) \in A$, with highest $\mid x^*_{ij} - \tilde{x}_{ij} \mid$;
11. **end-while**;
12. **return 0**;

Figure 4.1: Description of the OFP algorithm when considering formulation $\mathcal{F}_2$.

Aside from the corresponding model $\mathcal{F}_2$, the algorithm receives as input three values: the maximum number of iterations (pumps) to be performed (*max_iter*), a rate by which the γ value is decreased at each pump (λ) and a basis value (K) used to compute the amplitude of the perturbations to be performed in solutions that cycle.

At the beginning, γ and a variable that keeps the number of the current iteration (*iter_counter*) are both set to one (line 1, Figure 4.1). Then, the current problem $D(x, y, \mathbf{0}, \gamma)$ is solved, obtaining a solution $(x^*, y^*, f^*, \varphi^*)$. Notice that, since $\gamma = 1$ at this point, $D(x, y, \mathbf{0}, \gamma)$ corresponds to $\mathcal{L}_2$, and the integer solution $\mathbf{0}$ plays no role. If x^* is integer, and, thus, the current solution is feasible for $\mathcal{F}_2$, the algorithm stops. Otherwise, the rounded value of x^* is kept in a vector $\tilde{x}$ (see lines 2-4, Figure 4.1).

After the first pump, an iterative procedure takes place until either an integer feasible solution is found or the maximum number of iterations is reached. At each iteration, the γ value is decreased by the fixed rate λ, and the iteration counter is updated. Then, $D(x, y, \tilde{x}, \gamma)$ is solved, obtaining a solution $(x^*, y^*, f^*, \varphi^*)$. If x^* is integer at this point, the algorithm stops. Otherwise, it checks if the algorithm is caught up in a cycle of size one, i.e., if $\tilde{x}$ (the rounded solution from the previous iteration) is equal to $[x^*]$. If not, $\tilde{x}$ is simply updated to $[x^*]$. In turn, if a cycle is detected, the algorithm performs a perturbation on $\tilde{x}$. Precisely, a random integer in the open interval $(K/2,\ 3K/2)$ is selected as the quantity of binary entries $\tilde{x}_{ij}$, $(i,j) \in A$, to be flipped to the opposite bound. This perturbation prioritizes entries that have highest values in the distance vector $\mid x^* - \tilde{x} \mid$. The loop described above is summarized at lines 5-11, Figure 4.1. At last, if no feasible solution is found within *max_pumps* iterations, the algorithm terminates with a null solution (line 12, Figure 4.1).

The original FP framework follows the same algorithm described in Figure 4.1, with the exception that the decrease rate λ given as input is necessarily zero. Then, in the loop of lines 5-11, the $D(x, y, \tilde{x}, \gamma)$ problems are solved under $\gamma = 0$, i.e., without taking into account the original objective function.

Although we defined the OFP under the assumption that the original model is $\mathcal{F}_2$, we also tested the algorithm with a reinforced version of $\mathcal{F}_2$ that considers the inequalities discussed in Section 2.2. The idea is that the stronger model, being closer to the convex hull, might help the OFP converging to an integer solution within less pumps.

4.2 A Large Neighborhood Search (LNS) heuristic with Path Relinking (PR)

In this section, we describe an LNS heuristic for the STOP. The original LNS metaheuristic framework (Shaw, 1998) works by gradually improving an initial solution through a sequence of destroying and repairing procedures. In our heuristic, the LNS framework is coupled with classical local search procedures widely used to improve solutions of routing problems in general. In particular, these procedures, which consist of the classical k-opt (Lin, 1965), vertex shifting and vertex exchanges between routes, are also present in most of the successful heuristics proposed to solve the TOP (e.g., Ke et al. (2008); Vansteenwegen et al. (2009); Souffriau et al. (2010); Kim et al. (2013); Dang et al. (2013b); Ke et al. (2016)). The heuristic we propose also uses a memory component known as Path Relinking (PR). PR was devised by Glover (1997) and its original version explores a neighborhood defined by the set of intermediate solutions — namely, the "path" — between two given solutions. The PR framework has also been successfully applied to solve the TOP (Souffriau et al., 2010).

4.2.1 Main algorithm

We describe in Figure 4.2 the general algorithm of the LNS heuristic we propose. The heuristic receives four inputs: an initial feasible solution — built through the OFP or the FP —, the number of iterations to be performed (*max_iter*), the capacity of the *pool* of solutions (*max_pool_size*) and a parameter called *stalling_limit*, which manages how frequently the PR procedure is called. Precisely, it limits the number of iterations in stalling (i.e., with no improvement in the current best solution) before calling the PR procedure. Initially, the variables that keep the current number of iterations (*iter_counter*) and the number of iterations since the last solution improvement (*stalling_counter*) are set to zero (line 1, Figure 4.2). Then, the initial solution Y is improved through local search procedures (detailed in Section 4.2.4) and added to the initially empty *pool* of solutions Λ (lines 2 and 3, Figure 4.2).

At this point, an iterative procedure is performed *max_iter* times (lines 4-27, Figure 4.2). First, the iteration counter is incremented, and a solution Y' (randomly selected from the *pool*) is partially destroyed by the removal of some vertices, as later described in Section 4.2.2. Then, the algorithm successively tries to improve Y' (lines 8-15, Figure 4.2). To this end, the local searches of Section 4.2.4 are performed on Y'. If the improved Y' is better than the best solution currently in the pool (i.e., its total profit sum is strictly greater), then the stalling counter is set to -1, and a copy of Y' is added to Λ. The addition of solutions to the pool always considers its capacity. Accordingly, if the pool is not full, the new solution is simply added. Otherwise, it takes the place of the current worst solution in the pool (see lines 10-14, Figure 4.2).

```
Input: An initial feasible solution Y, max_iter ∈ ℤ+, max_iter ≥ 1, max_pool_size ∈ ℤ+,
max_pool_size ≥ 1 and stalling_limit ∈ ℤ+, stalling_limit ≥ 1.
Output: An ideally improved feasible solution.
1. Initialize iter_counter ← 0 and stalling_counter ← 0;
2. Improve Y through local searches (see Section 4.2.4);
3. Initialize the pool of solutions Λ ← {Y};
4. while (iter_counter ≤ max_iter);
5.      Update iter_counter ← iter_counter + 1;
6.      Randomly select a solution Y′ from Λ;
7.      Partially destroy Y′ by removing vertices (see Section 4.2.2);
8.      do
9.           Improve Y′ through local searches (Section 4.2.4);
10.          if (Y′ is better than the best solution in Λ) then
11.               stalling_counter ← −1;
12.               if (|Λ| < max_pool_size) then Λ ← Λ ∪ {Y′};
13.               else Replace the worst solution in Λ with Y′;
14.          end-if;
15.     while (perform inter-route shifting perturbations on Y′) (Section 4.2.5);
16.     if (Y′ is better than the best solution in Λ) then
17.          stalling_counter ← 0;
18.     else stalling_counter ← stalling_counter + 1;
19.     if (stalling_counter ≥ stalling_limit) then
20.          Perform the PR procedure considering Λ and Y′ (see Section 4.2.6);
21.          stalling_counter ← 0;
22.     end-if;
23.     if (Y′ ∉ Λ) and (Y′ is better than the worst solution in Λ) then
24.          if (|Λ| < max_pool_size) then Λ ← Λ ∪ {Y′};
25.          else Replace the worst solution in Λ with Y′;
26.     end-if;
27. end-while;
28. return best solution in Λ;
```

Figure 4.2: Description of the general LNS algorithm.

At this point, the algorithm attempts to do vertex shifting perturbations (line 15, Figure 4.2), which are detailed in Section 4.2.5. If it succeeds, the algorithm resumes to another round of local searches. Otherwise, the main loop proceeds by updating the stalling counter. Precisely, if the possibly improved Y' obtained after the successive local searches and shifting perturbations has greater profit sum than the best solution in Λ, then *stalling_counter* is reset to zero. Otherwise, it is incremented by one (lines 16-18, Figure 4.2). After that, the algorithm checks if the limit number of iterations in stalling was reached. If that is the case, the PR procedure, whose description is given in Section 4.2.6, is applied in the current iteration, and the stalling counter is reset once again (lines 19-22, Figure 4.2).

By the end of the main loop, it is checked if Y' should be added to Λ, which only occurs if the solution does not already belong to the pool and its profit sum is greater than the current worst solution available (lines 23-26, Figure 4.2). At last, the algorithm returns the best solution in the pool (line 28, Figure 4.2). In the next sections, we detail all the aforementioned procedures called within the heuristic.

4.2.2 Destroying procedure

The destroying procedure consists of removing some of the profitable vertices belonging to a given feasible solution. Consider a fixed parameter *removal_percentage* $\in [0, 1]$. First, we determine an upper bound on the number of vertices to be removed, namely *max_number_of_removals*. This value is randomly selected in the open interval defined by zero and the product of *removal_percentage* and the quantity of profitable vertices in the solution. Then, the procedure sequentially performs *max_number_of_removals* attempts of vertex removal, such that, at each time, a visited vertex is randomly selected. Nevertheless, a vertex is only actually removed if it is profitable and the resulting route remains feasible.

4.2.3 Insertion procedure

Every time the insertion procedure is called, one of two possible priority orders on the unvisited vertices to the inserted is randomly chosen: non-increasing or non-decreasing orders of profits. Then, according to the selected order, the unvisited vertices are individually tested for insertion in the current solution. If a given vertex can be added to the solution (i.e., its addition does not make the routes infeasible), it is inserted in the route and position that increase the least the sum of the routes' time durations. Otherwise, the vertex remains unvisited and the next one in the sequence is tested for insertion.

4.2.4 Local searches

Given a feasible solution, the local searches here adopted attempt to improve the solution quality, which, in this case, means either increasing the profit sum or decreasing the sum of the routes' times (while maintaining the same profit sum). The general algorithm sequentially performs inter and intra-route improvements, vertex replacements and attempts of vertex insertions, as summarized in Figure 4.3. The inter and intra-route improvements are detailed in Section 4.2.4.1, while the vertex replacements are described in Section 4.2.4.2. The attempts of vertex insertions (lines 3 and 5, Figure 4.3) are done as described in Section 4.2.3. The algorithm stops when it reaches a locally optimal solution with respect to the neighborhoods defined by the aforementioned improvement procedures, i.e., no more improvements are achieved.

Input: An initial feasible solution Y.
Output: A possibly improved version of Y.
1. **do**
2. Do inter and intra-route improvements (Section 4.2.4.1);
3. Try to insert unvisited vertices in Y (Section 4.2.3);
4. Do vertex replacements (Section 4.2.4.2);
5. Try to insert unvisited vertices in Y (Section 4.2.3);
6. **while** (did any improvement on Y);
7. **return** Y;

Figure 4.3: Description of the sequence of local searches.

4.2.4.1 Inter and intra-route improvements

```
Input: An initial feasible solution Y.
Output: A possibly improved version of Y.
1. do
2.     // inter-route improvements
3.     do
4.         for (all combinations of two routes in Y) do
5.             Do 1-1 vertex exchange;
6.             Do 1-0 vertex exchange;
7.             Do 2-1 vertex exchange;
8.         end-for;
9.     while (did any inter-route improvement);
10.    // intra-route improvements
11.    for (each route in Y) do
12.        Do 3-opt improvement;
13.    end-for;
14. while (did any intra-route improvement);
15. return Y;
```

Figure 4.4: Description of the sequence of inter and intra-route local searches.

At each iteration of the algorithm of Figure 4.4, the feasible solution available is first subject to inter-route improvements, i.e., procedures that exchange vertices between different routes (lines 2-9, Figure 4.4). Precisely, for all combinations of two routes, three kinds of vertex exchanges are performed: (i) 1-1, where a vertex from a route is exchanged with a vertex from another route, (ii) 1-0, where a vertex from a route is moved to another one and (iii) 2-1, where two adjacent vertices from a route are exchanged with a vertex from another route. In the three cases, given a pair of routes, an exchange is only allowed if it preserves the solution's feasibility and decreases the total sum of the routes' times. At a call of any of the vertex exchange procedures, the algorithm only performs a single exchange: the first possible by analyzing the routes from beginning to end.

The sequence of inter-route improvements described above is performed until no more exchanges are possible. Then, the algorithm performs the intra-route improvements (lines 10-13, Figure 4.4). Precisely, for each route of the solution, the classical 3-opt operator (Lin, 1965) is applied. If any improvement is achieved through the 3-opt operator, the algorithm resumes the main loop by performing inter-route improvements once again. Otherwise, it returns the current solution and terminates (line 15, Figure 4.4).

Without loss of generality, a k-opt operator works by repeatedly disconnecting a given route in up to k places and, then, testing for improvement all the possible routes obtained from reconnecting the initially damaged route in all the feasible manners. The operator terminates when no more improvements are possible from removing (and repairing) any combination of k or less arcs. At this point, the route is called k-optimal. In Figure 4.5, we give an example of a round of the 3-opt operator on an arbitrary route of a directed graph. Accordingly, the original route of Figure 4.5a is disconnected in three places — identified by dashed arcs —, and reconnected within seven possible manners (Figures 4.5b and 4.5c). The three first ones (Figure 4.5b) are also the moves of the 2-opt operator, as one of the initially removed arcs is

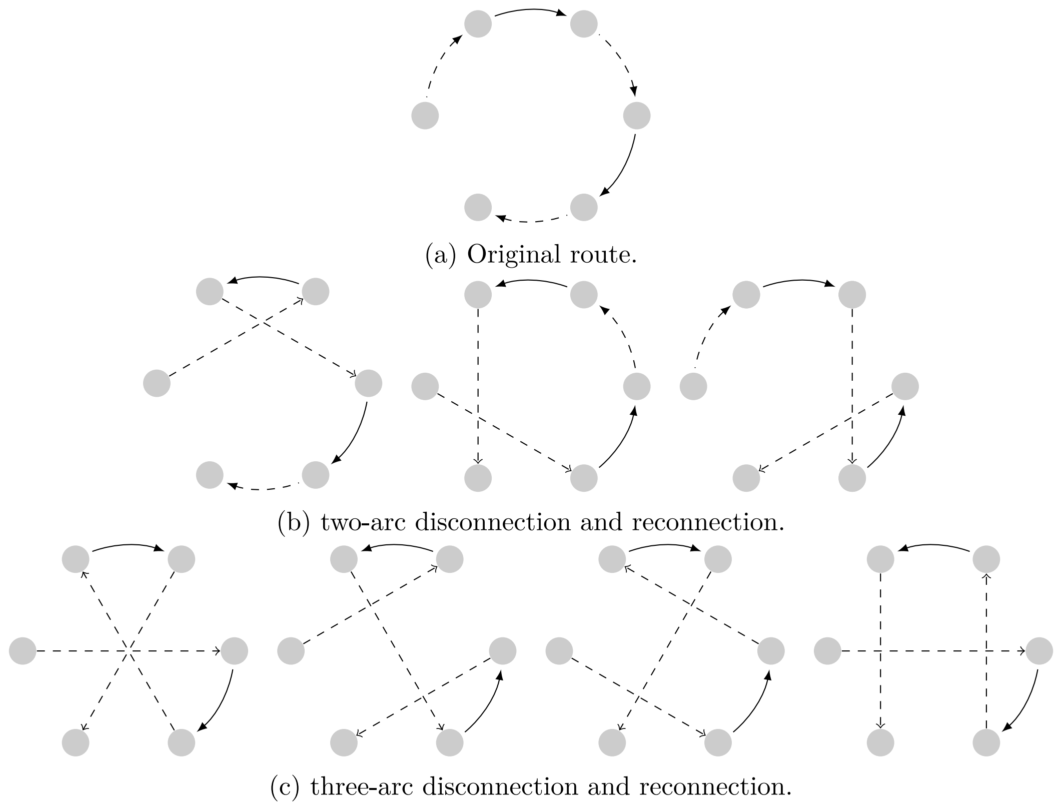


Figure 4.5: Example of a round of the 3-opt operator on an arbitrary route of a directed graph. The dashed arcs in the original route of (a) are the candidates for disconnection. The arc rearrangements of (b) and (c) disconnect two and three of the original arcs, respectively. Notice that, in some cases of reconnection, some arcs of the original route are preserved and others are reversed.

always restored (or reversed). In the latter cases (Figure 4.5c), three of the original arcs are actually discarded. Notice that, depending on the type of reconnection, some arcs of the original route have to be reversed. Thus, in the case of not necessarily complete graphs (such as the STOP), some of the rearrangements might not be always feasible.

4.2.4.2 Vertex replacements

Given a feasible solution, the vertex replacement procedure (Figure 4.6) works by replacing visited vertices with currently unvisited ones. The selection of unvisited vertices to be inserted in the solution is done exactly as in the insertion procedure described in Section 4.2.3, i.e., it follows an either non-decreasing or non-increasing order (chosen at random) of vertex profits. The procedure considers two types of replacements, namely 1-1 and 2-1 *unvisited vertex exchanges*. The former replaces a visited vertex with an unvisited one, while, in the latter, two visited vertices are replaced by an unvisited one. A replacement is only allowed if it preserves the solution's feasibility and either (i) increases the profit sum or (ii) decreases the sum of the routes' times while maintaining the same profit sum. At a call of any of the two types of replacements, the algorithm performs a single replacement: the first feasible one found by analyzing the routes from beginning to end.

Input: An initial feasible solution Y.
Output: A possibly improved version of Y.
1. **do**
2. Do 1-1 unvisited vertex exchange;
3. Do 2-1 unvisited vertex exchange;
4. **while** (did any improvement);
5. **return** Y;

Figure 4.6: Description of the sequence of vertex replacements.

4.2.5 Inter-route shifting perturbation

The inter-route shifting perturbation algorithm works by individually testing if the visited vertices can be moved to any other route, as described in Figure 4.7. Initially, a copy Y' of the initial solution Y is done (line 1, Figure 4.7). Then, Y is only used as a reference, while all the moves are performed in Y', as to avoid multiple moves of a same vertex. For each route, the algorithm attempts to move each of its vertices (either mandatory or profitable) to a different route in that solution, such that the destination route has the least possible increase in its time duration (lines 3-5, Figure 4.7). A vertex can be moved to any position of the destination route, as far as both the origin and the destination routes remain feasible. In this case, moves that increase the total sum of the routes' durations are also allowed.

Input: An initial feasible solution Y.
1. Create a copy Y' of Y;
2. **for** (each route r_1 in Y) **do**
3. **for** (each vertex i in r_1) **do**
4. In Y', try to move i to a route r_2, $r_2 \neq r_1$, with least time increase;
5. **end-for**;
6. **if** (any move from r_1 was done) **then**
7. Try to insert unvisited vertices in r_1 of Y';
8. **end-if**;
9. **end-for**;
10. Replace Y with Y';

Figure 4.7: Description of the inter-route shifting perturbation.

If at least one vertex is relocated during the attempt to move vertices from a given route, the algorithm proceeds by trying to insert in that route currently unvisited vertices (lines 6-8, Figure 4.7). In this case, vertices are inserted according to the procedure described in Section 4.2.3, but only in the route under consideration. After all the originally visited vertices are tested for relocation, Y is replaced by the possibly modified Y' (line 10, Figure 4.7).

4.2.6 The PR procedure

The PR procedure works by exploring neighborhoods connecting an initial solution to the ones of a given set of feasible solutions. Here, this set corresponds to the pool of solutions Λ, which plays the role of a long-term memory. Precisely, at each iteration of the PR procedure, we compute the intermediate solutions — namely, the "path" — between the initial solution and a solution selected from Λ, as detailed in Figure 4.8. The algorithm receives as input an initial solution Y, the current pool of solutions Λ, the capacity of Λ (*max_pool_size*) and a similarity limit $\epsilon_3 \in [0, 1]$ used to determine which pairs of solutions are eligible to be analyzed.

Input: An initial feasible solution Y, the pool of solutions Λ, *max_pool_size* $\in \mathbb{Z}^+$, *max_pool_size* ≥ 1 and a similarity limit ϵ_3.
Output: The possibly updated pool Λ.
1. Set *best_solution* $\leftarrow Y$;
2. **for** (each solution $X \in \Lambda$) **do**
3.   *// Checks similarity between X and Y*
4.   **if** $((2 \times n_{X \cap Y})/(n_X + n_Y) < \epsilon_3)$ **then**
5.   *current_solution* $\leftarrow$ best solution in the "path" from Y to X (see Figure 4.9);
6.   **if** (*current_solution* is better than *best_solution*) **then**
7.   *best_solution* $\leftarrow$ *current_solution*;
8.   **end-if**;
9.   *current_solution* $\leftarrow$ best solution in the "path" from X to Y (see Figure 4.9);
10.   **if** (*current_solution* is better than *best_solution*) **then**
11.   *best_solution* $\leftarrow$ *current_solution*;
12.   **end-if**;
13.   **end-if**;
14. **end-for**;
15. **if** (*best_solution* $\notin \Lambda$) **and** (*best_solution* is better than the worst solution in Λ) **then**
16.   **if** $(|\Lambda| <$ *max_pool_size*$)$ **then** $\Lambda \leftarrow \Lambda \cup \{$*best_solution*$\}$;
17.   **else** Replace the worst solution in Λ with *best_solution*;
18. **end-if**;
19. **return** Λ;

Figure 4.8: Description of the general PR procedure.

Initially, the variable that keeps the best solution found so far in the PR procedure (*best_solution*) is set to the initial solution Y (line 1, Figure 4.8). Then, for each solution $X \in \Lambda$, we compute its similarity with Y, which is given by $(2 \times n_{X \cap Y})/(n_X + n_Y)$, where n_X and n_Y stand for the number of vertices visited in the solutions X and Y, respectively, and $n_{X \cap Y}$ is the number of vertices common to both solutions. If the similarity metric is inferior to the input limit ϵ_3, the procedure computes the "path" from Y to X and the one from X to Y. In both cases, the best solution found in the corresponding "path" (kept in the variable *current_solution*) is compared with the best solution found so far in the whole PR procedure (*best_solution*). Then, if applicable (i.e., if the profit sum of *current_solution* is greater than that of *best_solution*), *best_solution* is updated to *current_solution*. The process of computing the "path" between two given solutions is later detailed in Figure 4.9, and the whole loop described above is summarized at lines 2-14, Figure 4.8. After that, the algorithm attempts to add *best_solution* to Λ (lines 15-18, Figure 4.8). Then, the possibly updated pool is returned and the procedure terminates (line 19, Figure 4.8).

Given two input feasible solutions — a *starting* one Y_s and a *guiding* one Y_g –, the procedure of computing the "path" between them is outlined in Figure 4.9. Initially, the variables that keep the current and the best solutions found so far in the "path" from Y_s and Y_g are both set to Y_s (line 1, Figure 4.9). Then, the set of vertices eligible for insertion (namely *vertices_to_add*) is defined as the vertices that are visited in Y_g and do not belong to Y_s (line 2, Figure 4.9). Notice that all the vertices in *vertices_to_add* are profitable, since Y_s and Y_g are

Input: A starting solution Y_s and a guiding one Y_g.
Output: The possibly improved solution.
1. *best_solution* ← *current_solution* ← Y_s;
2. *vertices_to_add* ← vertices that are visited in Y_g and not in Y_s;
3. Sort *vertices_to_add* in terms of vertex profits (non-increasing or non-decreasing order);
4. **while** (*vertices_to_add* $\neq \emptyset$) **do**
5.   **while** (*vertices_to_add* $\neq \emptyset$) **and** (there is a feasible route in *current_solution*) **do**
6.   Remove a vertex i from *vertices_to_add*, according to the sorting;
7.   Try to insert i in a still feasible route of *current_solution*, allowing infeasibility;
8.   **end-while**;
9.   **for** (each infeasible route j in *current_solution*) **do**
10.   Sequentially remove profitable vertices from j to restore its feasibility;
11.   **end-for**;
12.   Improve *current_solution* through local searches (Section 4.2.4);
13.   **if** (*current_solution* is better than *best_solution*) **then**
14.   *best_solution* ← *current_solution*;
15.   **end-if**;
16. **end-while**;
17. **return** *best_solution*;

Figure 4.9: Algorithm for computing the "path" between two solutions.

both feasible and, thus, visit all the mandatory vertices. After that, *vertices_to_add* is sorted in non-increasing or non-decreasing order of vertex profits (line 3, Figure 4.9). This order is chosen at random at each call of the procedure.

Then, while there are eligible vertices in *vertices_to_add*, the procedure alternates between adding vertices to the current solution — even when it leads to infeasible routes — and removing vertices to restore feasibility. Precisely, at each iteration of the main loop of Figure 4.9 (lines 4-16), the algorithm attempts to add vertices to *current_solution* as follows. First, the next vertex in the order established for *vertices_to_add* is removed from the set. Then, if possible, such vertex is inserted in the route and position that increase the least the sum of the routes' time durations. In this case, an insertion that makes a route infeasible in terms of time limit is also allowed. Nevertheless, once a route becomes infeasible, it is no longer considered for further insertions. The rounds of insertion continue until either *vertices_to_add* gets empty or all the routes of *current_solution* become infeasible (see lines 5-8, Figure 4.9).

At this point, the algorithm removes profitable vertices from *current_solution* to restore its feasibility. In particular, for each infeasible route, profitable vertices are sequentially removed in non-decreasing order of its profits until all the routes become feasible again (lines 9-11, Figure 4.9). If the removal of a vertex would disconnect the route it belongs, then this vertex is preserved. In these cases, the next candidate vertex is considered for removal, and so on.

After the removals, the algorithm attempts to improve the now feasible *current_solution* through the local searches described in Section 4.2.4 (line 12, Figure 4.9), and, if applicable, *best_solution* is updated to *current_solution* (lines 13-15, Figure 4.9). After the main loop terminates, *best_solution* is returned (line 17, Figure 4.9).

4.3 Implementation details

All the codes were developed in C++, and the LP problems that arise in the cutting-plane used to reinforce formulation $\mathcal{F}_2$ were solved by means of the optimization solver ILOG CPLEX 12.6. We kept the default configurations of CPLEX in our implementation.

In the sequel, we describe the parameter configurations adopted in the heuristic algorithms. These configurations were established according to previous studies in the literature, as well as to pilot tests on a control set of 10 STOP instances, composed of both challenging instances and some of the smallest ones. This control set is detailed in Table 4.1, where we report, for each instance, the number of vertices ($|N|$), the number of vehicles ($|M|$) and the route duration limit (T). The reduced number of instances was chosen as a way to avoid overfitting. The complete set of instances adopted in our experiments was previously detailed in Section 3.4.

Table 4.1: Control set of STOP instances used to tune the heuristics' parameters.

Instance	$\vert N\vert$	$\vert M\vert$	T
p3.3.r_5%	33	3	33.3
p4.3.j_5%	100	3	46.7
p4.3.n_5%	100	3	60.0
p5.3.m_5%	66	3	21.7
p5.3.r_5%	66	3	30.0
p6.2.k_5%	64	2	32.5
p6.3.m_5%	64	3	25.0
p6.3.n_5%	64	3	26.7
p7.3.o_5%	102	3	100.0
p7.3.p_5%	102	3	106.7

4.3.1 Parameter configuration adopted for the FP heuristic

In Table 4.2, we summarize the values adopted for the input parameters of the FP framework described in Figure 4.1.

Table 4.2: Parameter configuration adopted for the FP heuristic.

Parameter	*max_pumps*	λ	K
Value	2000	0.9	10

4.3.2 Parameter configuration adopted for the LNS heuristic

In Table 4.3, we summarize the values adopted for the input parameters of the LNS heuristic, including the ones related to the procedures called within the main algorithm described in Figure 4.2.

Table 4.3: Parameter configuration adopted for the LNS heuristic.

Parameter	*max_iter*	*max_pool_size*	*stalling_limit*	*removal_percentage*	ϵ_3
Value	$\{1000, 2000, 5000\}$	20	100	0.75	0.9

4.4 Computational experiments

The computational experiments were performed on a 64 bits Intel Core i7-4790K machine with 4.0 GHz and 15.0 GB of RAM, under Linux operating system. The machine has four physical cores, each one running at most two threads in hyper-threading mode. In our experiments, we tested several variations of the LNS heuristic obtained by considering the FP and the OFP, the original formulation $\mathcal{F}_2$ and its reinforced version, as well as different numbers of iterations for the main LNS algorithm. These variations are later detailed in Sections 4.4.2 and 4.4.3.

We used the same benchmarks of instances adopted in the experiments regarding the exact algorithms (see Section 3.4). The experiments detailed in this section focus on the instances in which the set of mandatory vertices is non-empty, as this specific structure is the factor that brings the extra level of difficulty and justifies using the FP heuristic to find initial solutions. Nevertheless, for completeness, we also report the results obtained for the original TOP instances, where no mandatory vertex exists.

Here, all the instances were also pre-processed by removing vertices and arcs that are inaccessible with respect to the limit T imposed on the total traverse times of the routes. Also in this case, the time spent in the pre-processing is included in the execution times of the algorithms tested.

4.4.1 Statistical analysis adopted

Since all the heuristic algorithms proposed have randomized choices within their execution, we ran each algorithm 10 times for each instance to properly assess their performance. In this sense, we considered a unique set of 10 seeds common to all algorithms and instances tested. To evaluate the quality of the solutions obtained by the heuristics proposed, we compared them with the primal solutions/bounds provided by the cutting-plane algorithm of CPA_1, as described in Section 3.4.

To assess the statistical significance of the results, we follow the tests suggested by Demšar (2006) for the simultaneous comparison of multiple algorithms on different data (instance) sets. Precisely, we first apply the Iman-Davenport test (Iman and Davenport, 1980) to check the so-called *null hypothesis*, i.e., the occurrence of equal performances with respect to a given indicator (e.g., quality of the solution's bounds). If the null hypothesis is rejected, and, thus, the algorithms' performances differ in a statistically significant way, a post-hoc test is performed to analyze these differences more closely. In our study, we adopt the post-hoc test proposed by Nemenyi (1963).

The Iman-Davenport test is a more accurate version of the non-parametric Friedman test (Friedman, 1937). In both tests, the algorithms considered are ranked according to an indicator of performance. Let I be the set of instances and J be the set of algorithms considered.

In our case, the algorithms are ranked for each instance separately, such that r_i^j stands for the rank of an algorithm $j \in J$ while solving an instance $i \in I$. Accordingly, the value of r_i^j lies within the interval $[1, |J|]$, such that better performances are linked to smaller ranks (in case of ties, average ranks are assigned). The Friedman test compares the average ranks (over all instances) of the algorithms, which are given by $R_j = \frac{1}{|I|} \sum_{i \in I} r_i^j$ for all $j \in J$. Then, the Friedman statistic is defined as

$$\chi_F^2 = \frac{12|I|}{|J|(|J|-1)} \left[\sum_{j \in J} R_j^2 - \frac{|J|(|J|+1)^2}{4} \right] \tag{4.5}$$

and is distributed according to the chi-squared (χ^2) distribution with $|J|-1$ degrees of freedom when $|I|$ and $|J|$ are large enough (as a rule of thumb, $|I| > 10$ and $|J| > 5$). For smaller numbers of algorithms and instances, exact critical values have been computed (Sheskin, 2007). By its turn, the less conservative statistic used in the Iman-Davenport test is given by

$$F_F = \frac{(|I|-1)\chi_F^2}{|I|(|J|-1) - \chi_F^2} \tag{4.6}$$

and follows the F distribution with $|J|-1$ and $(|J|-1)(|I|-1)$ degrees of freedom. The corresponding critical values can also be found in the book of Sheskin (2007).

Critical values are determined considering a significance level α, which, in this case, indicates the probability of the null hypothesis being erroneously rejected. In practical terms, the smaller α, the greater the statistical confidence of the test. Accordingly, in the case of the Iman-Davenport test, the null hypothesis is rejected if the statistic F_F is greater than the critical value. In our experiments, we alternatively test the null hypothesis by determining (through a statistical computing software) the so-called p-value, which provides the smallest level of significance at which the null hypothesis would be rejected. In other words, given an appropriate significance level α (usually, at most 5%), we can safely discard the null hypothesis if p-value $\leq \alpha$.

Once the null hypothesis is rejected, we can apply the post-hoc test of Nemenyi (1963), which compares the algorithms in a pairwise manner. The performances of two algorithms $j, k \in J$ are significantly different if the corresponding average ranks R_j and R_k differ by at least the Critical Difference

$$CD = q_\alpha \sqrt{\frac{|J|(|J|+1)}{6|I|}}, \tag{4.7}$$

where the critical value q_α is based on the Studentized range statistic. In our experiments, we used the R open software for statistical computing[1] to compute all of the statistics needed, including the average ranks and the critical differences.

[1] https://www.r-project.org/

4.4.2 Results for the FP algorithms

We first compared four variations of the FP framework discussed in Section 4.1.1, as summarized in Table 4.4. Precisely, we considered both the FP and the OFP while based on the original formulation $\mathcal{F}_2$ and the one reinforced according to the cutting-plane algorithm described in Section 3.2. The latter formulation is referred to as $\mathcal{F}_2$+cuts.

Table 4.4: Variations of the FP analyzed in our study. $\mathcal{F}_2$+cuts stands for the reinforced version of the formulation $\mathcal{F}_2$ discussed in Section 3.2.

	Framework		Base formulation	
Algorithm	FP	OFP	$\mathcal{F}_2$	$\mathcal{F}_2$+cuts
FP_raw	×		×	
FP_cuts	×			×
OFP_raw		×	×	
OFP_cuts		×		×

In Table 4.5, we report the results obtained by the four FP algorithms described in Table 4.4. For each algorithm and instance set, we report four values: (i) the average and (ii) the standard deviation of the relative gaps given by $100 \cdot \frac{LB^* - LB}{LB^*}$, where LB^* is the best primal bound (for the instance) obtained by CPA_1, and LB is the average bound (over the 10 executions) obtained by the corresponding FP algorithm; (iii) the average number of iterations/pumps (over the 10 executions and all instances of the set), and (iv) the average wall-clock processing time (in seconds). We highlight that, for the cases where a feasible solution is not found in any of the 10 executions, the corresponding relative gaps of (i) are set to 100%. The same is done in case no primal bound is available from the baseline exact algorithm. Moreover, when the instance is proven to be infeasible, the gap is set to 0%. Entries in bold discriminate the cases where the reinforcement of the original formulation led to better performances or not.

First, notice that, in terms of solution quality, the results of the four algorithms are rather poor, as the average relative gaps and standard deviations are quite high. Nevertheless, these results do not indicate unsuccessful performances, as the general FP framework was devised to find feasible solutions, usually at the expense of solution quality. As expected, the quality of the solutions obtained by the algorithms that use the OFP framework is slightly superior (on average) to that of solutions obtained by the algorithms based on pure FP. Nevertheless, as we will discuss later, we cannot assure that the gain of the OFP is statistically significant in this case.

More importantly, the results suggest that, on average, the reinforcement of the original formulation $\mathcal{F}_2$ improves both the quality of the solutions obtained and the convergence (signaled by the number of pumps) of the algorithms to feasible solutions. In particular, for all instance sets but 6_5%, the average gaps of the solutions obtained by FP_cuts are strictly smaller (and, thus, better) than those obtained by FP_raw. The same applies for the average numbers of pumps. When comparing OFP_cuts and OFP_raw, the same pattern is verified. Regarding average execution times, the four algorithms are comparable.

We applied the Iman-Davenport and the Nemenyi tests to validate the statistical sig-

Table 4.5: Summary of the results obtained by the four FP algorithms described in Table 4.4. Bold entries highlight, for each instance set, the best algorithm(s) in terms of average gaps and number of pumps.

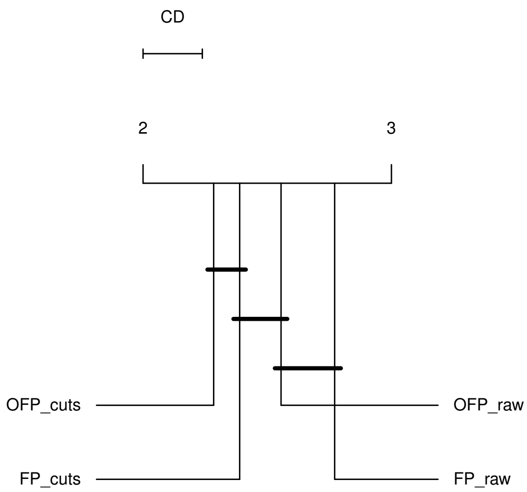
	FP_raw				FP_cuts			
	Gap (%)				Gap (%)			
Set	Avg	StDev	Pumps (#)	Time (s)	Avg	StDev	Pumps (#)	Time (s)
1_5%	26.60	29.92	5.86	0.06	**26.19**	29.58	**4.59**	0.10
2_5%	11.74	23.61	2.19	0.00	**9.18**	20.82	**1.75**	0.00
3_5%	29.47	26.30	7.32	0.08	**28.24**	24.33	**6.12**	0.14
4_5%	29.69	24.62	86.59	39.38	**18.49**	19.70	**53.41**	56.76
5_5%	27.76	26.33	38.06	3.33	**24.25**	25.67	**34.91**	10.96
6_5%	**12.46**	16.87	5.23	0.45	13.66	18.07	**5.05**	0.70
7_5%	27.66	34.21	161.26	109.12	**16.42**	24.44	**53.40**	83.76
Total	25.12	27.51	48.80	23.77	**20.60**	24.48	**25.88**	24.11

	OFP_raw				OFP_cuts			
	Gap (%)				Gap (%)			
Set	Avg	StDev	Pumps (#)	Time (s)	Avg	StDev	Pumps (#)	Time (s)
1_5%	25.46	29.64	19.74	0.30	**23.80**	26.71	**18.09**	0.44
2_5%	11.54	23.70	9.14	0.01	**11.18**	22.00	**8.07**	0.02
3_5%	27.30	23.63	25.65	0.38	**26.81**	23.41	**24.03**	0.54
4_5%	28.46	24.36	102.70	61.43	**20.21**	20.43	**72.20**	80.50
5_5%	24.48	25.10	65.49	7.46	**21.77**	23.48	**53.63**	16.86
6_5%	13.50	21.02	66.72	7.77	**12.49**	16.22	**18.89**	3.09
7_5%	25.20	31.66	187.30	133.95	**15.24**	23.20	**90.91**	175.60
Total	23.49	26.45	72.91	32.74	**19.67**	23.06	**45.08**	43.59

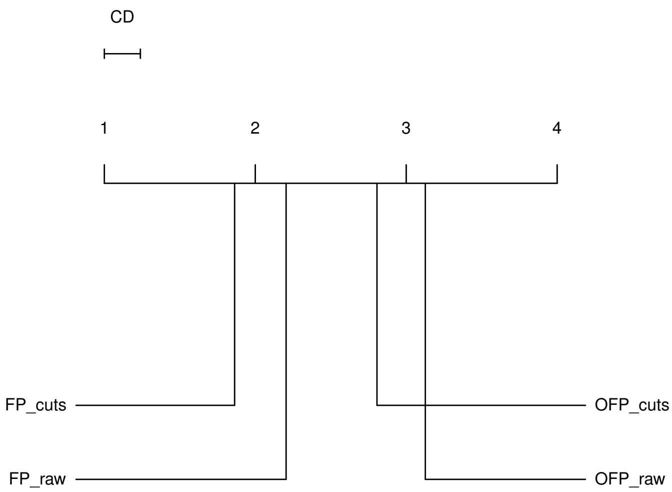
nificance of the results discussed above. While comparing the relative gaps, we ranked the algorithms according to the values $100 \cdot \left(1 - \frac{LB^* - LB}{LB^*}\right)$, as to make smaller gaps imply greater ranks. Similarly, while comparing the number of pumps, the algorithms were ranked based on the maximum number of pumps (parameter *max_iter* of Figure 4.1) minus the actual number of pumps performed. With respect to the relative gaps, we obtained the statistic $F_F = 10.76$, with p-value $= 5.62 \cdot 10^{-7}$. Regarding the number of pumps, we obtained the statistic $F_F = 94.06$, with p-value $= 2.2 \cdot 10^{-16}$. Then, in both cases, we can safely reject the null hypothesis of equal performances.

To compare the performance of the algorithms in a pairwise manner, we proceeded with the Nemenyi test. Figure 4.10 depicts the average ranks of the four FP algorithms and the Critical Difference (CD) while considering a significance level $\alpha = 5\%$. Connections between algorithms indicate non-significant differences, i.e., the difference between the corresponding pair of average ranks is no greater than the CD. Figures 4.10a and 4.10b are based on the relative gaps and the number of pumps, respectively.

With respect to the relative gaps, we can conclude that the performance of OFP_cuts is significantly better than those of OFP_raw and FP_raw. Moreover, FP_cuts outperforms FP_raw. Nevertheless, we cannot conclude if OFP_cuts is significantly better than FP_cuts, neither that OFP_raw is better than FP_raw in this same indicator. Regarding the convergence (number of pumps), FP_cuts outperforms all the other algorithms. In summary, from the results, we can safely conclude that the reinforcement of $\mathcal{F}_2$ here applied yields better performances both in terms of convergence and solution quality. In addition, considering both indicators, FP_cuts stands out as the best option.



(a) Ranks based on the relative gaps.



(b) Ranks based on the number of pumps.

Figure 4.10: FP algorithms' average ranks depicted on number lines, along with the Critical Difference (CD), when considering a significance level $\alpha = 5\%$. Connections between algorithms indicate non-significant differences.

4.4.3 Results for the LNS heuristic

In the experiments concerning the LNS heuristic, we tested the influence of the initial solution on the overall performance of the heuristic, as well as the impact of the number of iterations on the quality of the final solution obtained. To this end, we tested six variations of the heuristic by considering the two best FP algorithms — FP_cuts and OFP_cuts — coupled with the LNS framework of Section 4.2 running for $max\_iter \in \{1000, 2000, 5000\}$ iterations. Without loss of generality, an algorithm that receives an initial solution provided by FP_cuts and runs the LNS heuristic for 1000 iterations is referred to as FP_cuts_LNS_1000.

In Table 4.6, we summarize the results obtained by the six LNS algorithms tested. For each algorithm and instance set, we report five values: (i) the average and (ii) the standard deviation of the relative gaps given by $100 \cdot \frac{LB^* - LB}{LB^*}$, where LB is the average bound (over the 10 executions) obtained by the corresponding algorithm. Recall that LB^* is the best primal bound (for the instance) obtained by CPA_1; (iii) the average (over the 10 executions and all instances of the set) profit sum of the solutions obtained, and (iv) the average (over all instances of the set) profit sum of the best (in each round of 10 executions) solutions found. At last, we provide (v) the average wall-clock processing time (in seconds) of the execution of the LNS steps, excluding the time spent by the FP algorithm in finding the initial solution. We highlight that, for the cases where a feasible solution is not provided in any of the 10 executions of the corresponding FP algorithm, the due relative gaps of (i) are set to 100%. The same is done in case no primal bound is available from the baseline exact algorithm. Moreover, negative gaps indicate that the heuristic gives better primal bounds than the baseline ones. Entries in bold discriminate the cases where the OFP framework led to better performances than the FP or otherwise.

The results indicate that, on average, the LNS algorithms that use the initial solutions provided by FP_cuts and OFP_cuts reach final solutions with comparable quality. In addition, as expected, increasing the number of iterations does improve (on average) the quality of the solutions obtained. Also notice that the average execution times are almost the same for algorithms with a same number of iterations, which is in accordance with the stopping criterion adopted in the algorithms. In fact, the LNS heuristics run fairly fast, being the search for initial feasible solutions responsible for the majority of the computational effort (see again Table 4.5). Such behaviour was expected, since finding an initial solution for the STOP is an NP-hard task (Corollary 4).

As for the FP heuristics, we also applied the Iman-Davenport and the Nemenyi tests to validate the statistical significance of the results discussed above. We highlight that, although better (greater) profit sums imply smaller relative gaps on a per-instance basis, the same does not hold when we consider average values, since relative gaps are normalized by definition. Once our statistical tests use the results on a per-instance basis, considering either relative gaps or profit sums leads to a same ranking of the algorithms. In particular, we ranked the algorithms according to the values $100 \cdot \left(1 - \frac{LB^* - LB}{LB^*}\right)$, as to make smaller gaps imply greater ranks.

Regarding the Iman-Davenport test, we obtained the statistic $F_F = 7.26$, with p-value $= 9.49 \cdot 10^{-7}$, thus rejecting the null hypothesis of equal performances of all the six algorithms tested. Then, proceeding with the Nemenyi test, Figure 4.11 depicts the average ranks of the six

Table 4.6: Summary of the results obtained by six variations of the LNS algorithm tested. The execution times do not consider the time spent by the FP algorithms in finding initial solutions. Bold entries highlight, for each instance set, the best algorithm(s) in terms of average gaps and profit sums.

	FP_cuts_LNS_1000					OFP_cuts_LNS_1000				
	Gap (%)		Profit sum			Gap (%)		Profit sum		
Set	Avg	StDev	Avg	Best	Time (s)	Avg	StDev	Avg	Best	Time (s)
1_5%	0.37	1.91	79.34	**79.63**	0.10	**0.00**	0.00	**79.63**	**79.63**	0.10
2_5%	**0.00**	0.00	**41.52**	**41.52**	0.01	**0.00**	0.00	**41.52**	**41.52**	0.01
3_5%	**0.00**	0.00	**308.50**	**308.50**	0.20	**0.00**	0.00	**308.50**	**308.50**	0.20
4_5%	−0.07	1.22	609.37	613.75	7.37	**-0.19**	1.08	**612.15**	**614.23**	7.10
5_5%	**1.34**	11.32	602.36	603.14	1.04	**1.34**	11.32	**602.40**	**603.21**	1.04
6_5%	**0.01**	0.03	**375.20**	**375.29**	0.55	0.02	0.05	375.11	**375.29**	0.55
7_5%	**1.79**	12.90	**306.32**	**311.22**	1.42	3.48	18.08	292.82	295.20	1.36
Total	**0.59**	7.23	**366.53**	**368.18**	1.68	0.78	8.79	364.91	365.78	1.63

	FP_cuts_LNS_2000					OFP_cuts_LNS_2000				
	Gap (%)		Profit sum			Gap (%)		Profit sum		
Set	Avg	StDev	Avg	Best	Time (s)	Avg	StDev	Avg	Best	Time (s)
1_5%	0.37	1.91	79.34	**79.63**	0.20	**0.00**	0.00	**79.63**	**79.63**	0.20
2_5%	**0.00**	0.00	**41.52**	**41.52**	0.03	**0.00**	0.00	**41.52**	**41.52**	0.03
3_5%	**0.00**	0.00	**308.50**	**308.50**	0.39	**0.00**	0.00	**308.50**	**308.50**	0.39
4_5%	−0.15	1.19	610.07	614.02	14.75	**-0.25**	1.11	**612.78**	**614.30**	14.35
5_5%	1.31	11.32	602.70	603.14	2.08	**1.30**	11.32	**602.74**	**603.21**	2.09
6_5%	**0.00**	0.02	**375.24**	**375.29**	1.09	0.01	0.03	375.21	**375.29**	1.11
7_5%	**1.74**	12.90	**306.73**	**311.28**	3.04	3.44	18.08	293.13	295.25	2.85
Total	**0.56**	7.23	**366.78**	**368.23**	3.39	0.76	8.80	365.14	365.80	3.30

	FP_cuts_LNS_5000					OFP_cuts_LNS_5000				
	Gap (%)		Profit sum			Gap (%)		Profit sum		
Set	Avg	StDev	Avg	Best	Time (s)	Avg	StDev	Avg	Best	Time (s)
1_5%	0.37	1.91	79.34	**79.63**	0.50	**0.00**	0.00	**79.63**	**79.63**	0.51
2_5%	**0.00**	0.00	**41.52**	**41.52**	0.07	**0.00**	0.00	**41.52**	**41.52**	0.07
3_5%	**0.00**	0.00	**308.50**	**308.50**	0.98	**0.00**	0.00	**308.50**	**308.50**	1.00
4_5%	−0.24	1.17	611.07	**614.47**	37.46	**-0.30**	1.14	**613.28**	614.43	35.98
5_5%	**1.29**	11.32	**602.96**	**603.21**	5.30	**1.29**	11.32	602.94	**603.21**	5.30
6_5%	**0.00**	0.01	**375.26**	**375.29**	2.74	**0.00**	0.02	**375.26**	**375.29**	2.76
7_5%	**1.72**	12.90	**306.92**	**311.30**	7.50	3.40	18.09	293.45	295.27	7.15
Total	**0.54**	7.23	**367.02**	**368.32**	8.57	0.74	8.80	365.31	365.82	8.29

LNS algorithms and the Critical Difference (CD) while considering a significance level $\alpha = 5\%$. Connections between algorithms indicate non-significant differences with respect to CD. From the results, we cannot conclude if the LNS algorithms based on FP_cuts differ significantly (in terms of the quality of the solutions obtained) from the corresponding ones based on OFP_cuts.

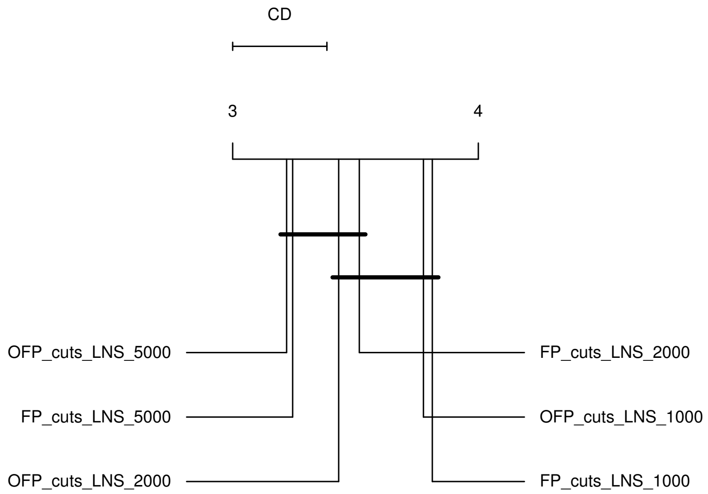


Figure 4.11: LNS algorithms' average ranks depicted on a number line, along with the Critical Difference (CD), when considering a significance level $\alpha = 5\%$. Connections between algorithms indicate non-significant differences.

On the other hand, the results clearly indicate that increasing the number of iterations from 1000 to 5000 does lead to a statistically significant improvement.

In Table 4.7, we summarize the number of instances for which the two best variations of the heuristic (FP_cuts_LNS_5000 and OFP_cuts_LNS_5000) were able to reach the best previously known bounds. For each algorithm and instance set, we also indicate the number of cases where the LNS algorithm improved over the solutions provided by CPA_1. In Appendix A.5, we provide a per-instance report of the primal bounds obtained by the LNS heuristic proposed.

Table 4.7: Number of times the two best algorithms (FP_cuts_LNS_5000 and OFP_cuts_LNS_5000) reached and/or improved over the best known bounds.

	FP_cuts_LNS_5000		OFP_cuts_LNS_5000	
Set	Reached best (#)	Improved best (#)	Reached best (#)	Improved best (#)
1_5%	54/54	0/54	54/54	0/54
2_5%	33/33	0/33	33/33	0/33
3_5%	60/60	0/60	60/60	0/60
4_5%	59/60	18/60	59/60	17/60
5_5%	76/78	3/78	76/78	3/78
6_5%	42/42	0/42	42/42	0/42
7_5%	58/60	0/60	54/60	0/60
Total	382/387	21/387	378/387	20/387

4.4.4 Results for the original benchmark of TOP instances

For completeness, the two variations of the heuristic that performed better for the STOP instances (FP_cuts_LNS_5000 and OFP_cuts_LNS_5000) were also run to solve the original benchmark of TOP instances. The results obtained are summarized in Tables 4.8, 4.9 and 4.10.

Table 4.8: Summary of the results obtained by FP_cuts and OFP_cuts while solving the original benchmark of TOP instances. Bold entries highlight, for each instance set, the best algorithm(s) in terms of average gaps and number of pumps.

	FP_cuts				OFP_cuts			
	Gap (%)				Gap (%)			
Set	Avg	StDev	Pumps (#)	Time (s)	Avg	StDev	Pumps (#)	Time (s)
1	27.91	26.17	**4.18**	0.09	**27.35**	24.43	21.97	0.45
2	35.96	31.49	**3.49**	0.01	**34.93**	30.14	21.51	0.03
3	**32.16**	23.69	**5.75**	0.14	32.49	21.29	29.97	0.59
4	24.72	13.85	**14.31**	27.57	**22.65**	12.12	41.03	62.89
5	**26.60**	22.23	**9.15**	2.68	27.22	22.98	31.64	8.81
6	17.06	19.64	**4.78**	0.57	**13.61**	17.83	20.09	2.94
7	31.50	21.01	**13.46**	59.77	**28.71**	17.91	35.22	112.67
Total	27.88	22.88	**8.44**	14.18	**26.76**	21.68	29.93	29.47

Table 4.9: Summary of the results obtained by FP_cuts_LNS_5000 and OFP_cuts_LNS_5000 while solving the original benchmark of TOP instances. The execution times do not consider the time spent by the FP algorithms in finding initial solutions. Bold entries highlight, for each instance set, the best algorithm(s) in terms of average gaps and profit sums.

	FP_cuts_LNS_5000					OFP_cuts_LNS_5000				
	Gap (%)		Profit sum			Gap (%)		Profit sum		
Set	Avg	StDev	Avg	Best	Time (s)	Avg	StDev	Avg	Best	Time (s)
1	1.67	9.66	111.07	**112.04**	0.65	**0.00**	0.00	**112.04**	**112.04**	0.65
2	6.67	15.14	131.45	**140.45**	0.18	**0.61**	2.42	**139.14**	**140.45**	0.19
3	3.17	13.96	411.22	**414.67**	1.18	**0.00**	0.00	**414.67**	**414.67**	1.18
4	0.34	3.11	801.22	804.23	38.20	**-0.17**	1.14	**802.63**	**804.25**	37.31
5	0.38	2.52	756.16	756.92	5.72	**0.00**	0.11	**756.74**	**756.99**	5.69
6	0.71	4.63	449.20	**450.57**	3.10	**0.24**	1.54	**450.11**	**450.57**	2.92
7	2.66	12.00	562.26	568.25	10.90	**0.14**	0.27	**567.33**	**568.55**	10.97
Total	1.91	9.59	503.01	506.14	9.39	**0.07**	1.00	**505.56**	**506.21**	9.24

In Table 4.8, we report, for each FP algorithm tested (FP_cuts and OFP_cuts) and instance set, four values: (i) the average and (ii) the standard deviation of the relative gaps given by $100 \cdot \frac{LB^* - LB}{LB^*}$, where LB^* is the best primal bound (for the instance) obtained by the baseline exact algorithm CPA_1, and LB is the average bound (over the 10 executions) obtained by the corresponding FP algorithm; (iii) the average number of iterations/pumps (over the 10 executions and all instances of the set), and (iv) the average wall-clock processing time (in seconds). We highlight that, whenever $LB^* = LB = 0$, the corresponding gap of (i) is set to 0%.

In Table 4.9, for each LNS algorithm tested (FP_cuts_LNS_5000 and OFP_cuts_LNS_5000) and instance set, we report five values: (i) the average and (ii) the standard deviation of the relative gaps given by $100 \cdot \frac{LB^* - LB}{LB^*}$, where LB is the average

Table 4.10: Number of times the two best algorithms (FP_cuts_LNS_5000 and OFP_cuts_LNS_5000) reached and/or improved over the primal bounds provided by CPA_1 while solving the original benchmark of TOP instances.

	FP_cuts_LNS_5000		OFP_cuts_LNS_5000	
Set	Reached best (#)	Improved best (#)	Reached best (#)	Improved best (#)
1	54/54	0/54	54/54	0/54
2	33/33	0/33	33/33	0/33
3	60/60	0/60	60/60	0/60
4	55/60	12/60	53/60	12/60
5	78/78	2/78	78/78	3/78
6	42/42	0/42	42/42	0/42
7	56/60	1/60	59/60	1/60
Total	378/387	15/387	379/387	16/387

bound (over the 10 executions) obtained by the corresponding algorithm. LB^* is the best primal bound (for the instance) obtained by the baseline exact algorithm (CPA_1); (iii) the average (over the 10 executions and all instances of the set) profit sum of the solutions obtained, and (iv) the average (over all instances of the set) profit sum of the best (in each round of 10 executions) solutions found. At last, we provide (v) the average wall-clock processing time (in seconds) of the execution of the LNS steps, excluding the time spent by the FP algorithm in finding the initial solution. Also in this case, whenever $LB^* = LB = 0$, the corresponding gap of (i) is set to 0%.

In Table 4.10, we summarize the number of instances for which the two best variations of the heuristic (FP_cuts_LNS_5000 and OFP_cuts_LNS_5000) were able to reach the primal bounds obtained by CPA_1. For each algorithm and instance set, we also indicate the number of cases where the LNS algorithm improved over the solutions provided by CPA_1. In Appendix A.5, we provide a per-instance report of the primal bounds obtained by the LNS heuristic proposed.

4.4.5 Summary of the results

Here, we summarize the main conclusions inferred from the results of the experiments performed to asses the behaviour of the heuristics proposed.

1. From Table 4.5 and the due statistical tests (see Figure 4.10), we conclude that:

   a) the FP algorithms that use the reinforced version of formulation $\mathcal{F}_2$ perform better (on average) both in terms of the quality of the solutions obtained and convergence (signaled by the number of pumps) than the ones that are based on the plain formulation $\mathcal{F}_2$;

   b) the FP algorithms based on the OFP framework obtain slightly better quality solutions (on average) than the ones that use the pure FP framework. Nevertheless, our tests could not conclude if this gain is statistically significant. Moreover, the better quality solutions of the OFP come at the expense of slower convergences;

   c) considering both indicators (solution quality and convergence), FP_cuts stands out as the best option;

2. From Tables 4.6 and 4.7 and the due statistical tests (see Figure 4.11):

   a) we cannot conclude if the LNS algorithms based on FP_cuts differ significantly (in terms of the quality of the solutions obtained) from the corresponding ones based on OFP_cuts;

   b) the results clearly indicate that increasing the number of iterations from 1000 to 5000 leads to a statistically significant improvement in terms of solution quality;

   c) in total, the heuristic proposed could reach the best previously known bounds on 382 of the 387 STOP instances tested and even improve them on 21 of these cases.

Chapter 5

Concluding remarks and future work directions

In this thesis, we introduced the Steiner Team Orienteering Problem (STOP), a generalization of capacitated routing problems with profit collection. The STOP is defined on a digraph in which arcs are associated with traverse times, and whose vertices are labeled as either *mandatory* or *profitable*, being the latter provided with rewards (profits). Given a homogeneous fleet of vehicles M, the goal is to find up to $m = |M|$ disjoint routes (from an origin vertex to a destination one) that maximize the total sum of rewards collected while satisfying a given limit on the route's duration. Naturally, all mandatory vertices must be visited.

We first extended to the problem a state-of-the-art branch-and-cut algorithm (referred to as Baseline Branch-and-Cut — B-B&C) from the literature of the Team Orienteering Problem (TOP), a closely related specialization of the STOP in which no mandatory vertices are considered. Then, we proposed other exact and heuristic algorithms to solve the STOP. The algorithms are based on a compact (with a polynomial number of variables and constraints) commodity-based MILP formulation we also introduced in this work. We also devised the formal proof that our formulation gives the same bounds as the one used within B-B&C.

Regarding exact solutions, we proposed two variations of Cutting-Plane Algorithms (CPA) which work by reinforcing our formulation by means of the separation of five families of inequalities: (i) General Connectivity Constraints (GCCs), (ii) classical Lifted Cover Inequalities (LCIs) based on dual bounds, (iii) Arc-Vertex Inference Cuts (AVICs), (iv) Conflict Cuts (CCs) and (v) Clique Conflict Cuts (CCCs). To our knowledge, the last three classes of inequalities are also introduced in this thesis. The due proofs regarding the validity of the new inequalities and some dominance results are also given, along with examples of the fractional solutions cut off by the inequalities.

Extensive computational experiments showed that CPA is highly competitive in solving a benchmark of TOP instances. In particular, the algorithm solved, in total, 14 more instances than any other exact algorithm in the literature of the TOP. Moreover, our approach was able to find optimal solutions for nine previously unsolved instances. Regarding the new STOP instances introduced in this work, our algorithm solved 31 more instances than B-B&C. From the results, we concluded that CPA benefits from both the strength and compactness of the model used as backbone, as well as from the reinforcement provided by the aforementioned families of inequalities.

Regarding heuristic approaches, we first discussed the additional difficulty of the STOP by showing that solely finding a feasible solution for the problem is NP-hard. To address the issue of finding initial solutions, we made use of the Feasibility Pump (FP) matheuristic of Fischetti et al. (2005). As a matheuristic, FP uses a mathematical formulation to guide its search for a feasible solution, which, in this case, consists of the compact formulation we proposed. In order to refine the initial solutions obtained by FP, we proposed a Large Neighborhood Search (LNS) heuristic that combines classical local searches from the literature of routing problems with a memory component based on Path Relinking (PR).

We used the primal bounds provided by CPA to guide our computational experiments, which showed the efficiency and effectiveness of the proposed heuristic in solving a benchmark of 387 STOP instances. In particular, the heuristic could reach the best previously known bounds on 382 of these instances. Additionally, in 21 of these cases, our heuristic was even able to improve over the best known bounds. Overall, the heuristic solutions imply an average percentage gap of only 0.54% when compared to the bounds of CPA.

For completeness, we also tested the hybrid algorithm obtained from warm starting CPA with the solutions provided by our LNS heuristic. The computational results show that this approach is able to improve the convergence of CPA. In total, the hybrid algorithm is able to solve to optimality four more TOP instances and seven more STOP instances than CPA alone. Additionally, it provides the optimality certificates of five previously unsolved (even by CPA) TOP instances.

Future works

In terms of future work directions, we first remark that our commodity-based formulation and the three new classes of inequalities here proposed can be naturally extended for other routing problems. In particular, we believe that our algorithms might be also successful in solving a closely related problem known as the orienteering arc routing problem (Archetti et al., 2016), in which arcs (instead of vertices) are sub-divided into mandatory and profitable.

During the research, we have also made some pre-processing attempts by considering reduced costs and other bounds incurred by fixing and removing vertices/arcs of the original problem. These attempts were not successful, specially because of the time overhead of individually solving several MILP auxiliary problems (one for each vertex/arc fixation/removal). Nevertheless, we still believe that investing in pre-processing procedures is a direction of research worthy to follow in this case.

In terms of exact algorithms, we believe that only innovative approaches can lead to further significant improvements for the problem. In the particular case of our formulation, we noticed that the reinforcements proposed (the families of valid inequalities) only pay off to a certain extent, as separating the inequalities in other nodes of the branch-and-bound tree — aside from the root — showed to slow down the convergence of the algorithms, as discussed by the end of Section 3.2. In summary, the loss in compactness has a greater impact than the gain in strengthening in this case.

Regarding heuristics, there is still a wide range of possibilities to explore. Once we proved that finding an initial solution for the STOP is NP-hard — and our experiments indicated that this task can be time consuming — we believe that investigating alternative ways to build initial solutions might be a promising research direction for the problem. In this sense, other rounding and diving matheuristics like Relaxation Induced Neighborhood Search (RINS) (Danna et al., 2005) could be tested. In addition, there are plenty of improvement metaheuristic frameworks to be tested for the problem, including the ones already applied to the TOP (as detailed in Section 1.1). In particular, we anticipate that we have already done some pilot experiments on applying Local Branching (Fischetti and Lodi, 2003) to refine initial solutions for the STOP and the results were quite poor. Precisely, we could only achieve minor solution improvements at the cost of great computational effort. Thus, this is a research path we do not encourage.

Bibliography

Achterberg, T. and Berthold, T. (2007). Improving the feasibility pump. *Discrete Optimization*, 4(1):77 – 86.

Archetti, C., Corberán, A., Plana, I., Sanchis, J. M., and Speranza, M. G. (2016). A branch-and-cut algorithm for the orienteering arc routing problem. *Computers & Operations Research*, 66:95 – 104.

Archetti, C., Hertz, A., and Speranza, M. G. (2007). Metaheuristics for the team orienteering problem. *Journal of Heuristics*, 13(1):49–76.

Balas, E. (1975). Facets of the knapsack polytope. *Mathematical Programming*, 8(1):146--164.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition.

Berthold, T., Lodi, A., and Salvagnin, D. (2019). Ten years of feasibility pump, and counting. *EURO Journal on Computational Optimization*, 7(1):1--14.

Bianchessi, N., Mansini, R., and Speranza, M. G. (2018). A branch-and-cut algorithm for the team orienteering problem. *International Transactions in Operational Research*, 25(2):627--635.

Bicalho, L. H., da Cunha, A. S., and Lucena, A. (2016). Branch-and-cut-and-price algorithms for the degree constrained minimum spanning tree problem. *Computational Optimization and Applications*, 63(3):755--792.

Boussier, S., Feillet, D., and Gendreau, M. (2007). An exact algorithm for team orienteering problems. *4OR*, 5(3):211–230.

Butt, S. E. and Cavalier, T. M. (1994). A heuristic for the multiple tour maximum collection problem. *Computers & Operations Research*, 21(1):101–111.

Butt, S. E. and Ryan, D. M. (1999). An optimal solution procedure for the multiple tour maximum collection problem using column generation. *Computers & Operations Research*, 26(4):427–441.

Chao, I.-M., Golden, B. L., and Wasil, E. A. (1996). The team orienteering problem. *European Journal of Operational Research*, 88(3):464–474.

Cormen, T. H., Stein, C., Rivest, R. L., and Leiserson, C. E. (2001). *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition.

Dang, D.-C., El-Hajj, R., and Moukrim, A. (2013a). A branch-and-cut algorithm for solving the team orienteering problem. In Gomes, C. and Sellmann, M., editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 7874 of *Lecture Notes in Computer Science*, pages 332–339. Springer Berlin Heidelberg.

Dang, D.-C., Guibadj, R. N., and Moukrim, A. (2013b). An effective PSO-inspired algorithm for the team orienteering problem. *European Journal of Operational Research*, 229(2):332–344.

Danna, E., Rothberg, E., and Pape, C. L. (2005). Exploring relaxation induced neighborhoods to improve MIP solutions. *Mathematical Programming*, 102(1):71--90.

Dantzig, G. B., Fulkerson, D. R., and Johnson, S. M. (1954). Solution of a large-scale traveling-salesman problem. *Operations Research*, 2:393–410.

de Freitas Viana, F. H. (2011). *Modelos e Algoritmos para o Team Orienteering Problem*. PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro, Brazil.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1--30.

Dezsõ, B., Jüttner, A., and Péter Kovács (2011). LEMON — an open source C++ graph template library. *Electronic Notes in Theoretical Computer Science*, 264(5):23 -- 45. Proceedings of the Second Workshop on Generative Technologies (WGT) 2010.

El-Hajj, R., Dang, D.-C., and Moukrim, A. (2016). Solving the team orienteering problem with cutting planes. *Computers & Operations Research*, 74:21 – 30.

Fischetti, M., Glover, F., and Lodi, A. (2005). The feasibility pump. *Mathematical Programming*, 104(1):91--104.

Fischetti, M., González, J. J. S., and Toth, P. (1998). Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*, 10(2):133–148.

Fischetti, M. and Lodi, A. (2003). Local branching. *Mathematical Programming*, 98(1):23--47.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675--701.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.

Glover, F. (1997). *Tabu Search and Adaptive Memory Programming — Advances, Applications and Challenges*, pages 1--75. Springer US, Boston, MA.

Goldberg, A. V. and Tarjan, R. E. (1988). A new approach to the maximum-flow problem. *J. ACM*, 35(4):921--940.

Gu, Z., Nemhauser, G. L., and Savelsbergh, M. W. P. (1998). Lifted cover inequalities for 0–1 integer programs: Computation. *INFORMS Journal on Computing*, 10(4):427–437.

Gunawan, A., Lau, H. C., and Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315 – 332.

Iman, R. and Davenport, J. (1980). Approximations of the critical region of the Friedman statistic. *Communications in Statistics*, 9:571–595.

Kaparis, K. and Letchford, A. N. (2008). Local and global lifted cover inequalities for the 0–1 multidimensional knapsack problem. *European Journal of Operational Research*, 186(1):91 – 103.

Kara, I., Laporte, G., and Bektas, T. (2004). A note on the lifted Miller-Tucker-Zemlin subtour elimination constraints for the capacitated vehicle routing problem. *European Journal of Operational Research*, 158(3):793 – 795.

Ke, L., Archetti, C., and Feng, Z. (2008). Ants can solve the team orienteering problem. *Computers & Industrial Engineering*, 54(3):648–665.

Ke, L., Zhai, L., Li, J., and Chan, F. T. (2016). Pareto mimic algorithm: An approach to the team orienteering problem. *Omega*, 61:155 – 166.

Keshtkaran, M., Ziarati, K., Bettinelli, A., and Vigo, D. (2016). Enhanced exact solution methods for the team orienteering problem. *International Journal of Production Research*, 54(2):591–601.

Kim, B.-I., Li, H., and Johnson, A. L. (2013). An augmented large neighborhood search method for solving the team orienteering problem. *Expert Systems with Applications*, 40(8):3065–3072.

Kulkarni, R. and Bhave, P. (1985). Integer programming formulations of vehicle routing problems. *European Journal of Operational Research*, 20(1):58–67.

Laporte, G. and Martello, S. (1990). The selective travelling salesman problem. *Discrete Applied Mathematics*, 26(2–3):193–207.

Letchford, A. N., Nasiri, S. D., and Theis, D. O. (2013). Compact formulations of the Steiner traveling salesman problem and related problems. *European Journal of Operational Research*, 228(1):83–92.

Lin, S. (1965). Computer solutions of the traveling salesman problem. *The Bell System Technical Journal*, 44(10):2245–2269.

Lin, S.-W. (2013). Solving the team orienteering problem using effective multi-start simulated annealing. *Applied Soft Computing*, 13(2):1064–1073.

Maffioli, F. and Sciomachen, A. (1997). A mixed-integer model for solving ordering problems with side constraints. *Annals of Operations Research*, 69(0):277--297.

Miller, C. E., Tucker, A. W., and Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *J. ACM*, 7(4):326--329.

Nemenyi, P. (1963). *Distribution-free Multiple Comparisons.* Princeton University.

Poggi, M., Viana, H., and Uchoa, E. (2010). The team orienteering problem: Formulations and branch-cut and price. In *ATMOS 2010 - 10th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems, Liverpool, United Kingdom, September 6-10, 2010*, pages 142–155.

Samer, P. and Urrutia, S. (2015). A branch and cut algorithm for minimum spanning trees under conflict constraints. *Optimization Letters*, 9(1):41--55.

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In Maher, M. and Puget, J.-F., editors, *Principles and Practice of Constraint Programming — CP98*, pages 417--431, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sheskin, D. J. (2007). *Handbook of Parametric and Nonparametric Statistical Procedures.* Chapman & Hall/CRC, 4th edition.

Souffriau, W., Vansteenwegen, P., Berghe, G. V., and Oudheusden, D. V. (2010). A path relinking approach for the team orienteering problem. *Computers & Operations Research*, 37(11):1853–1859.

Tang, H. and Miller-Hooks, E. (2005). A TABU search heuristic for the team orienteering problem. *Computers & Operations Research*, 32(6):1379–1407.

Tomita, E., Tanaka, A., and Takahashi, H. (2006). The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28 – 42.

Toth, P. and Vigo, D., editors (2001). *The Vehicle Routing Problem.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Tsiligirides, T. (1984). Heuristic methods applied to orienteering. *The Journal of the Operational Research Society*, 35(9):797–809.

Vansteenwegen, P., Souffriau, W., Berghe, G. V., and Oudheusden, D. V. (2009). A guided local search metaheuristic for the team orienteering problem. *European Journal of Operational Research*, 196(1):118–127.

Vansteenwegen, P., Souffriau, W., and Oudheusden, D. V. (2011). The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1 – 10.

Wesselmann, F. and Suhl, U. H. (2012). Implementing cutting plane management and selection techniques. Technical report, University of Paderborn, Paderborn, Germany.

Wolsey, L. A. (1975). Faces for a linear inequality in 0–1 variables. *Mathematical Programming*, 8(1):165--178.

Appendix A

A.1 Bound comparison between $\mathcal{F}_1$ and $\mathcal{F}_2$

Here, we summarize the results obtained from our experimental analysis of the impact of the valid inequalities (2.12) and (2.22) on the strength of formulations $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively. The results for the original benchmark of TOP instances of Chao et al. (1996) and the new STOP instances introduced in this work are detailed in Tables A.1 and A.2, respectively. In both tables, the first column displays the name of each instance set. Then, we give the average and the standard deviation (over all the instances in each set) of the percentage bound improvements referred to (2.12) on $\mathcal{F}_1$ and to (2.22) on $\mathcal{F}_2$. In both tables, the last row displays the numerical results while considering the complete benchmark of instances.

Without loss of generality, given an instance, the percentage bound improvement referred to (2.12) on $\mathcal{F}_1$ is given by $100 \cdot \frac{UB_{\mathcal{L}_1 \setminus (2.12)} - UB_{\mathcal{L}_1}}{UB_{\mathcal{L}_1 \setminus (2.12)}}$, where $UB_{\mathcal{L}_1}$ denotes the bound provided by $\mathcal{L}_1$ (the linear relaxation of $\mathcal{F}_1$), and $UB_{\mathcal{L}_1 \setminus (2.12)}$ stands for the bound obtained from solving $\mathcal{L}_1$ without inequalities (2.12). Likewise, given an instance, the percentage bound improvement referred to (2.22) on $\mathcal{F}_2$ is given by $100 \cdot \frac{UB_{\mathcal{L}_2 \setminus (2.22)} - UB_{\mathcal{L}_2}}{UB_{\mathcal{L}_2 \setminus (2.22)}}$, where $UB_{\mathcal{L}_2}$ denotes the bound provided by $\mathcal{L}_2$ (the linear relaxation of $\mathcal{F}_2$), and $UB_{\mathcal{L}_2 \setminus (2.22)}$ stands for the bound obtained from solving $\mathcal{L}_2$ without inequalities (2.22). Recall that, from Theorem 1, $UB_{\mathcal{L}_1}$ is always equal to $UB_{\mathcal{L}_2}$.

Table A.1: Impact — in terms of percentage bound improvement — of the valid inequalities (2.12) and (2.22) on the linear relaxations of $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively. Results for the original benchmark of TOP instances.

	$\mathcal{L}_1$ — with (2.12)		$\mathcal{L}_2$ — with (2.22)	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1	4.52	4.20	2.70	2.79
2	0.33	0.97	0.30	0.84
3	3.42	3.08	2.12	1.85
4	2.38	2.68	2.29	2.53
5	3.09	3.34	1.89	2.10
6	0.18	0.47	0.17	0.47
7	4.73	4.65	3.50	3.61
Total	2.93	3.60	2.03	2.56

Table A.2: Impact — in terms of percentage bound improvement — of the valid inequalities (2.12) and (2.22) on the linear relaxations of $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively. Results for the new STOP instances.

	$\mathcal{L}_1$ — with (2.12)		$\mathcal{L}_2$ — with (2.22)	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1_5%	3.51	4.96	2.06	3.12
2_5%	0.17	0.99	0.15	0.85
3_5%	2.30	3.05	1.39	1.86
4_5%	1.78	3.13	1.83	3.20
5_5%	2.70	6.20	1.89	5.11
6_5%	0.23	0.67	0.22	0.66
7_5%	2.61	4.97	2.01	3.95
Total	2.11	4.36	1.52	3.40

A.2 Proof of Proposition 4

Proposition 4. *If the cover C is minimal, then performing down-lifting on any of the variables y_j, $j \in C$, of the original cover inequality (2.60) is ineffective, as it always leads to lifted coefficients equal to one.*

Proof. Consider that C is minimal, and let C_1 and C_2 define an arbitrary partition of C, with $C_1 \neq \emptyset$. In this case, also consider $C_2 \neq \emptyset$, since the opposite implies no variable being down-lifted. The proof is done by induction on $|C'|$, i.e., the number of variables y_i, for all $i \in C_2$, whose lifted coefficients have already been computed. In this sense, given a non-negative integer n, consider the following property.

P(n): *For any configuration of $C' \subset C_2$, $|C'| = n$, performing down-lifting on any of the variables y_j, $j \in C_2 \backslash C'$, of (2.61) always leads to lifted coefficients equal to one.*

We want to prove that $P(n)$ holds for all $n \in \mathbb{Z}$, $0 \leq n \leq |C_2| - 1$.

Base case. In the base case, $n = 0$, and, thus, $C' = \emptyset$. Let $j \in C_2$ be the index of the first variable to be down-lifted. During the lifting of y_j, the corresponding auxiliary knapsack problem $\mathcal{A}_1$ assumes the form

$$\max \left\{ \sum_{i \in C_1} y_i + \overbrace{\sum_{i \in C'} \pi_i y_i}^{=0} : \text{(2.51)}, \, y_j = 0, \, y_i = 1 \, \forall \, i \in C_2 \backslash \{j\}, \, y \in \{0,1\}^{|P|} \right\}.$$

Since C is a minimal cover for (2.51), the solution $\bar{y}$, with $\bar{y}_j = 0$, $\bar{y}_i = 1$ for all $i \in C \backslash \{j\}$ and $\bar{y}_k = 0$ for all $k \in P \backslash C$, is feasible and, thus, optimal for this maximization problem. Accordingly, the optimal value of this problem is given by $\sum_{i \in C_1} 1 = |C_1|$, and the lifted coefficient is $\pi_j = |C_1| - (|C_1| + \sum_{i \in C'} \pi_i - 1)$. Since $C' = \emptyset$, we have $\pi_j = |C_1| - (|C_1| - 1) = 1$.

Induction step. Given a value $k \in \mathbb{Z}$, $0 \leq k < |C_2| - 1$, assume, as inductive hypothesis, that $P(n')$ is true for all $n' \in \mathbb{Z}$, $0 \leq n' \leq k$. Thus, we must show that $P(k+1)$ is also true. To this end, consider an arbitrary $C' \subset C_2$, with $|C'| = k + 1$. Notice that, in this case, $|C'|$ is at most $|C_2| - 1$. Since $P(n')$ is supposedly true for all $n' \in \mathbb{Z}$, $0 \leq n' \leq k$, we necessarily have

$$\pi_i = 1 \quad \forall \, i \in C'. \tag{A.1}$$

Without loss of generality, assume that one wants to down-lift a variable y_j, with $j \in C_2 \backslash C'$. Then, from (A.1), the corresponding auxiliary knapsack problem $\mathcal{A}_1$ assumes the form

$$\max \left\{ \sum_{i \in C_1} y_i + \sum_{i \in C'} y_i : \text{(2.51)}, \, y_j = 0, \, y_i = 1 \, \forall \, i \in C_2 \backslash (C' \cup \{j\}), \, y \in \{0,1\}^{|P|} \right\}.$$

Since C is a minimal cover for (2.51), the solution $\bar{y}$, with $\bar{y}_j = 0$, $\bar{y}_i = 1$ for all $i \in C \backslash \{j\}$ and $\bar{y}_k = 0$ for all $k \in P \backslash C$, is also feasible (and optimal) for this auxiliary problem.

Accordingly, the optimal value of this problem is given by $\sum_{i \in C_1} 1 + \sum_{i \in C'} 1 = |C_1| + |C'|$, and the lifted coefficient is $\pi_j = |C_1| + |C'| - (|C_1| + \sum_{i \in C'} \pi_i - 1)$. From (A.1), $\pi_j = |C_1| + |C'| - (|C_1| + |C'| - 1) = 1$. □

A.3 Detailed results for the exact algorithms at solving the original benchmark of TOP instances

In Tables A.3 and A.4, we display some information on the bounds available at the root nodes of the branch-and-bound trees of B-B&C and CPA while solving the original benchmark of TOP instances. In Table A.3, for each algorithm and instance set, we give the averages and the corresponding standard deviations of three gap values. The first one is the relative optimality gap, given by $\frac{UB_r - LB_r}{UB_r}$, where LB_r and UB_r are, respectively, the lower and upper bounds at the root node. The second and third gaps, namely *primal* and *dual* gaps, are given by $\frac{LB^* - LB_r}{UB_r}$ and $\frac{UB_r - LB^*}{UB_r}$, respectively. Here, LB^* corresponds to the best lower bound obtained by B-B&C or CPA for a given instance after two hours of execution. Notice that, for every instance, the summation of the corresponding primal and dual gaps gives the root relative optimality gap. Then, these last two gaps are given as a way to estimate the contribution of the primal and the dual bounds to the actual optimality gap. All the bounds considered in Table A.3 also take into account the CPLEX built-in cuts separated at the root nodes. In Table A.4, we give the same information displayed in Table A.3, but for formulations $\mathcal{F}_1$ and $\mathcal{F}_2$ without the addition of the inequalities proposed in section 2.2. Recall that $\mathcal{F}_1$ and $\mathcal{F}_2$ are the formulations solved within B-B&C and CPA, respectively. Also in this case, the CPLEX built-in cuts are enabled.

In Table A.5, we display some extra information concerning the execution of B-B&C and CPA at solving the original benchmark of TOP instances. For each algorithm and instance set, we first display the average and the standard deviation of the relative optimality gaps of the solutions obtained within two hours of execution. Here, we consider the results for the complete benchmark, and not only the instances not solved to optimality (as in Table 3.6). In Table A.5, we also expose the average number of nodes explored in the branch-and-bound tree, as well as the average number of cuts separated for each class of inequalities discussed in section 2.2. In the results for CPA_2, we omitted the average number of AVICs, as they are separated by complete enumeration and the number of these cuts in each instance always corresponds to $2 \times |E|$.

In Table A.6, we display a per-instance report of the results obtained by our implementations of B-B&C and CPA at solving the original benchmark of TOP instances. For all algorithms, we report, for each instance, the best lower and upper bounds obtained within two hours of execution (columns "LB" and "UB", respectively) and the wall-clock execution time in seconds. We marked in bold the instances for which CPA found previously unknown optimality certificates.

Table A.6: Detailed results for the exact algorithms at solving the original benchmark of TOP instances. Instances in bold are the ones for which CPA found previously unknown optimality certificates.

	B-B&C			CPA_1			CPA_2		
Instance	LB	UB	Time (s)	LB	UB	Time (s)	LB	UB	Time (s)
p1.2.a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p1.2.b	15.00	15.00	0.00	15.00	15.00	0.00	15.00	15.00	0.00

Continued on next page

Table A.6 — continued from previous page.

	B-B&C			CPA_1			CPA_2		
Instance	LB	UB	Time (s)	LB	UB	Time (s)	LB	UB	Time (s)
p1.2.c	20.00	20.00	0.00	20.00	20.00	0.00	20.00	20.00	0.00
p1.2.d	30.00	30.00	0.00	30.00	30.00	0.01	30.00	30.00	0.01
p1.2.e	45.00	45.00	0.01	45.00	45.00	0.03	45.00	45.00	0.05
p1.2.f	80.00	80.00	0.03	80.00	80.00	0.06	80.00	80.00	0.06
p1.2.g	90.00	90.00	0.39	90.00	90.00	0.98	90.00	90.00	0.77
p1.2.h	110.00	110.00	0.70	110.00	110.00	1.09	110.00	110.00	2.97
p1.2.i	135.00	135.00	2.27	135.00	135.00	3.52	135.00	135.00	4.83
p1.2.j	155.00	155.00	0.97	155.00	155.00	1.11	155.00	155.00	1.28
p1.2.k	175.00	175.00	1.17	175.00	175.00	2.38	175.00	175.00	2.97
p1.2.l	195.00	195.00	1.36	195.00	195.00	2.85	195.00	195.00	2.53
p1.2.m	215.00	215.00	1.89	215.00	215.00	0.89	215.00	215.00	1.17
p1.2.n	235.00	235.00	0.65	235.00	235.00	3.45	235.00	235.00	2.89
p1.2.o	240.00	240.00	1.66	240.00	240.00	3.48	240.00	240.00	3.29
p1.2.p	250.00	250.00	1.62	250.00	250.00	3.04	250.00	250.00	3.72
p1.2.q	265.00	265.00	1.76	265.00	265.00	3.72	265.00	265.00	4.63
p1.2.r	280.00	280.00	2.41	280.00	280.00	2.06	280.00	280.00	2.45
p1.3.a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p1.3.b	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p1.3.c	15.00	15.00	0.00	15.00	15.00	0.00	15.00	15.00	0.00
p1.3.d	15.00	15.00	0.00	15.00	15.00	0.00	15.00	15.00	0.00
p1.3.e	30.00	30.00	0.00	30.00	30.00	0.01	30.00	30.00	0.00
p1.3.f	40.00	40.00	0.00	40.00	40.00	0.01	40.00	40.00	0.01
p1.3.g	50.00	50.00	0.02	50.00	50.00	0.03	50.00	50.00	0.03
p1.3.h	70.00	70.00	0.03	70.00	70.00	0.07	70.00	70.00	0.08
p1.3.i	105.00	105.00	0.05	105.00	105.00	0.14	105.00	105.00	0.21
p1.3.j	115.00	115.00	0.36	115.00	115.00	2.02	115.00	115.00	2.62
p1.3.k	135.00	135.00	0.50	135.00	135.00	2.76	135.00	135.00	3.93
p1.3.l	155.00	155.00	0.95	155.00	155.00	6.09	155.00	155.00	2.96
p1.3.m	175.00	175.00	1.22	175.00	175.00	3.77	175.00	175.00	2.91
p1.3.n	190.00	190.00	3.14	190.00	190.00	13.82	190.00	190.00	11.58
p1.3.o	205.00	205.00	2.77	205.00	205.00	6.67	205.00	205.00	6.17
p1.3.p	220.00	220.00	0.68	220.00	220.00	1.35	220.00	220.00	2.80
p1.3.q	230.00	230.00	2.88	230.00	230.00	8.40	230.00	230.00	6.45
p1.3.r	250.00	250.00	3.06	250.00	250.00	6.51	250.00	250.00	5.71
p1.4.a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p1.4.b	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p1.4.c	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p1.4.d	15.00	15.00	0.00	15.00	15.00	0.00	15.00	15.00	0.00
p1.4.e	15.00	15.00	0.00	15.00	15.00	0.00	15.00	15.00	0.00
p1.4.f	25.00	25.00	0.00	25.00	25.00	0.00	25.00	25.00	0.00
p1.4.g	35.00	35.00	0.00	35.00	35.00	0.01	35.00	35.00	0.00
p1.4.h	45.00	45.00	0.00	45.00	45.00	0.01	45.00	45.00	0.00
p1.4.i	60.00	60.00	0.01	60.00	60.00	0.02	60.00	60.00	0.02
p1.4.j	75.00	75.00	0.02	75.00	75.00	0.03	75.00	75.00	0.04
p1.4.k	100.00	100.00	0.04	100.00	100.00	0.08	100.00	100.00	0.09
p1.4.l	120.00	120.00	0.31	120.00	120.00	0.46	120.00	120.00	0.23
p1.4.m	130.00	130.00	0.46	130.00	130.00	3.40	130.00	130.00	2.81
p1.4.n	155.00	155.00	0.51	155.00	155.00	3.89	155.00	155.00	2.65
p1.4.o	165.00	165.00	0.67	165.00	165.00	5.22	165.00	165.00	2.75
p1.4.p	175.00	175.00	0.55	175.00	175.00	2.53	175.00	175.00	2.61
p1.4.q	190.00	190.00	2.04	190.00	190.00	3.66	190.00	190.00	3.28
p1.4.r	210.00	210.00	0.73	210.00	210.00	3.62	210.00	210.00	3.69
p2.2.a	90.00	90.00	0.01	90.00	90.00	0.01	90.00	90.00	0.03
p2.2.b	120.00	120.00	0.01	120.00	120.00	0.02	120.00	120.00	0.03
p2.2.c	140.00	140.00	0.05	140.00	140.00	0.05	140.00	140.00	0.05

Continued on next page

Table A.6 — continued from previous page.

Instance	B-B&C LB	B-B&C UB	B-B&C Time (s)	CPA_1 LB	CPA_1 UB	CPA_1 Time (s)	CPA_2 LB	CPA_2 UB	CPA_2 Time (s)
p2.2.d	160.00	160.00	0.06	160.00	160.00	0.13	160.00	160.00	0.16
p2.2.e	190.00	190.00	0.16	190.00	190.00	0.20	190.00	190.00	0.17
p2.2.f	200.00	200.00	0.07	200.00	200.00	0.10	200.00	200.00	0.08
p2.2.g	200.00	200.00	0.05	200.00	200.00	0.10	200.00	200.00	0.08
p2.2.h	230.00	230.00	0.07	230.00	230.00	0.08	230.00	230.00	0.14
p2.2.i	230.00	230.00	0.04	230.00	230.00	0.13	230.00	230.00	0.13
p2.2.j	260.00	260.00	0.08	260.00	260.00	0.76	260.00	260.00	0.45
p2.2.k	275.00	275.00	1.22	275.00	275.00	2.14	275.00	275.00	1.85
p2.3.a	70.00	70.00	0.00	70.00	70.00	0.01	70.00	70.00	0.00
p2.3.b	70.00	70.00	0.00	70.00	70.00	0.00	70.00	70.00	0.00
p2.3.c	105.00	105.00	0.02	105.00	105.00	0.01	105.00	105.00	0.01
p2.3.d	105.00	105.00	0.00	105.00	105.00	0.01	105.00	105.00	0.01
p2.3.e	120.00	120.00	0.00	120.00	120.00	0.01	120.00	120.00	0.02
p2.3.f	120.00	120.00	0.01	120.00	120.00	0.02	120.00	120.00	0.03
p2.3.g	145.00	145.00	0.08	145.00	145.00	0.07	145.00	145.00	0.09
p2.3.h	165.00	165.00	0.04	165.00	165.00	0.11	165.00	165.00	0.12
p2.3.i	200.00	200.00	0.04	200.00	200.00	0.13	200.00	200.00	0.15
p2.3.j	200.00	200.00	0.07	200.00	200.00	0.10	200.00	200.00	0.10
p2.3.k	200.00	200.00	0.05	200.00	200.00	0.01	200.00	200.00	0.01
p2.4.a	10.00	10.00	0.00	10.00	10.00	0.00	10.00	10.00	0.00
p2.4.b	70.00	70.00	0.00	70.00	70.00	0.00	70.00	70.00	0.00
p2.4.c	70.00	70.00	0.00	70.00	70.00	0.01	70.00	70.00	0.00
p2.4.d	70.00	70.00	0.00	70.00	70.00	0.00	70.00	70.00	0.00
p2.4.e	70.00	70.00	0.00	70.00	70.00	0.01	70.00	70.00	0.00
p2.4.f	105.00	105.00	0.00	105.00	105.00	0.01	105.00	105.00	0.00
p2.4.g	105.00	105.00	0.00	105.00	105.00	0.00	105.00	105.00	0.00
p2.4.h	120.00	120.00	0.01	120.00	120.00	0.01	120.00	120.00	0.00
p2.4.i	120.00	120.00	0.01	120.00	120.00	0.02	120.00	120.00	0.02
p2.4.j	120.00	120.00	0.01	120.00	120.00	0.01	120.00	120.00	0.02
p2.4.k	180.00	180.00	0.03	180.00	180.00	0.04	180.00	180.00	0.05
p3.2.a	90.00	90.00	0.00	90.00	90.00	0.01	90.00	90.00	0.01
p3.2.b	150.00	150.00	0.01	150.00	150.00	0.03	150.00	150.00	0.03
p3.2.c	180.00	180.00	0.14	180.00	180.00	0.19	180.00	180.00	0.21
p3.2.d	220.00	220.00	0.12	220.00	220.00	0.19	220.00	220.00	0.24
p3.2.e	260.00	260.00	0.61	260.00	260.00	2.71	260.00	260.00	2.27
p3.2.f	300.00	300.00	1.37	300.00	300.00	7.75	300.00	300.00	4.20
p3.2.g	360.00	360.00	1.23	360.00	360.00	9.08	360.00	360.00	5.06
p3.2.h	410.00	410.00	5.91	410.00	410.00	10.00	410.00	410.00	10.98
p3.2.i	460.00	460.00	3.06	460.00	460.00	10.55	460.00	460.00	11.57
p3.2.j	510.00	510.00	11.51	510.00	510.00	26.41	510.00	510.00	12.10
p3.2.k	550.00	550.00	26.32	550.00	550.00	16.60	550.00	550.00	10.95
p3.2.l	590.00	590.00	3.30	590.00	590.00	8.46	590.00	590.00	5.64
p3.2.m	620.00	620.00	3.74	620.00	620.00	8.02	620.00	620.00	6.06
p3.2.n	660.00	660.00	1.36	660.00	660.00	5.28	660.00	660.00	5.34
p3.2.o	690.00	690.00	2.97	690.00	690.00	4.97	690.00	690.00	3.40
p3.2.p	720.00	720.00	5.15	720.00	720.00	8.62	720.00	720.00	7.69
p3.2.q	760.00	760.00	1.80	760.00	760.00	5.69	760.00	760.00	6.05
p3.2.r	790.00	790.00	14.99	790.00	790.00	9.27	790.00	790.00	7.93
p3.2.s	800.00	800.00	3.73	800.00	800.00	4.83	800.00	800.00	0.92
p3.2.t	800.00	800.00	1.73	800.00	800.00	0.83	800.00	800.00	1.09
p3.3.a	30.00	30.00	0.00	30.00	30.00	0.00	30.00	30.00	0.00
p3.3.b	90.00	90.00	0.00	90.00	90.00	0.01	90.00	90.00	0.00
p3.3.c	120.00	120.00	0.00	120.00	120.00	0.01	120.00	120.00	0.01
p3.3.d	170.00	170.00	0.01	170.00	170.00	0.03	170.00	170.00	0.06
p3.3.e	200.00	200.00	0.03	200.00	200.00	0.06	200.00	200.00	0.04

Continued on next page

Table A.6 — continued from previous page.

Instance	B-B&C			CPA_1			CPA_2		
	LB	UB	Time (s)	LB	UB	Time (s)	LB	UB	Time (s)
p3.3.f	230.00	230.00	0.19	230.00	230.00	0.36	230.00	230.00	0.53
p3.3.g	270.00	270.00	0.20	270.00	270.00	0.66	270.00	270.00	0.53
p3.3.h	300.00	300.00	0.65	300.00	300.00	5.88	300.00	300.00	1.58
p3.3.i	330.00	330.00	6.36	330.00	330.00	20.58	330.00	330.00	22.07
p3.3.j	380.00	380.00	7.07	380.00	380.00	20.09	380.00	380.00	25.01
p3.3.k	440.00	440.00	4.62	440.00	440.00	25.52	440.00	440.00	10.34
p3.3.l	480.00	480.00	29.98	480.00	480.00	53.74	480.00	480.00	112.66
p3.3.m	520.00	520.00	66.18	520.00	520.00	119.05	520.00	520.00	101.19
p3.3.n	570.00	570.00	11.57	570.00	570.00	14.34	570.00	570.00	11.42
p3.3.o	590.00	590.00	544.59	590.00	590.00	456.10	590.00	590.00	364.39
p3.3.p	640.00	640.00	1170.08	640.00	640.00	2843.94	640.00	640.00	1510.00
p3.3.q	680.00	680.00	26.73	680.00	680.00	47.17	680.00	680.00	19.53
p3.3.r	710.00	710.00	1.60	710.00	710.00	8.18	710.00	710.00	10.86
p3.3.s	720.00	720.00	53.40	720.00	720.00	116.35	720.00	720.00	33.41
p3.3.t	760.00	760.00	115.41	760.00	760.00	91.57	760.00	760.00	153.61
p3.4.a	20.00	20.00	0.00	20.00	20.00	0.00	20.00	20.00	0.00
p3.4.b	30.00	30.00	0.00	30.00	30.00	0.00	30.00	30.00	0.00
p3.4.c	90.00	90.00	0.00	90.00	90.00	0.00	90.00	90.00	0.00
p3.4.d	100.00	100.00	0.00	100.00	100.00	0.01	100.00	100.00	0.02
p3.4.e	140.00	140.00	0.01	140.00	140.00	0.01	140.00	140.00	0.01
p3.4.f	190.00	190.00	0.02	190.00	190.00	0.02	190.00	190.00	0.03
p3.4.g	220.00	220.00	0.03	220.00	220.00	0.03	220.00	220.00	0.03
p3.4.h	240.00	240.00	0.11	240.00	240.00	0.16	240.00	240.00	0.23
p3.4.i	270.00	270.00	0.18	270.00	270.00	0.15	270.00	270.00	0.16
p3.4.j	310.00	310.00	0.42	310.00	310.00	1.04	310.00	310.00	0.89
p3.4.k	350.00	350.00	0.27	350.00	350.00	0.28	350.00	350.00	0.37
p3.4.l	380.00	380.00	0.40	380.00	380.00	6.42	380.00	380.00	0.20
p3.4.m	390.00	390.00	606.88	390.00	390.00	404.88	390.00	390.00	402.78
p3.4.n	440.00	440.00	48.98	440.00	440.00	65.25	440.00	440.00	61.12
p3.4.o	500.00	500.00	24.45	500.00	500.00	17.11	500.00	500.00	53.28
p3.4.p	560.00	560.00	2.89	560.00	560.00	50.88	560.00	560.00	18.30
p3.4.q	560.00	560.00	3703.06	560.00	560.00	1780.26	560.00	560.00	2698.82
p3.4.r	600.00	600.00	37.68	600.00	600.00	45.36	600.00	600.00	23.79
p3.4.s	670.00	670.00	1.72	670.00	670.00	37.26	670.00	670.00	4.34
p3.4.t	670.00	670.00	23.21	670.00	670.00	7.41	670.00	670.00	4.48
p4.2.a	206.00	206.00	0.15	206.00	206.00	0.22	206.00	206.00	0.38
p4.2.b	341.00	341.00	1.53	341.00	341.00	1.48	341.00	341.00	2.53
p4.2.c	452.00	452.00	2.85	452.00	452.00	2.35	452.00	452.00	4.43
p4.2.d	531.00	531.00	66.63	531.00	531.00	36.39	531.00	531.00	42.87
p4.2.e	618.00	618.00	39.41	618.00	618.00	28.44	618.00	618.00	56.86
p4.2.f	687.00	687.00	1407.29	687.00	687.00	303.13	687.00	687.00	316.83
p4.2.g	757.00	757.00	3347.87	757.00	757.00	375.21	757.00	757.00	218.06
p4.2.h	835.00	835.00	2812.21	835.00	835.00	563.81	835.00	835.00	101.72
p4.2.i	918.00	918.00	1831.07	918.00	918.00	321.89	918.00	918.00	69.21
p4.2.j	962.00	977.71	7200.00	965.00	965.00	496.17	965.00	965.00	358.04
p4.2.k	1022.00	1048.77	7200.00	1022.00	1022.00	1522.30	1022.00	1022.00	391.10
p4.2.l	1072.00	1114.09	7200.00	1074.00	1074.00	2542.10	1074.00	1074.00	748.97
p4.2.m	1130.00	1161.46	7200.00	1132.00	1132.00	2295.54	1132.00	1132.00	155.79
p4.2.n	1172.00	1205.34	7200.00	1174.00	1174.00	1576.98	1174.00	1174.00	609.13
p4.2.o	1217.00	1235.06	7200.00	1218.00	1218.00	663.38	1218.00	1218.00	752.99
p4.2.p	1241.00	1270.44	7200.00	1242.00	1242.00	5192.14	1242.00	1242.00	2138.22
p4.2.q	1258.00	1283.10	7200.00	1265.00	1273.22	7200.00	1268.00	1268.00	5924.10
p4.2.r	1285.00	1298.59	7200.00	1292.00	1292.00	3328.89	1292.00	1292.00	1586.48
p4.2.s	1304.00	1306.00	7200.00	1300.00	1306.00	7200.00	1299.00	1305.72	7200.00
p4.2.t	1306.00	1306.00	189.76	1306.00	1306.00	3882.98	1306.00	1306.00	6800.75

Continued on next page

Table A.6 — continued from previous page.

	B-B&C			CPA_1			CPA_2		
Instance	LB	UB	Time (s)	LB	UB	Time (s)	LB	UB	Time (s)
p4.3.a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
p4.3.b	38.00	38.00	0.00	38.00	38.00	0.01	38.00	38.00	0.01
p4.3.c	193.00	193.00	0.10	193.00	193.00	0.16	193.00	193.00	0.17
p4.3.d	335.00	335.00	1.42	335.00	335.00	10.33	335.00	335.00	10.11
p4.3.e	468.00	468.00	12.87	468.00	468.00	18.02	468.00	468.00	15.24
p4.3.f	579.00	579.00	3.92	579.00	579.00	19.64	579.00	579.00	17.07
p4.3.g	653.00	653.00	95.22	653.00	653.00	181.03	653.00	653.00	261.50
p4.3.h	729.00	729.00	811.58	729.00	729.00	4764.03	729.00	729.00	592.34
p4.3.i	809.00	809.00	643.36	809.00	809.00	534.24	809.00	809.00	636.13
p4.3.j	861.00	861.00	6997.81	861.00	863.39	7200.00	861.00	866.48	7200.00
p4.3.k	919.00	948.80	7200.00	919.00	935.32	7200.00	919.00	930.52	7200.00
p4.3.l	966.00	1022.34	7200.00	970.00	1013.68	7200.00	978.00	1018.70	7200.00
p4.3.m	1063.00	1080.13	7200.00	1063.00	1063.00	4553.93	1063.00	1074.99	7200.00
p4.3.n	1121.00	1139.22	7200.00	1121.00	1121.00	3400.28	1121.00	1121.00	3005.56
p4.3.o	1172.00	1196.69	7200.00	1172.00	1172.00	5322.10	1170.00	1176.50	7200.00
p4.3.p	1222.00	1234.29	7200.00	1222.00	1222.00	2436.57	1222.00	1222.00	4985.21
p4.3.q	1251.00	1272.24	7200.00	1245.00	1262.44	7200.00	1253.00	1258.35	7200.00
p4.3.r	1266.00	1290.13	7200.00	1257.00	1289.60	7200.00	1271.00	1286.83	7200.00
p4.3.s	1295.00	1302.20	7200.00	1295.00	1299.89	7200.00	1295.00	1298.77	7200.00
p4.3.t	1301.00	1306.00	7200.00	1298.00	1306.00	7200.00	1299.00	1306.00	7200.00
p4.4.a	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
p4.4.b	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
p4.4.c	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
p4.4.d	38.00	38.00	0.00	38.00	38.00	0.01	38.00	38.00	0.01
p4.4.e	183.00	183.00	0.01	183.00	183.00	0.02	183.00	183.00	0.02
p4.4.f	324.00	324.00	0.24	324.00	324.00	3.08	324.00	324.00	0.88
p4.4.g	461.00	461.00	4.47	461.00	461.00	20.81	461.00	461.00	18.43
p4.4.h	571.00	571.00	33.96	571.00	571.00	77.72	571.00	571.00	102.37
p4.4.i	657.00	657.00	338.82	657.00	657.00	358.45	657.00	657.00	629.99
p4.4.j	732.00	732.00	6987.50	732.00	732.00	6298.35	732.00	732.00	7130.64
p4.4.k	821.00	821.00	1473.53	821.00	821.00	1913.23	821.00	821.00	3445.36
p4.4.l	880.00	880.00	4445.93	880.00	880.00	2273.87	880.00	880.00	3055.40
p4.4.m	918.00	946.43	7200.00	912.00	943.51	7200.00	915.00	943.36	7200.00
p4.4.n	972.00	1016.84	7200.00	972.00	1012.45	7200.00	961.00	1013.73	7200.00
p4.4.o	1037.00	1090.84	7200.00	1061.00	1089.72	7200.00	1050.00	1094.17	7200.00
p4.4.p	1124.00	1145.92	7200.00	1064.00	1148.61	7200.00	1124.00	1152.78	7200.00
p4.4.q	1158.00	1215.85	7200.00	1113.00	1216.72	7200.00	1149.00	1208.78	7200.00
p4.4.r	1210.00	1259.41	7200.00	1172.00	1254.08	7200.00	1197.00	1253.07	7200.00
p4.4.s	1191.00	1288.28	7200.00	1256.00	1281.49	7200.00	1259.00	1279.99	7200.00
p4.4.t	1273.00	1304.05	7200.00	1240.00	1302.00	7200.00	1245.00	1299.10	7200.00
p5.2.a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p5.2.b	20.00	20.00	0.01	20.00	20.00	0.01	20.00	20.00	0.00
p5.2.c	50.00	50.00	0.01	50.00	50.00	0.01	50.00	50.00	0.03
p5.2.d	80.00	80.00	0.27	80.00	80.00	0.41	80.00	80.00	0.45
p5.2.e	180.00	180.00	0.03	180.00	180.00	0.11	180.00	180.00	0.43
p5.2.f	240.00	240.00	0.19	240.00	240.00	1.03	240.00	240.00	0.62
p5.2.g	320.00	320.00	2.97	320.00	320.00	8.43	320.00	320.00	15.32
p5.2.h	410.00	410.00	4.09	410.00	410.00	18.86	410.00	410.00	12.29
p5.2.i	480.00	480.00	107.34	480.00	480.00	80.11	480.00	480.00	81.68
p5.2.j	580.00	580.00	26.56	580.00	580.00	52.42	580.00	580.00	78.04
p5.2.k	670.00	670.00	184.95	670.00	670.00	92.04	670.00	670.00	151.27
p5.2.l	800.00	800.00	1.00	800.00	800.00	1.77	800.00	800.00	2.06
p5.2.m	860.00	860.00	38.27	860.00	860.00	12.57	860.00	860.00	14.36
p5.2.n	925.00	925.00	490.18	925.00	925.00	188.87	925.00	925.00	209.98
p5.2.o	1020.00	1020.00	277.15	1020.00	1020.00	74.67	1020.00	1020.00	73.91

Continued on next page

Table A.6 — continued from previous page.

	B-B&C			CPA_1			CPA_2		
Instance	LB	UB	Time (s)	LB	UB	Time (s)	LB	UB	Time (s)
p5.2.p	1150.00	1150.00	0.92	1150.00	1150.00	1.16	1150.00	1150.00	1.45
p5.2.q	1195.00	1195.00	21.69	1195.00	1195.00	17.90	1195.00	1195.00	14.27
p5.2.r	1260.00	1260.00	116.40	1260.00	1260.00	42.42	1260.00	1260.00	62.75
p5.2.s	1340.00	1340.00	0.86	1340.00	1340.00	2.49	1340.00	1340.00	2.57
p5.2.t	1400.00	1400.00	34.20	1400.00	1400.00	6.60	1400.00	1400.00	6.61
p5.2.u	1460.00	1460.00	11.11	1460.00	1460.00	5.81	1460.00	1460.00	6.03
p5.2.v	1505.00	1505.00	5937.96	1505.00	1505.00	609.34	1505.00	1505.00	570.04
p5.2.w	1565.00	1565.00	1210.50	1565.00	1565.00	103.07	1565.00	1565.00	368.40
p5.2.x	1610.00	1610.00	268.72	1610.00	1610.00	37.74	1610.00	1610.00	6.67
p5.2.y	1645.00	1645.00	264.90	1645.00	1645.00	86.92	1645.00	1645.00	41.54
p5.2.z	1680.00	1680.00	48.06	1680.00	1680.00	6.54	1680.00	1680.00	45.76
p5.3.a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p5.3.b	15.00	15.00	0.00	15.00	15.00	0.00	15.00	15.00	0.00
p5.3.c	20.00	20.00	0.01	20.00	20.00	0.00	20.00	20.00	0.00
p5.3.d	60.00	60.00	0.01	60.00	60.00	0.01	60.00	60.00	0.01
p5.3.e	95.00	95.00	0.03	95.00	95.00	0.03	95.00	95.00	0.02
p5.3.f	110.00	110.00	2.15	110.00	110.00	9.06	110.00	110.00	7.40
p5.3.g	185.00	185.00	0.25	185.00	185.00	0.44	185.00	185.00	0.52
p5.3.h	260.00	260.00	2.08	260.00	260.00	8.09	260.00	260.00	11.25
p5.3.i	335.00	335.00	1.75	335.00	335.00	3.99	335.00	335.00	4.98
p5.3.j	470.00	470.00	0.36	470.00	470.00	1.15	470.00	470.00	4.23
p5.3.k	495.00	495.00	1000.88	495.00	495.00	3678.80	495.00	495.00	4505.21
p5.3.l	595.00	595.00	626.80	595.00	595.00	546.37	595.00	595.00	538.07
p5.3.m	650.00	686.69	7200.00	650.00	689.83	7200.00	650.00	696.63	7200.00
p5.3.n	755.00	755.00	2179.52	755.00	755.00	6162.82	755.00	766.20	7200.00
p5.3.o	870.00	870.00	386.17	870.00	870.00	285.57	870.00	870.00	178.54
p5.3.p	990.00	990.00	9.72	990.00	990.00	24.82	990.00	990.00	37.58
p5.3.q	1070.00	1070.00	42.31	1070.00	1070.00	53.80	1070.00	1070.00	53.88
p5.3.r	1125.00	1134.14	7200.00	1125.00	1125.00	2264.09	1125.00	1133.86	7200.00
p5.3.s	1185.00	1239.32	7200.00	1190.00	1229.46	7200.00	1190.00	1228.12	7200.00
p5.3.t	1260.00	1313.84	7200.00	1260.00	1306.55	7200.00	1260.00	1306.70	7200.00
p5.3.u	1345.00	1390.85	7200.00	1345.00	1378.18	7200.00	1345.00	1375.11	7200.00
p5.3.v	1425.00	1452.80	7200.00	1425.00	1438.03	7200.00	1425.00	1435.22	7200.00
p5.3.w	1485.00	1513.21	7200.00	1485.00	1499.74	7200.00	1465.00	1506.47	7200.00
p5.3.x	1555.00	1578.62	7200.00	1555.00	1555.00	3833.30	1555.00	1555.00	4355.18
p5.3.y	1595.00	1625.99	7200.00	1590.00	1623.06	7200.00	1595.00	1620.19	7200.00
p5.3.z	1635.00	1665.32	7200.00	1635.00	1659.11	7200.00	1635.00	1660.01	7200.00
p5.4.a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p5.4.b	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p5.4.c	20.00	20.00	0.00	20.00	20.00	0.00	20.00	20.00	0.00
p5.4.d	20.00	20.00	0.00	20.00	20.00	0.01	20.00	20.00	0.00
p5.4.e	20.00	20.00	0.00	20.00	20.00	0.01	20.00	20.00	0.00
p5.4.f	80.00	80.00	0.01	80.00	80.00	0.09	80.00	80.00	0.01
p5.4.g	140.00	140.00	0.02	140.00	140.00	0.03	140.00	140.00	0.03
p5.4.h	140.00	140.00	5.49	140.00	140.00	16.78	140.00	140.00	7.43
p5.4.i	240.00	240.00	0.05	240.00	240.00	0.06	240.00	240.00	0.08
p5.4.j	340.00	340.00	0.06	340.00	340.00	0.15	340.00	340.00	0.29
p5.4.k	340.00	340.00	459.14	340.00	340.00	799.44	340.00	340.00	2960.84
p5.4.l	430.00	430.00	12.10	430.00	430.00	43.02	430.00	430.00	52.56
p5.4.m	555.00	555.00	7.72	555.00	555.00	14.96	555.00	555.00	14.69
p5.4.n	620.00	620.00	534.94	620.00	620.00	3037.07	620.00	620.00	3215.61
p5.4.o	690.00	709.19	7200.00	690.00	718.01	7200.00	690.00	728.81	7200.00
p5.4.p	760.00	813.20	7200.00	760.00	820.95	7200.00	760.00	815.17	7200.00
p5.4.q	860.00	885.35	7200.00	860.00	890.01	7200.00	860.00	892.49	7200.00
p5.4.r	960.00	960.00	2443.39	960.00	960.00	1838.42	960.00	960.00	5689.60
p5.4.s	1030.00	1071.78	7200.00	1025.00	1075.95	7200.00	1030.00	1077.02	7200.00

Continued on next page

Table A.6 — continued from previous page.

	B-B&C			CPA_1			CPA_2		
Instance	LB	UB	Time (s)	LB	UB	Time (s)	LB	UB	Time (s)
p5.4.t	1160.00	1160.00	682.75	1160.00	1160.00	183.36	1160.00	1160.00	366.48
p5.4.u	1300.00	1300.00	1.30	1300.00	1300.00	2.25	1300.00	1300.00	4.10
p5.4.v	1320.00	1351.02	7200.00	1320.00	1349.62	7200.00	1320.00	1354.78	7200.00
p5.4.w	1390.00	1429.19	7200.00	1390.00	1417.86	7200.00	1385.00	1420.57	7200.00
p5.4.x	1450.00	1494.34	7200.00	1450.00	1488.33	7200.00	1445.00	1489.65	7200.00
p5.4.y	1520.00	1553.50	7200.00	1520.00	1545.77	7200.00	1520.00	1543.66	7200.00
p5.4.z	1620.00	1620.00	47.49	1620.00	1620.00	258.32	1620.00	1620.00	270.84
p6.2.a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.2.b	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.2.c	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.2.d	192.00	192.00	0.16	192.00	192.00	0.27	192.00	192.00	0.26
p6.2.e	360.00	360.00	1.81	360.00	360.00	3.52	360.00	360.00	2.59
p6.2.f	588.00	588.00	0.30	588.00	588.00	0.45	588.00	588.00	0.58
p6.2.g	660.00	660.00	43.42	660.00	660.00	70.86	660.00	660.00	79.02
p6.2.h	780.00	780.00	11.89	780.00	780.00	18.09	780.00	780.00	12.74
p6.2.i	888.00	888.00	1.43	888.00	888.00	6.08	888.00	888.00	4.57
p6.2.j	948.00	966.19	7200.00	948.00	948.00	1266.70	948.00	948.00	1190.73
p6.2.k	1032.00	1058.97	7200.00	1032.00	1032.00	1715.31	1032.00	1032.00	2093.70
p6.2.l	1116.00	1116.00	424.97	1116.00	1116.00	229.80	1116.00	1116.00	132.97
p6.2.m	1188.00	1188.00	3108.01	1188.00	1188.00	71.25	1188.00	1188.00	38.70
p6.2.n	1260.00	1260.00	1.14	1260.00	1260.00	1.53	1260.00	1260.00	1.90
p6.3.a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.3.b	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.3.c	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.3.d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.3.e	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.3.f	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.3.g	282.00	282.00	0.29	282.00	282.00	0.96	282.00	282.00	0.95
p6.3.h	444.00	444.00	59.19	444.00	444.00	69.66	444.00	444.00	63.49
p6.3.i	642.00	642.00	0.80	642.00	642.00	2.11	642.00	642.00	2.04
p6.3.j	828.00	828.00	0.91	828.00	828.00	1.36	828.00	828.00	2.05
p6.3.k	894.00	894.00	1051.25	894.00	894.00	1704.91	894.00	894.00	532.27
p6.3.l	1002.00	1002.00	326.48	1002.00	1002.00	107.48	1002.00	1002.00	102.28
p6.3.m	1080.00	1095.53	7200.00	1080.00	1080.00	5010.76	1080.00	1099.83	7200.00
p6.3.n	1170.00	1170.00	1885.01	1170.00	1170.00	609.09	1170.00	1170.00	482.82
p6.4.a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.4.b	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.4.c	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.4.d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.4.e	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.4.f	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.4.g	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.4.h	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.4.i	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p6.4.j	366.00	366.00	0.40	366.00	366.00	1.96	366.00	366.00	1.89
p6.4.k	528.00	528.00	217.08	528.00	528.00	24.05	528.00	528.00	19.61
p6.4.l	696.00	696.00	34.92	696.00	696.00	94.21	696.00	696.00	90.52
p6.4.m	912.00	912.00	3.11	912.00	912.00	12.65	912.00	912.00	9.29
p6.4.n	1068.00	1068.00	1.63	1068.00	1068.00	5.34	1068.00	1068.00	4.20
p7.2.a	30.00	30.00	0.00	30.00	30.00	0.01	30.00	30.00	0.00
p7.2.b	64.00	64.00	0.00	64.00	64.00	0.01	64.00	64.00	0.01
p7.2.c	101.00	101.00	0.06	101.00	101.00	0.13	101.00	101.00	0.13
p7.2.d	190.00	190.00	0.67	190.00	190.00	2.20	190.00	190.00	1.45
p7.2.e	290.00	290.00	6.43	290.00	290.00	10.29	290.00	290.00	13.33

Continued on next page

Table A.6 — continued from previous page.

	B-B&C			CPA_1			CPA_2		
Instance	LB	UB	Time (s)	LB	UB	Time (s)	LB	UB	Time (s)
p7.2.f	387.00	387.00	8.04	387.00	387.00	3.73	387.00	387.00	5.68
p7.2.g	459.00	459.00	20.81	459.00	459.00	11.80	459.00	459.00	18.54
p7.2.h	521.00	521.00	526.83	521.00	521.00	102.22	521.00	521.00	89.05
p7.2.i	580.00	580.00	2753.82	580.00	580.00	145.95	580.00	580.00	173.35
p7.2.j	646.00	653.02	7200.00	646.00	646.00	278.74	646.00	646.00	289.37
p7.2.k	705.00	720.74	7200.00	705.00	705.00	140.54	705.00	705.00	323.09
p7.2.l	767.00	783.05	7200.00	767.00	767.00	85.12	767.00	767.00	454.05
p7.2.m	827.00	857.05	7200.00	827.00	827.00	832.30	827.00	827.00	439.31
p7.2.n	888.00	912.97	7200.00	888.00	888.00	318.79	888.00	888.00	312.76
p7.2.o	945.00	965.56	7200.00	945.00	945.00	920.81	945.00	945.00	292.73
p7.2.p	1002.00	1021.83	7200.00	1002.00	1002.00	2118.39	1002.00	1002.00	431.12
p7.2.q	1043.00	1061.78	7200.00	1044.00	1044.00	2324.04	1044.00	1044.00	1659.69
p7.2.r	1082.00	1108.41	7200.00	1094.00	1094.00	3080.47	1094.00	1094.00	370.02
p7.2.s	1119.00	1151.39	7200.00	1136.00	1136.00	1044.59	1136.00	1136.00	355.02
p7.2.t	1166.00	1187.31	7200.00	1179.00	1179.00	1911.50	1179.00	1179.00	555.45
p7.3.a	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
p7.3.b	46.00	46.00	0.00	46.00	46.00	0.01	46.00	46.00	0.00
p7.3.c	79.00	79.00	0.00	79.00	79.00	0.02	79.00	79.00	0.02
p7.3.d	117.00	117.00	0.03	117.00	117.00	0.05	117.00	117.00	0.05
p7.3.e	175.00	175.00	0.19	175.00	175.00	0.59	175.00	175.00	0.60
p7.3.f	247.00	247.00	1.14	247.00	247.00	6.75	247.00	247.00	4.99
p7.3.g	344.00	344.00	3.88	344.00	344.00	9.83	344.00	344.00	15.66
p7.3.h	425.00	425.00	967.41	425.00	425.00	601.99	425.00	425.00	588.76
p7.3.i	487.00	487.00	3722.25	487.00	487.00	1446.77	487.00	487.00	4788.93
p7.3.j	564.00	564.00	505.07	564.00	564.00	271.76	564.00	564.00	704.18
p7.3.k	633.00	633.00	514.09	633.00	633.00	97.47	633.00	633.00	180.00
p7.3.l	684.00	697.81	7200.00	684.00	687.92	7200.00	684.00	684.00	6812.05
p7.3.m	762.00	762.00	2355.60	762.00	762.00	320.73	762.00	762.00	425.37
p7.3.n	820.00	824.91	7200.00	820.00	820.00	522.20	820.00	820.00	464.41
p7.3.o	859.00	904.70	7200.00	874.00	874.00	1655.47	874.00	874.00	7016.51
p7.3.p	917.00	965.27	7200.00	929.00	929.00	5562.65	929.00	929.00	6001.06
p7.3.q	967.00	1020.14	7200.00	987.00	987.00	3455.92	987.00	987.00	6705.31
p7.3.r	1014.00	1064.63	7200.00	1020.00	1057.35	7200.00	1020.00	1055.82	7200.00
p7.3.s	1074.00	1108.62	7200.00	1081.00	1097.61	7200.00	1081.00	1098.71	7200.00
p7.3.t	1115.00	1150.06	7200.00	1120.00	1142.78	7200.00	1120.00	1142.42	7200.00
p7.4.a	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
p7.4.b	30.00	30.00	0.00	30.00	30.00	0.01	30.00	30.00	0.00
p7.4.c	46.00	46.00	0.00	46.00	46.00	0.01	46.00	46.00	0.00
p7.4.d	79.00	79.00	0.01	79.00	79.00	0.01	79.00	79.00	0.01
p7.4.e	123.00	123.00	0.01	123.00	123.00	0.04	123.00	123.00	0.02
p7.4.f	164.00	164.00	0.08	164.00	164.00	0.16	164.00	164.00	0.18
p7.4.g	217.00	217.00	0.36	217.00	217.00	1.71	217.00	217.00	2.38
p7.4.h	285.00	285.00	6.48	285.00	285.00	29.02	285.00	285.00	40.78
p7.4.i	366.00	366.00	7.81	366.00	366.00	13.44	366.00	366.00	7.86
p7.4.j	462.00	462.00	1042.57	462.00	462.00	869.92	462.00	462.00	219.11
p7.4.k	520.00	529.53	7200.00	520.00	529.96	7200.00	520.00	532.38	7200.00
p7.4.l	590.00	597.62	7200.00	590.00	596.26	7200.00	590.00	600.64	7200.00
p7.4.m	646.00	679.16	7200.00	646.00	664.78	7200.00	646.00	670.28	7200.00
p7.4.n	730.00	730.00	1855.22	730.00	730.00	1229.16	730.00	730.00	311.97
p7.4.o	781.00	795.56	7200.00	781.00	793.20	7200.00	781.00	794.54	7200.00
p7.4.p	846.00	870.45	7200.00	846.00	862.54	7200.00	846.00	865.81	7200.00
p7.4.q	909.00	939.78	7200.00	909.00	929.80	7200.00	909.00	929.44	7200.00
p7.4.r	970.00	995.32	7200.00	970.00	994.58	7200.00	970.00	990.16	7200.00
p7.4.s	1022.00	1049.24	7200.00	1022.00	1041.82	7200.00	1022.00	1042.78	7200.00
p7.4.t	1077.00	1096.33	7200.00	1077.00	1096.39	7200.00	1077.00	1092.04	7200.00

Table A.3: Gaps referred to the bounds available at the root nodes of the branch-and-bound trees of B-B&C and CPA at solving the original benchmark of TOP instances. Here, the CPLEX built-in cuts separated at the root nodes are considered.

B-B&C						
	Root gap		Root *primal* gap		Root *dual* gap	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1	6.14	7.19	2.91	4.15	3.23	3.67
2	1.44	4.24	0.85	3.30	0.59	2.00
3	7.37	6.70	3.12	3.99	4.26	4.00
4	14.73	10.54	10.08	9.53	4.65	2.75
5	8.77	9.68	4.45	7.52	4.32	4.64
6	3.29	4.10	0.75	1.84	2.54	2.96
7	14.67	12.33	8.91	10.64	5.76	3.36
Total	8.81	9.81	4.88	7.72	3.92	3.82

CPA_1						
	Root gap		Root *primal* gap		Root *dual* gap	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1	2.76	4.75	1.14	2.75	1.62	2.81
2	1.60	4.18	1.00	2.90	0.59	2.00
3	6.81	6.34	3.34	4.30	3.47	3.67
4	13.37	7.84	9.51	6.82	3.86	2.58
5	9.02	9.02	4.44	6.99	4.58	5.00
6	3.51	4.71	0.98	2.35	2.53	3.04
7	11.20	10.05	7.64	9.32	3.56	2.30
Total	7.59	8.36	4.42	6.70	3.16	3.57

CPA_2						
	Root gap		Root *primal* gap		Root *dual* gap	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1	2.28	4.61	1.00	2.83	1.28	2.51
2	2.36	4.85	1.77	3.92	0.59	2.00
3	7.18	6.52	3.99	4.76	3.19	3.12
4	10.83	7.24	7.77	6.26	3.06	2.41
5	8.76	9.58	4.69	7.55	4.08	4.60
6	3.48	4.68	0.87	2.25	2.61	3.06
7	8.46	7.73	5.36	5.99	3.09	2.42
Total	6.77	7.66	3.98	5.88	2.78	3.29

Table A.4: Gaps referred to the bounds available at the root nodes of the branch-and-bound trees of CPLEX while directly solving formulations $\mathcal{F}_1$ and $\mathcal{F}_2$. Results for the original benchmark of TOP instances. Here, the CPLEX built-in cuts are enabled.

$\mathcal{F}_1$						
	Root gap		Root *primal* gap		Root *dual* gap	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1	7.23	8.60	3.75	5.84	3.48	3.81
2	2.96	7.19	2.37	6.82	0.59	2.00
3	8.01	6.86	3.70	4.38	4.31	4.05
4	15.89	10.54	10.87	9.84	5.03	3.08
5	9.11	9.93	4.76	7.72	4.35	4.67
6	3.29	4.10	0.75	1.84	2.54	2.96
7	15.56	12.35	9.39	10.52	6.17	3.64
Total	9.57	10.22	5.48	8.16	4.09	3.97

$\mathcal{F}_2$						
	Root gap		Root *primal* gap		Root *dual* gap	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1	6.62	7.35	2.94	4.28	3.68	4.05
2	1.97	4.32	1.34	3.72	0.63	2.09
3	8.68	7.36	4.06	4.79	4.62	4.20
4	16.99	10.54	11.91	10.03	5.07	3.03
5	11.93	11.28	7.28	9.39	4.65	5.03
6	3.86	5.37	1.36	3.61	2.50	2.94
7	16.77	12.97	10.44	11.46	6.33	3.80
Total	10.50	10.73	6.23	8.74	4.26	4.15

Table A.5: Extra information on the execution of B-B&C and CPA at solving the original benchmark of TOP instances.

B-B&C				
	Final gap			
Set	Avg (%)	StDev (%)	Nodes (#)	GCCs (#)
1	0.00	0.00	709.28	37.65
2	0.00	0.00	61.45	0.79
3	0.00	0.00	53 325.30	176.53
4	1.21	1.73	188 373.25	1081.75
5	0.68	1.42	528 237.04	868.40
6	0.14	0.52	271 593.05	272.17
7	1.27	1.61	356 152.85	684.77
Total	0.54	1.25	228 735.94	511.13

CPA_1						
	Final gap			User cuts (#)		
Set	Avg (%)	StDev (%)	Nodes (#)	GCCs	CCs	LCIs
1	0.00	0.00	941.07	8.94	27.76	2.00
2	0.00	0.00	290.48	0.73	1.24	0.91
3	0.00	0.00	42 045.15	6.63	21.80	3.53
4	0.85	1.92	269 974.58	27.12	111.00	0.65
5	0.62	1.45	936 014.97	2.38	29.12	1.00
6	0.00	0.00	352 712.33	0.55	2.76	0.55
7	0.42	0.87	334 659.02	18.52	331.18	0.55
Total	0.32	1.10	327 349.39	9.95	82.08	1.35

CPA_2					
	Final gap			User cuts (#)	
Set	Avg (%)	StDev (%)	Nodes (#)	LCIs	CCCs
1	0.00	0.00	907.72	2.11	19.15
2	0.00	0.00	189.85	0.85	1.58
3	0.00	0.00	57 883.95	4.57	16.03
4	0.67	1.41	298 083.80	1.22	98.45
5	0.69	1.53	949 125.00	1.64	23.79
6	0.04	0.28	316 775.64	0.52	0.79
7	0.44	0.93	275 486.58	0.85	104.37
Total	0.32	1.01	323 718.01	1.78	41.62

A.4 Detailed results for the exact algorithms at solving the new STOP instances

In Tables A.7 and A.8, we display some information on the bounds available at the root nodes of the branch-and-bound trees of B-B&C and CPA while solving the original benchmark of TOP instances. In Table A.7, for each algorithm and instance set, we give the averages and the corresponding standard deviations of three gap values. The first one is the relative optimality gap, given by $\frac{UB_r - LB_r}{UB_r}$, where LB_r and UB_r are, respectively, the lower and upper bounds at the root node. The second and third gaps, namely *primal* and *dual* gaps, are given by $\frac{LB^* - LB_r}{UB_r}$ and $\frac{UB_r - LB^*}{UB_r}$, respectively. Here, LB^* corresponds to the best lower bound obtained by B-B&C or CPA for a given instance within two hours of execution. Notice that, for every instance, the summation of the corresponding primal and dual gaps gives the root relative optimality gap. Then, these last two gaps are given as a way to estimate the contribution of the primal and the dual bounds to the actual optimality gap. All the bounds considered in Table A.7 also take into account the CPLEX built-in cuts separated at the root nodes. In Table A.8, we give the same information displayed in Table A.7, but for formulations $\mathcal{F}_1$ and $\mathcal{F}_2$ without the addition of the inequalities proposed in section 2.2. Recall that $\mathcal{F}_1$ and $\mathcal{F}_2$ are the formulations solved within B-B&C and CPA, respectively. Also in this case, the CPLEX built-in cuts are enabled.

In Table A.9, we display some extra information concerning the execution of B-B&C and CPA at solving the new STOP instances. For each algorithm and instance set, we first display the average and the standard deviation of the relative optimality gaps of the solutions obtained within two hours of execution. Here, we consider the results for the complete benchmark, and not only the instances not solved to optimality (as in Table 3.9). In Table A.9, we also expose the average number of nodes explored in the branch-and-bound tree, as well as the average number of cuts separated for each class of inequalities discussed in section 2.2. In the results for CPA_2, we omitted the average number of AVICs, as they are separated by complete enumeration and the number of these cuts in each instance always corresponds to $2 \times |E|$.

In Table A.10, we display a per-instance report of the results obtained by our implementations of B-B&C and CPA at solving the new benchmark of STOP instances. For all algorithms, we report, for each instance, the best lower and upper bounds obtained within two hours of execution (columns "LB" and "UB", respectively) and the wall-clock execution time in seconds. For the cases where neither a feasible solution was found nor the infeasibility was proven, the entries "LB" and "UB" were filled with "-inf" and "inf", respectively. We filled with dashes the entries related to the instances that were proven to be infeasible within the time limit of two hours.

Table A.10: Detailed results for the exact algorithms at solving the new STOP instances.

	B-B&C			CPA_1			CPA_2		
Instance	LB	UB	Time (s)	LB	UB	Time (s)	LB	UB	Time (s)
p1.2.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.2.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.2.c_5%	–	–	0.00	–	–	0.00	–	–	0.00

Continued on next page

Table A.10 — continued from previous page.

	B-B&C			CPA_1			CPA_2		
Instance	LB	UB	Time (s)	LB	UB	Time (s)	LB	UB	Time (s)
p1.2.d_5%	15.00	15.00	0.00	15.00	15.00	0.00	15.00	15.00	0.01
p1.2.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.2.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.2.g_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.2.h_5%	45.00	45.00	0.29	45.00	45.00	0.81	45.00	45.00	0.58
p1.2.i_5%	110.00	110.00	1.38	110.00	110.00	2.21	110.00	110.00	3.67
p1.2.j_5%	140.00	140.00	0.52	140.00	140.00	1.11	140.00	140.00	0.91
p1.2.k_5%	150.00	150.00	2.08	150.00	150.00	2.86	150.00	150.00	2.69
p1.2.l_5%	185.00	185.00	1.65	185.00	185.00	4.44	185.00	185.00	2.44
p1.2.m_5%	190.00	190.00	0.61	190.00	190.00	1.02	190.00	190.00	1.19
p1.2.n_5%	215.00	215.00	1.02	215.00	215.00	1.17	215.00	215.00	0.98
p1.2.o_5%	220.00	220.00	1.30	220.00	220.00	0.81	220.00	220.00	1.21
p1.2.p_5%	225.00	225.00	1.15	225.00	225.00	2.33	225.00	225.00	3.25
p1.2.q_5%	245.00	245.00	0.57	245.00	245.00	1.48	245.00	245.00	2.04
p1.2.r_5%	265.00	265.00	3.24	265.00	265.00	1.58	265.00	265.00	5.18
p1.3.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.3.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.3.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.3.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.3.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.3.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.3.g_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.3.h_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.3.i_5%	90.00	90.00	0.04	90.00	90.00	0.10	90.00	90.00	0.16
p1.3.j_5%	80.00	80.00	0.24	80.00	80.00	0.75	80.00	80.00	2.09
p1.3.k_5%	115.00	115.00	0.44	115.00	115.00	2.48	115.00	115.00	2.92
p1.3.l_5%	140.00	140.00	0.68	140.00	140.00	3.67	140.00	140.00	3.45
p1.3.m_5%	130.00	130.00	26.14	130.00	130.00	19.13	130.00	130.00	44.96
p1.3.n_5%	165.00	165.00	3.57	165.00	165.00	8.13	165.00	165.00	8.07
p1.3.o_5%	180.00	180.00	1.66	180.00	180.00	4.30	180.00	180.00	5.80
p1.3.p_5%	200.00	200.00	0.77	200.00	200.00	0.97	200.00	200.00	2.47
p1.3.q_5%	210.00	210.00	2.89	210.00	210.00	9.27	210.00	210.00	9.87
p1.3.r_5%	225.00	225.00	2.24	225.00	225.00	6.26	225.00	225.00	6.03
p1.4.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.4.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.4.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.4.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.4.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.4.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.4.g_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.4.h_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.4.i_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.4.j_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.4.k_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.4.l_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.4.m_5%	115.00	115.00	0.72	115.00	115.00	2.17	115.00	115.00	3.15
p1.4.n_5%	135.00	135.00	0.42	135.00	135.00	3.02	135.00	135.00	2.21
p1.4.o_5%	–	–	0.00	–	–	0.00	–	–	0.00
p1.4.p_5%	150.00	150.00	0.49	150.00	150.00	2.29	150.00	150.00	4.40
p1.4.q_5%	165.00	165.00	1.00	165.00	165.00	2.77	165.00	165.00	4.27
p1.4.r_5%	195.00	195.00	0.77	195.00	195.00	2.38	195.00	195.00	3.80
p2.2.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.2.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.2.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.2.d_5%	–	–	0.00	–	–	0.00	–	–	0.00

Continued on next page

Table A.10 — continued from previous page.

	B-B&C			CPA_1			CPA_2		
Instance	LB	UB	Time (s)	LB	UB	Time (s)	LB	UB	Time (s)
p2.2.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.2.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.2.g_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.2.h_5%	200.00	200.00	0.23	200.00	200.00	0.26	200.00	200.00	0.14
p2.2.i_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.2.j_5%	220.00	220.00	0.20	220.00	220.00	0.31	220.00	220.00	0.22
p2.2.k_5%	210.00	210.00	0.04	210.00	210.00	0.11	210.00	210.00	0.39
p2.3.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.3.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.3.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.3.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.3.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.3.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.3.g_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.3.h_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.3.i_5%	170.00	170.00	0.15	170.00	170.00	0.12	170.00	170.00	0.12
p2.3.j_5%	160.00	160.00	0.15	160.00	160.00	0.07	160.00	160.00	0.07
p2.3.k_5%	170.00	170.00	0.05	170.00	170.00	0.04	170.00	170.00	0.05
p2.4.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.4.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.4.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.4.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.4.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.4.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.4.g_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.4.h_5%	95.00	95.00	0.01	95.00	95.00	0.01	95.00	95.00	0.00
p2.4.i_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.4.j_5%	–	–	0.00	–	–	0.00	–	–	0.00
p2.4.k_5%	145.00	145.00	0.02	145.00	145.00	0.06	145.00	145.00	0.05
p3.2.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.2.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.2.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.2.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.2.e_5%	220.00	220.00	0.68	220.00	220.00	1.82	220.00	220.00	2.14
p3.2.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.2.g_5%	290.00	290.00	1.36	290.00	290.00	4.14	290.00	290.00	6.13
p3.2.h_5%	360.00	360.00	1.40	360.00	360.00	4.42	360.00	360.00	4.89
p3.2.i_5%	400.00	400.00	3.19	400.00	400.00	11.09	400.00	400.00	9.75
p3.2.j_5%	470.00	470.00	9.47	470.00	470.00	20.00	470.00	470.00	12.72
p3.2.k_5%	510.00	510.00	15.22	510.00	510.00	14.02	510.00	510.00	23.63
p3.2.l_5%	540.00	540.00	3.59	540.00	540.00	4.84	540.00	540.00	4.80
p3.2.m_5%	560.00	560.00	4.51	560.00	560.00	4.38	560.00	560.00	6.45
p3.2.n_5%	610.00	610.00	1.64	610.00	610.00	5.67	610.00	610.00	6.75
p3.2.o_5%	660.00	660.00	2.39	660.00	660.00	5.62	660.00	660.00	5.60
p3.2.p_5%	690.00	690.00	5.50	690.00	690.00	5.52	690.00	690.00	7.38
p3.2.q_5%	710.00	710.00	5.65	710.00	710.00	7.94	710.00	710.00	3.16
p3.2.r_5%	740.00	740.00	12.26	740.00	740.00	8.03	740.00	740.00	4.45
p3.2.s_5%	730.00	730.00	13.98	730.00	730.00	8.15	730.00	730.00	3.81
p3.2.t_5%	700.00	700.00	1.76	700.00	700.00	0.66	700.00	700.00	0.66
p3.3.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.3.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.3.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.3.d_5%	150.00	150.00	0.02	150.00	150.00	0.03	150.00	150.00	0.04
p3.3.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.3.f_5%	160.00	160.00	0.17	160.00	160.00	0.38	160.00	160.00	0.43

Continued on next page

Table A.10 — continued from previous page.

	B-B&C			CPA_1			CPA_2		
Instance	LB	UB	Time (s)	LB	UB	Time (s)	LB	UB	Time (s)
p3.3.g_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.3.h_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.3.i_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.3.j_5%	360.00	360.00	11.77	360.00	360.00	18.13	360.00	360.00	18.19
p3.3.k_5%	400.00	400.00	32.86	400.00	400.00	14.90	400.00	400.00	37.33
p3.3.l_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.3.m_5%	470.00	470.00	31.19	470.00	470.00	43.21	470.00	470.00	60.92
p3.3.n_5%	500.00	500.00	4.49	500.00	500.00	14.68	500.00	500.00	15.48
p3.3.o_5%	560.00	560.00	664.38	560.00	560.00	446.60	560.00	560.00	425.11
p3.3.p_5%	570.00	570.00	966.90	570.00	570.00	1934.16	570.00	570.00	2361.59
p3.3.q_5%	650.00	650.00	101.12	650.00	650.00	14.13	650.00	650.00	49.10
p3.3.r_5%	660.00	660.00	0.48	660.00	660.00	1.68	660.00	660.00	77.51
p3.3.s_5%	670.00	670.00	39.79	670.00	670.00	845.76	670.00	670.00	20.99
p3.3.t_5%	700.00	700.00	643.37	700.00	700.00	109.41	700.00	700.00	99.74
p3.4.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.4.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.4.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.4.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.4.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.4.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.4.g_5%	190.00	190.00	0.04	190.00	190.00	0.02	190.00	190.00	0.03
p3.4.h_5%	220.00	220.00	0.09	220.00	220.00	0.16	220.00	220.00	0.22
p3.4.i_5%	230.00	230.00	0.22	230.00	230.00	0.52	230.00	230.00	0.27
p3.4.j_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.4.k_5%	280.00	280.00	0.24	280.00	280.00	0.47	280.00	280.00	0.48
p3.4.l_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.4.m_5%	340.00	340.00	3075.14	340.00	340.00	1752.61	340.00	340.00	681.31
p3.4.n_5%	380.00	380.00	7.85	380.00	380.00	46.07	380.00	380.00	73.50
p3.4.o_5%	–	–	0.00	–	–	0.00	–	–	0.00
p3.4.p_5%	530.00	530.00	104.55	530.00	530.00	33.76	530.00	530.00	22.95
p3.4.q_5%	500.00	500.00	2218.34	500.00	500.00	2383.30	500.00	500.00	2430.31
p3.4.r_5%	560.00	560.00	24.53	560.00	560.00	35.00	560.00	560.00	47.31
p3.4.s_5%	610.00	610.00	8.13	610.00	610.00	4.37	610.00	610.00	15.27
p3.4.t_5%	630.00	630.00	2.60	630.00	630.00	13.08	630.00	630.00	6.48
p4.2.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.2.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.2.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.2.d_5%	–	–	4827.47	–	–	7200.00	–	–	222.44
p4.2.e_5%	443.00	443.00	802.64	443.00	443.00	661.15	443.00	443.00	519.41
p4.2.f_5%	618.00	618.00	245.12	618.00	618.00	43.57	618.00	618.00	82.30
p4.2.g_5%	574.00	604.19	7200.00	574.00	574.00	4272.56	574.00	574.00	1307.21
p4.2.h_5%	718.00	748.65	7200.00	718.00	718.00	493.26	718.00	718.00	595.76
p4.2.i_5%	721.00	762.68	7200.00	728.00	728.00	1003.05	728.00	728.00	1650.08
p4.2.j_5%	873.00	873.00	977.63	873.00	873.00	566.79	873.00	873.00	306.44
p4.2.k_5%	955.00	967.03	7200.00	955.00	955.00	670.97	955.00	955.00	311.05
p4.2.l_5%	1046.00	1072.88	7200.00	1049.00	1049.00	2266.38	1049.00	1049.00	160.49
p4.2.m_5%	1092.00	1126.93	7200.00	1096.00	1096.00	2408.78	1096.00	1096.00	227.01
p4.2.n_5%	1071.00	1155.54	7200.00	1111.00	1111.00	3776.28	1111.00	1111.00	1618.87
p4.2.o_5%	1147.00	1169.66	7200.00	1149.00	1149.00	3987.48	1149.00	1149.00	368.74
p4.2.p_5%	1154.00	1183.30	7200.00	1153.00	1172.52	7200.00	1138.00	1166.50	7200.00
p4.2.q_5%	1192.00	1218.34	7200.00	1201.00	1209.52	7200.00	1201.00	1208.39	7200.00
p4.2.r_5%	1224.00	1232.28	7200.00	1225.00	1225.00	3640.27	1225.00	1225.00	5741.00
p4.2.s_5%	1213.00	1214.79	7200.00	1206.00	1215.00	7200.00	1213.00	1213.00	2659.11
p4.2.t_5%	1242.00	1243.00	7200.00	1242.00	1243.00	7200.00	1243.00	1243.00	3309.66
p4.3.a_5%	–	–	0.00	–	–	0.00	–	–	0.00

Continued on next page

Table A.10 — continued from previous page.

	B-B&C			CPA_1			CPA_2		
Instance	LB	UB	Time (s)	LB	UB	Time (s)	LB	UB	Time (s)
p4.3.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.3.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.3.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.3.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.3.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.3.g_5%	–	–	0.00	–	–	0.01	–	–	0.01
p4.3.h_5%	384.00	410.28	7200.00	383.00	421.23	7200.00	383.00	408.72	7200.00
p4.3.i_5%	742.00	742.00	1798.13	742.00	742.00	879.93	742.00	742.00	1091.06
p4.3.j_5%	724.00	724.00	2425.22	724.00	724.00	1730.63	724.00	724.00	1938.23
p4.3.k_5%	829.00	829.00	6695.63	829.00	829.00	1422.84	829.00	829.00	2261.30
p4.3.l_5%	886.00	929.39	7200.00	884.00	915.40	7200.00	888.00	914.19	7200.00
p4.3.m_5%	982.00	1015.89	7200.00	977.00	1003.68	7200.00	966.00	1010.54	7200.00
p4.3.n_5%	1030.00	1053.99	7200.00	1030.00	1030.00	2840.10	1030.00	1030.00	4381.97
p4.3.o_5%	1105.00	1154.38	7200.00	1110.00	1134.46	7200.00	1114.00	1131.78	7200.00
p4.3.p_5%	1152.00	1160.44	7200.00	1152.00	1152.00	5135.90	1152.00	1152.00	632.50
p4.3.q_5%	1176.00	1193.44	7200.00	1169.00	1185.89	7200.00	1176.00	1182.89	7200.00
p4.3.r_5%	1203.00	1237.57	7200.00	1215.00	1226.23	7200.00	1198.00	1229.37	7200.00
p4.3.s_5%	1259.00	1267.59	7200.00	1228.00	1266.49	7200.00	1259.00	1263.39	7200.00
p4.3.t_5%	1240.00	1256.00	7200.00	1249.00	1256.00	7200.00	1244.00	1256.00	7200.00
p4.4.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.4.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.4.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.4.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.4.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.4.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.4.g_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.4.h_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.4.i_5%	–	–	0.00	–	–	0.00	–	–	0.00
p4.4.j_5%	573.00	573.00	997.56	573.00	573.00	519.73	573.00	573.00	342.38
p4.4.k_5%	–	–	0.00	–	–	0.01	–	–	0.01
p4.4.l_5%	682.00	682.00	426.34	682.00	682.00	984.17	682.00	682.00	975.27
p4.4.m_5%	–	–	0.01	–	–	0.01	–	–	0.01
p4.4.n_5%	852.00	904.07	7200.00	842.00	908.24	7200.00	841.00	901.36	7200.00
p4.4.o_5%	922.00	985.09	7200.00	933.00	981.73	7200.00	942.00	980.38	7200.00
p4.4.p_5%	1048.00	1075.56	7200.00	1034.00	1069.96	7200.00	1048.00	1061.33	7200.00
p4.4.q_5%	1071.00	1132.58	7200.00	1048.00	1134.25	7200.00	1045.00	1130.15	7200.00
p4.4.r_5%	1109.00	1183.93	7200.00	1115.00	1174.28	7200.00	1084.00	1174.54	7200.00
p4.4.s_5%	1207.00	1234.43	7200.00	1127.00	1232.40	7200.00	1206.00	1226.32	7200.00
p4.4.t_5%	1243.00	1265.52	7200.00	1203.00	1261.96	7200.00	1242.00	1260.54	7200.00
p5.2.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.2.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.2.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.2.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.2.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.2.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.2.g_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.2.h_5%	–	–	0.91	–	–	1.65	–	–	1.87
p5.2.i_5%	350.00	350.00	116.18	350.00	350.00	90.32	350.00	350.00	95.09
p5.2.j_5%	460.00	460.00	11.19	460.00	460.00	7.96	460.00	460.00	21.61
p5.2.k_5%	505.00	505.00	26.42	505.00	505.00	18.25	505.00	505.00	24.96
p5.2.l_5%	660.00	660.00	12.16	660.00	660.00	17.67	660.00	660.00	12.51
p5.2.m_5%	730.00	730.00	10.03	730.00	730.00	4.96	730.00	730.00	8.22
p5.2.n_5%	815.00	815.00	119.75	815.00	815.00	78.33	815.00	815.00	77.57
p5.2.o_5%	900.00	900.00	126.37	900.00	900.00	122.83	900.00	900.00	66.57
p5.2.p_5%	1030.00	1030.00	0.80	1030.00	1030.00	1.68	1030.00	1030.00	2.35

Continued on next page

Table A.10 — continued from previous page.

	B-B&C			CPA$_1$			CPA$_2$		
Instance	LB	UB	Time (s)	LB	UB	Time (s)	LB	UB	Time (s)
p5.2.q_5%	1080.00	1080.00	5.73	1080.00	1080.00	9.38	1080.00	1080.00	63.83
p5.2.r_5%	1170.00	1170.00	13.63	1170.00	1170.00	12.91	1170.00	1170.00	12.34
p5.2.s_5%	1250.00	1250.00	2.10	1250.00	1250.00	5.00	1250.00	1250.00	2.90
p5.2.t_5%	1310.00	1310.00	26.58	1310.00	1310.00	4.04	1310.00	1310.00	8.43
p5.2.u_5%	1360.00	1360.00	7.73	1360.00	1360.00	6.65	1360.00	1360.00	9.48
p5.2.v_5%	1395.00	1395.00	3428.61	1395.00	1395.00	512.27	1395.00	1395.00	516.79
p5.2.w_5%	1465.00	1465.00	91.10	1465.00	1465.00	141.53	1465.00	1465.00	17.08
p5.2.x_5%	1490.00	1490.00	464.27	1490.00	1490.00	5.94	1490.00	1490.00	5.07
p5.2.y_5%	1535.00	1535.00	715.13	1535.00	1535.00	80.05	1535.00	1535.00	111.82
p5.2.z_5%	1570.00	1570.00	7.92	1570.00	1570.00	170.60	1570.00	1570.00	165.48
p5.3.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.3.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.3.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.3.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.3.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.3.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.3.g_5%	95.00	95.00	0.31	95.00	95.00	0.58	95.00	95.00	0.52
p5.3.h_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.3.i_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.3.j_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.3.k_5%	425.00	425.00	52.47	425.00	425.00	119.60	425.00	425.00	143.61
p5.3.l_5%	490.00	490.00	116.05	490.00	490.00	199.79	490.00	490.00	299.48
p5.3.m_5%	535.00	535.00	1020.06	535.00	535.00	2255.23	535.00	535.00	4186.74
p5.3.n_5%	665.00	665.00	295.31	665.00	665.00	595.05	665.00	665.00	339.62
p5.3.o_5%	740.00	740.00	468.27	740.00	740.00	880.16	740.00	740.00	239.89
p5.3.p_5%	860.00	860.00	1.78	860.00	860.00	2.76	860.00	860.00	4.08
p5.3.q_5%	965.00	965.00	11.40	965.00	965.00	11.31	965.00	965.00	12.72
p5.3.r_5%	985.00	985.00	2576.34	985.00	985.00	682.86	985.00	985.00	857.75
p5.3.s_5%	1090.00	1120.00	7200.00	1090.00	1121.63	7200.00	1090.00	1116.89	7200.00
p5.3.t_5%	1150.00	1205.02	7200.00	1150.00	1195.68	7200.00	1150.00	1190.61	7200.00
p5.3.u_5%	1225.00	1261.15	7200.00	1225.00	1253.97	7200.00	1225.00	1252.27	7200.00
p5.3.v_5%	1310.00	1339.87	7200.00	1315.00	1315.00	3594.90	1315.00	1323.80	7200.00
p5.3.w_5%	1385.00	1409.88	7200.00	1385.00	1405.53	7200.00	1390.00	1407.25	7200.00
p5.3.x_5%	1455.00	1474.64	7200.00	1455.00	1455.00	5868.47	1455.00	1455.00	4063.02
p5.3.y_5%	1500.00	1533.11	7200.00	1500.00	1531.32	7200.00	1500.00	1531.39	7200.00
p5.3.z_5%	1515.00	1542.96	7200.00	1515.00	1541.57	7200.00	1515.00	1540.00	7200.00
p5.4.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.4.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.4.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.4.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.4.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.4.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.4.g_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.4.h_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.4.i_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.4.j_5%	220.00	220.00	0.11	220.00	220.00	0.31	220.00	220.00	0.57
p5.4.k_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.4.l_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.4.m_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.4.n_5%	440.00	440.00	3.44	440.00	440.00	14.95	440.00	440.00	15.88
p5.4.o_5%	–	–	0.00	–	–	0.00	–	–	0.00
p5.4.p_5%	665.00	694.52	7200.00	665.00	701.27	7200.00	665.00	697.52	7200.00
p5.4.q_5%	715.00	725.40	7200.00	715.00	743.10	7200.00	715.00	730.05	7200.00
p5.4.r_5%	850.00	850.00	530.14	850.00	850.00	368.64	850.00	850.00	1244.17
p5.4.s_5%	935.00	955.06	7200.00	935.00	961.69	7200.00	935.00	961.25	7200.00
p5.4.t_5%	1060.00	1079.15	7200.00	1060.00	1060.00	6149.68	1060.00	1060.00	6238.71

Continued on next page

Table A.10 — continued from previous page.

	B-B&C			CPA_1			CPA_2		
Instance	LB	UB	Time (s)	LB	UB	Time (s)	LB	UB	Time (s)
p5.4.u_5%	1190.00	1190.00	1.32	1190.00	1190.00	2.56	1190.00	1190.00	2.23
p5.4.v_5%	1230.00	1252.75	7200.00	1230.00	1251.71	7200.00	1230.00	1250.59	7200.00
p5.4.w_5%	1260.00	1299.42	7200.00	1260.00	1286.46	7200.00	1260.00	1289.06	7200.00
p5.4.x_5%	1310.00	1353.36	7200.00	1300.00	1348.69	7200.00	1310.00	1348.68	7200.00
p5.4.y_5%	1400.00	1431.14	7200.00	1400.00	1425.33	7200.00	1400.00	1423.62	7200.00
p5.4.z_5%	1530.00	1530.00	282.43	1530.00	1530.00	943.60	1530.00	1530.00	135.11
p6.2.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.2.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.2.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.2.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.2.e_5%	282.00	282.00	0.35	282.00	282.00	0.33	282.00	282.00	0.35
p6.2.f_5%	474.00	474.00	0.55	474.00	474.00	0.75	474.00	474.00	0.71
p6.2.g_5%	522.00	522.00	9.11	522.00	522.00	20.01	522.00	522.00	27.67
p6.2.h_5%	708.00	708.00	1.38	708.00	708.00	1.81	708.00	708.00	2.49
p6.2.i_5%	828.00	828.00	1.57	828.00	828.00	7.43	828.00	828.00	4.72
p6.2.j_5%	846.00	864.36	7200.00	846.00	846.00	1143.76	846.00	846.00	881.75
p6.2.k_5%	948.00	969.29	7200.00	948.00	948.00	711.32	948.00	948.00	723.58
p6.2.l_5%	1008.00	1008.00	1170.47	1008.00	1008.00	362.08	1008.00	1008.00	175.41
p6.2.m_5%	1086.00	1086.00	4797.85	1086.00	1086.00	1018.09	1086.00	1086.00	1689.98
p6.2.n_5%	1170.00	1170.00	6.39	1170.00	1170.00	2.92	1170.00	1170.00	3.50
p6.3.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.3.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.3.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.3.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.3.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.3.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.3.g_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.3.h_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.3.i_5%	582.00	582.00	0.60	582.00	582.00	1.38	582.00	582.00	1.62
p6.3.j_5%	678.00	678.00	1.81	678.00	678.00	5.16	678.00	678.00	5.68
p6.3.k_5%	840.00	840.00	120.94	840.00	840.00	85.29	840.00	840.00	113.55
p6.3.l_5%	930.00	930.00	42.43	930.00	930.00	32.08	930.00	930.00	44.34
p6.3.m_5%	1014.00	1037.75	7200.00	1014.00	1014.00	4402.47	1014.00	1031.34	7200.00
p6.3.n_5%	1092.00	1092.00	591.30	1092.00	1092.00	1248.71	1092.00	1092.00	1567.65
p6.4.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.4.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.4.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.4.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.4.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.4.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.4.g_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.4.h_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.4.i_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.4.j_5%	–	–	0.00	–	–	0.00	–	–	0.00
p6.4.k_5%	318.00	318.00	0.36	318.00	318.00	2.23	318.00	318.00	2.91
p6.4.l_5%	660.00	660.00	36.49	660.00	660.00	144.18	660.00	660.00	107.05
p6.4.m_5%	786.00	786.00	13.16	786.00	786.00	30.63	786.00	786.00	55.59
p6.4.n_5%	990.00	990.00	1.85	990.00	990.00	5.19	990.00	990.00	5.93
p7.2.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.2.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.2.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.2.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.2.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.2.f_5%	–	–	0.00	–	–	0.00	–	–	0.00

Continued on next page

Table A.10 — continued from previous page.

	B-B&C			CPA_1			CPA_2		
Instance	LB	UB	Time (s)	LB	UB	Time (s)	LB	UB	Time (s)
p7.2.g_5%	–	–	0.75	–	–	1.37	–	–	1.72
p7.2.h_5%	337.00	337.00	48.54	337.00	337.00	20.55	337.00	337.00	42.74
p7.2.i_5%	–	–	0.01	–	–	0.01	–	–	0.01
p7.2.j_5%	-inf	inf	7200.00	–	–	7200.00	–	–	7200.00
p7.2.k_5%	-inf	inf	7200.00	–	–	7200.00	–	–	7200.00
p7.2.l_5%	636.00	691.57	7200.00	636.00	636.00	403.97	636.00	636.00	493.25
p7.2.m_5%	731.00	767.46	7200.00	731.00	731.00	835.87	731.00	731.00	448.92
p7.2.n_5%	614.00	739.49	7200.00	656.00	669.52	7200.00	656.00	656.00	1315.99
p7.2.o_5%	802.00	830.52	7200.00	802.00	802.00	1725.97	802.00	802.00	477.10
p7.2.p_5%	831.00	886.20	7200.00	844.00	844.00	7075.30	844.00	844.00	6525.18
p7.2.q_5%	862.00	917.67	7200.00	885.00	885.00	4880.19	885.00	885.00	5513.97
p7.2.r_5%	968.00	968.00	3237.01	968.00	968.00	393.81	968.00	968.00	332.91
p7.2.s_5%	992.00	1022.62	7200.00	1007.00	1007.00	1287.75	1007.00	1007.00	1017.59
p7.2.t_5%	1074.00	1084.17	7200.00	1075.00	1075.00	945.38	1075.00	1075.00	622.74
p7.3.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.3.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.3.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.3.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.3.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.3.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.3.g_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.3.h_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.3.i_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.3.j_5%	-inf	inf	7200.00	-inf	inf	7200.00	-inf	inf	7200.00
p7.3.k_5%	395.00	395.00	607.68	395.00	395.00	39.97	395.00	395.00	87.83
p7.3.l_5%	522.00	522.00	1461.13	522.00	522.00	486.58	522.00	522.00	467.19
p7.3.m_5%	–	–	0.01	–	–	0.01	–	–	0.01
p7.3.n_5%	–	–	82.96	–	–	344.24	–	–	135.44
p7.3.o_5%	602.00	602.00	105.77	602.00	602.00	196.10	602.00	602.00	204.23
p7.3.p_5%	757.00	794.38	7200.00	757.00	757.00	2489.18	757.00	757.00	4962.79
p7.3.q_5%	866.00	912.94	7200.00	874.00	874.00	1332.51	874.00	874.00	2216.66
p7.3.r_5%	923.00	959.05	7200.00	923.00	923.00	2550.24	923.00	923.00	5412.53
p7.3.s_5%	942.00	995.88	7200.00	949.00	974.87	7200.00	949.00	974.64	7200.00
p7.3.t_5%	1012.00	1062.79	7200.00	1034.00	1047.03	7200.00	1029.00	1053.26	7200.00
p7.4.a_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.4.b_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.4.c_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.4.d_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.4.e_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.4.f_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.4.g_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.4.h_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.4.i_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.4.j_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.4.k_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.4.l_5%	–	–	0.00	–	–	0.00	–	–	0.00
p7.4.m_5%	–	–	0.00	–	–	0.00	–	–	0.01
p7.4.n_5%	–	–	0.00	–	–	0.01	–	–	0.01
p7.4.o_5%	537.00	558.14	7200.00	537.00	537.00	892.01	537.00	537.00	1325.01
p7.4.p_5%	771.00	791.01	7200.00	771.00	779.03	7200.00	771.00	780.09	7200.00
p7.4.q_5%	778.00	809.66	7200.00	781.00	799.92	7200.00	781.00	799.11	7200.00
p7.4.r_5%	794.00	859.61	7200.00	798.00	808.51	7200.00	792.00	828.10	7200.00
p7.4.s_5%	857.00	902.83	7200.00	857.00	898.97	7200.00	857.00	895.08	7200.00
p7.4.t_5%	862.00	995.11	7200.00	939.00	959.26	7200.00	939.00	961.15	7200.00

Table A.7: Gaps referred to the bounds available at the root nodes of the branch-and-bound trees of B-B&C and CPA at solving the new STOP instances. Here, the CPLEX built-in cuts separated at the root nodes are considered.

B-B&C						
	Root gap		Root *primal* gap		Root *dual* gap	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1_5%	5.26	7.88	2.27	4.42	2.99	4.78
2_5%	1.29	3.54	1.29	3.54	0.00	0.00
3_5%	6.43	6.87	2.87	3.98	3.56	4.26
4_5%	17.47	18.30	11.18	11.87	4.62	5.75
5_5%	6.46	7.72	3.42	4.95	3.03	3.91
6_5%	2.00	3.79	0.43	1.83	1.57	2.45
7_5%	19.15	28.67	8.82	15.12	3.66	5.04
Total	9.04	15.78	4.71	9.04	3.04	4.48

CPA_1						
	Root gap		Root *primal* gap		Root *dual* gap	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1_5%	2.22	5.19	1.04	2.82	1.18	2.94
2_5%	1.16	3.74	1.16	3.74	0.00	0.00
3_5%	5.89	7.04	3.01	4.65	2.87	3.77
4_5%	13.47	12.32	9.80	9.53	3.67	4.38
5_5%	7.31	9.13	4.49	6.71	2.82	3.55
6_5%	2.45	3.83	0.63	1.49	1.82	2.58
7_5%	10.93	17.88	6.95	11.36	2.32	3.16
Total	6.84	10.96	4.28	7.63	2.31	3.48

CPA_2						
	Root gap		Root *primal* gap		Root *dual* gap	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1_5%	2.51	4.81	0.97	2.44	1.54	3.03
2_5%	1.06	3.41	1.06	3.41	0.00	0.00
3_5%	6.08	7.06	3.48	5.04	2.60	3.43
4_5%	11.84	15.69	7.21	8.02	2.96	4.49
5_5%	8.08	10.18	5.40	7.81	2.68	3.41
6_5%	2.35	4.34	0.62	2.24	1.73	2.60
7_5%	7.49	14.96	4.03	6.77	1.79	2.51
Total	6.26	10.86	3.67	6.36	2.08	3.29

Table A.8: Gaps referred to the bounds available at the root nodes of the branch-and-bound trees of CPLEX while directly solving formulations $\mathcal{F}_1$ and $\mathcal{F}_2$. Results for the new STOP instances. Here, the CPLEX built-in cuts are enabled.

$\mathcal{F}_1$						
	Root gap		Root *primal* gap		Root *dual* gap	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1_5%	6.47	9.56	2.90	5.36	3.57	5.80
2_5%	1.36	3.75	1.36	3.75	0.00	0.00
3_5%	6.91	7.25	3.29	4.51	3.62	4.30
4_5%	17.93	18.40	11.29	11.94	4.97	6.20
5_5%	6.59	7.67	3.54	5.04	3.05	3.94
6_5%	2.03	3.79	0.46	1.84	1.57	2.45
7_5%	19.23	28.67	8.86	15.10	3.70	5.11
Total	9.40	15.93	4.92	9.14	3.19	4.77

$\mathcal{F}_2$						
	Root gap		Root *primal* gap		Root *dual* gap	
Set	Avg (%)	StDev (%)	Avg (%)	StDev (%)	Avg (%)	StDev (%)
1_5%	6.20	8.67	2.67	4.68	3.53	5.38
2_5%	1.25	5.28	1.25	5.28	0.00	0.00
3_5%	7.56	7.96	3.73	4.87	3.84	4.50
4_5%	17.70	17.43	11.19	11.15	4.84	5.81
5_5%	8.30	9.82	5.21	7.33	3.09	4.04
6_5%	2.40	3.65	0.70	1.95	1.70	2.50
7_5%	18.87	27.44	8.58	12.89	3.62	4.99
Total	9.75	15.57	5.25	8.75	3.21	4.65

Table A.9: Extra information on the execution of B-B&C and CPA at solving the new STOP instances.

B-B&C				
	Final gap			
Set	Avg (%)	StDev (%)	Nodes (#)	GCCs (#)
1_5%	0.00	0.00	1966.44	30.83
2_5%	0.00	0.00	8.24	1.88
3_5%	0.00	0.00	78 611.77	144.73
4_5%	1.58	2.17	138 863.02	1059.98
5_5%	0.50	1.08	456 363.04	760.64
6_5%	0.16	0.57	259 702.45	269.10
7_5%	6.83	21.82	124 164.17	516.25
Total	1.42	8.91	173 407.28	453.79

CPA_1						
	Final gap			User cuts (#)		
Set	Avg (%)	StDev (%)	Nodes (#)	GCCs	CCs	LCIs
1_5%	0.00	0.00	1392.67	6.59	22.09	1.17
2_5%	0.00	0.00	59.06	0.15	0.45	0.15
3_5%	0.00	0.00	57 958.87	5.30	16.20	2.37
4_5%	1.13	2.30	214 328.93	23.37	66.45	0.37
5_5%	0.45	1.10	731 486.51	1.04	22.59	0.54
6_5%	0.00	0.00	319 153.62	0.38	5.12	0.24
7_5%	1.96	12.90	171 646.52	9.72	162.82	0.10
Total	0.57	5.20	251 094.60	7.13	46.29	0.75

CPA_2					
	Final gap			User cuts (#)	
Set	Avg (%)	StDev (%)	Nodes (#)	LCIs	CCCs
1_5%	0.00	0.00	2850.56	1.78	13.26
2_5%	0.00	0.00	10.82	0.33	1.00
3_5%	0.00	0.00	50 461.72	3.12	10.80
4_5%	0.88	1.93	222 886.62	1.02	88.57
5_5%	0.40	0.96	667 742.45	0.67	17.41
6_5%	0.04	0.26	353 146.43	0.29	0.71
7_5%	1.99	12.90	128 465.40	0.38	68.28
Total	0.53	5.17	235 605.10	1.14	31.51

A.5 Primal bounds obtained by the LNS heuristic

In Table A.11, we display the primal bounds obtained by the two best versions of the LNS heuristic (FP_cuts_LNS_5000 and OFP_cuts_LNS_5000) at solving the original TOP instances and the new STOP ones. For each instance and algorithm, we give the average (over the 10 executions) profit sum of the solutions obtained, and the profit sum of the best (in each round of 10 executions) solution found. We filled with dashes the entries related to the instances for which no feasible solution was found by the heuristic, including the cases of infeasible instances.

Table A.11: Bounds obtained by the two best versions of the LNS heuristic proposed.

	TOP				STOP			
	FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000	
	Profit sum		Profit sum		Profit sum		Profit sum	
	Avg	Best	Avg	Best	Avg	Best	Avg	Best
p1.2.a(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p1.2.b(_5%)	15.00	15.00	15.00	15.00	–	–	–	–
p1.2.c(_5%)	20.00	20.00	20.00	20.00	–	–	–	–
p1.2.d(_5%)	30.00	30.00	30.00	30.00	15.00	15.00	15.00	15.00
p1.2.e(_5%)	45.00	45.00	45.00	45.00	–	–	–	–
p1.2.f(_5%)	80.00	80.00	80.00	80.00	–	–	–	–
p1.2.g(_5%)	90.00	90.00	90.00	90.00	–	–	–	–
p1.2.h(_5%)	99.00	110.00	110.00	110.00	40.50	45.00	45.00	45.00
p1.2.i(_5%)	135.00	135.00	135.00	135.00	99.00	110.00	110.00	110.00
p1.2.j(_5%)	155.00	155.00	155.00	155.00	140.00	140.00	140.00	140.00
p1.2.k(_5%)	175.00	175.00	175.00	175.00	150.00	150.00	150.00	150.00
p1.2.l(_5%)	195.00	195.00	195.00	195.00	185.00	185.00	185.00	185.00
p1.2.m(_5%)	215.00	215.00	215.00	215.00	190.00	190.00	190.00	190.00
p1.2.n(_5%)	235.00	235.00	235.00	235.00	215.00	215.00	215.00	215.00
p1.2.o(_5%)	240.00	240.00	240.00	240.00	220.00	220.00	220.00	220.00
p1.2.p(_5%)	250.00	250.00	250.00	250.00	225.00	225.00	225.00	225.00
p1.2.q(_5%)	265.00	265.00	265.00	265.00	245.00	245.00	245.00	245.00
p1.2.r(_5%)	280.00	280.00	280.00	280.00	265.00	265.00	265.00	265.00
p1.3.a(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p1.3.b(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p1.3.c(_5%)	15.00	15.00	15.00	15.00	–	–	–	–
p1.3.d(_5%)	15.00	15.00	15.00	15.00	–	–	–	–
p1.3.e(_5%)	30.00	30.00	30.00	30.00	–	–	–	–
p1.3.f(_5%)	40.00	40.00	40.00	40.00	–	–	–	–
p1.3.g(_5%)	15.00	50.00	50.00	50.00	–	–	–	–
p1.3.h(_5%)	70.00	70.00	70.00	70.00	–	–	–	–
p1.3.i(_5%)	105.00	105.00	105.00	105.00	90.00	90.00	90.00	90.00
p1.3.j(_5%)	115.00	115.00	115.00	115.00	80.00	80.00	80.00	80.00
p1.3.k(_5%)	135.00	135.00	135.00	135.00	115.00	115.00	115.00	115.00
p1.3.l(_5%)	155.00	155.00	155.00	155.00	140.00	140.00	140.00	140.00
p1.3.m(_5%)	175.00	175.00	175.00	175.00	130.00	130.00	130.00	130.00
p1.3.n(_5%)	190.00	190.00	190.00	190.00	165.00	165.00	165.00	165.00
p1.3.o(_5%)	205.00	205.00	205.00	205.00	180.00	180.00	180.00	180.00
p1.3.p(_5%)	220.00	220.00	220.00	220.00	200.00	200.00	200.00	200.00
p1.3.q(_5%)	230.00	230.00	230.00	230.00	210.00	210.00	210.00	210.00
p1.3.r(_5%)	250.00	250.00	250.00	250.00	225.00	225.00	225.00	225.00
p1.4.a(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p1.4.b(_5%)	0.00	0.00	0.00	0.00	–	–	–	–

Continued on next page

Table A.11 — continued from previous page.

	TOP				STOP			
	FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000	
	Profit sum		Profit sum		Profit sum		Profit sum	
	Avg	Best	Avg	Best	Avg	Best	Avg	Best
p1.4.c(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p1.4.d(_5%)	15.00	15.00	15.00	15.00	–	–	–	–
p1.4.e(_5%)	15.00	15.00	15.00	15.00	–	–	–	–
p1.4.f(_5%)	25.00	25.00	25.00	25.00	–	–	–	–
p1.4.g(_5%)	35.00	35.00	35.00	35.00	–	–	–	–
p1.4.h(_5%)	45.00	45.00	45.00	45.00	–	–	–	–
p1.4.i(_5%)	54.00	60.00	60.00	60.00	–	–	–	–
p1.4.j(_5%)	75.00	75.00	75.00	75.00	–	–	–	–
p1.4.k(_5%)	100.00	100.00	100.00	100.00	–	–	–	–
p1.4.l(_5%)	120.00	120.00	120.00	120.00	–	–	–	–
p1.4.m(_5%)	130.00	130.00	130.00	130.00	115.00	115.00	115.00	115.00
p1.4.n(_5%)	155.00	155.00	155.00	155.00	135.00	135.00	135.00	135.00
p1.4.o(_5%)	165.00	165.00	165.00	165.00	–	–	–	–
p1.4.p(_5%)	175.00	175.00	175.00	175.00	150.00	150.00	150.00	150.00
p1.4.q(_5%)	190.00	190.00	190.00	190.00	165.00	165.00	165.00	165.00
p1.4.r(_5%)	210.00	210.00	210.00	210.00	195.00	195.00	195.00	195.00
p2.2.a(_5%)	90.00	90.00	90.00	90.00	–	–	–	–
p2.2.b(_5%)	48.00	120.00	120.00	120.00	–	–	–	–
p2.2.c(_5%)	140.00	140.00	140.00	140.00	–	–	–	–
p2.2.d(_5%)	112.00	160.00	144.00	160.00	–	–	–	–
p2.2.e(_5%)	190.00	190.00	190.00	190.00	–	–	–	–
p2.2.f(_5%)	200.00	200.00	200.00	200.00	–	–	–	–
p2.2.g(_5%)	200.00	200.00	200.00	200.00	–	–	–	–
p2.2.h(_5%)	230.00	230.00	230.00	230.00	200.00	200.00	200.00	200.00
p2.2.i(_5%)	230.00	230.00	230.00	230.00	–	–	–	–
p2.2.j(_5%)	234.00	260.00	260.00	260.00	220.00	220.00	220.00	220.00
p2.2.k(_5%)	275.00	275.00	247.50	275.00	210.00	210.00	210.00	210.00
p2.3.a(_5%)	70.00	70.00	70.00	70.00	–	–	–	–
p2.3.b(_5%)	70.00	70.00	70.00	70.00	–	–	–	–
p2.3.c(_5%)	42.00	105.00	105.00	105.00	–	–	–	–
p2.3.d(_5%)	105.00	105.00	105.00	105.00	–	–	–	–
p2.3.e(_5%)	120.00	120.00	120.00	120.00	–	–	–	–
p2.3.f(_5%)	108.00	120.00	120.00	120.00	–	–	–	–
p2.3.g(_5%)	145.00	145.00	145.00	145.00	–	–	–	–
p2.3.h(_5%)	165.00	165.00	165.00	165.00	–	–	–	–
p2.3.i(_5%)	180.00	200.00	200.00	200.00	170.00	170.00	170.00	170.00
p2.3.j(_5%)	180.00	200.00	200.00	200.00	160.00	160.00	160.00	160.00
p2.3.k(_5%)	200.00	200.00	200.00	200.00	170.00	170.00	170.00	170.00
p2.4.a(_5%)	10.00	10.00	10.00	10.00	–	–	–	–
p2.4.b(_5%)	70.00	70.00	70.00	70.00	–	–	–	–
p2.4.c(_5%)	70.00	70.00	70.00	70.00	–	–	–	–
p2.4.d(_5%)	70.00	70.00	70.00	70.00	–	–	–	–
p2.4.e(_5%)	70.00	70.00	70.00	70.00	–	–	–	–
p2.4.f(_5%)	105.00	105.00	105.00	105.00	–	–	–	–
p2.4.g(_5%)	105.00	105.00	105.00	105.00	–	–	–	–
p2.4.h(_5%)	108.00	120.00	120.00	120.00	95.00	95.00	95.00	95.00
p2.4.i(_5%)	108.00	120.00	120.00	120.00	–	–	–	–
p2.4.j(_5%)	108.00	120.00	120.00	120.00	–	–	–	–
p2.4.k(_5%)	180.00	180.00	180.00	180.00	145.00	145.00	145.00	145.00
p3.2.a(_5%)	90.00	90.00	90.00	90.00	–	–	–	–
p3.2.b(_5%)	150.00	150.00	150.00	150.00	–	–	–	–

Continued on next page

Table A.11 — continued from previous page.

	TOP				STOP			
	FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000	
	Profit sum		Profit sum		Profit sum		Profit sum	
	Avg	Best	Avg	Best	Avg	Best	Avg	Best
p3.2.c(_5%)	180.00	180.00	180.00	180.00	–	–	–	–
p3.2.d(_5%)	220.00	220.00	220.00	220.00	–	–	–	–
p3.2.e(_5%)	260.00	260.00	260.00	260.00	220.00	220.00	220.00	220.00
p3.2.f(_5%)	300.00	300.00	300.00	300.00	–	–	–	–
p3.2.g(_5%)	360.00	360.00	360.00	360.00	290.00	290.00	290.00	290.00
p3.2.h(_5%)	410.00	410.00	410.00	410.00	360.00	360.00	360.00	360.00
p3.2.i(_5%)	460.00	460.00	460.00	460.00	400.00	400.00	400.00	400.00
p3.2.j(_5%)	510.00	510.00	510.00	510.00	470.00	470.00	470.00	470.00
p3.2.k(_5%)	550.00	550.00	550.00	550.00	510.00	510.00	510.00	510.00
p3.2.l(_5%)	590.00	590.00	590.00	590.00	540.00	540.00	540.00	540.00
p3.2.m(_5%)	620.00	620.00	620.00	620.00	560.00	560.00	560.00	560.00
p3.2.n(_5%)	660.00	660.00	660.00	660.00	610.00	610.00	610.00	610.00
p3.2.o(_5%)	690.00	690.00	690.00	690.00	660.00	660.00	660.00	660.00
p3.2.p(_5%)	720.00	720.00	720.00	720.00	690.00	690.00	690.00	690.00
p3.2.q(_5%)	760.00	760.00	760.00	760.00	710.00	710.00	710.00	710.00
p3.2.r(_5%)	790.00	790.00	790.00	790.00	740.00	740.00	740.00	740.00
p3.2.s(_5%)	800.00	800.00	800.00	800.00	730.00	730.00	730.00	730.00
p3.2.t(_5%)	800.00	800.00	800.00	800.00	700.00	700.00	700.00	700.00
p3.3.a(_5%)	30.00	30.00	30.00	30.00	–	–	–	–
p3.3.b(_5%)	27.00	90.00	90.00	90.00	–	–	–	–
p3.3.c(_5%)	120.00	120.00	120.00	120.00	–	–	–	–
p3.3.d(_5%)	170.00	170.00	170.00	170.00	150.00	150.00	150.00	150.00
p3.3.e(_5%)	200.00	200.00	200.00	200.00	–	–	–	–
p3.3.f(_5%)	230.00	230.00	230.00	230.00	160.00	160.00	160.00	160.00
p3.3.g(_5%)	270.00	270.00	270.00	270.00	–	–	–	–
p3.3.h(_5%)	300.00	300.00	300.00	300.00	–	–	–	–
p3.3.i(_5%)	330.00	330.00	330.00	330.00	–	–	–	–
p3.3.j(_5%)	380.00	380.00	380.00	380.00	360.00	360.00	360.00	360.00
p3.3.k(_5%)	440.00	440.00	440.00	440.00	400.00	400.00	400.00	400.00
p3.3.l(_5%)	480.00	480.00	480.00	480.00	–	–	–	–
p3.3.m(_5%)	520.00	520.00	520.00	520.00	470.00	470.00	470.00	470.00
p3.3.n(_5%)	570.00	570.00	570.00	570.00	500.00	500.00	500.00	500.00
p3.3.o(_5%)	590.00	590.00	590.00	590.00	560.00	560.00	560.00	560.00
p3.3.p(_5%)	640.00	640.00	640.00	640.00	570.00	570.00	570.00	570.00
p3.3.q(_5%)	680.00	680.00	680.00	680.00	650.00	650.00	650.00	650.00
p3.3.r(_5%)	710.00	710.00	710.00	710.00	660.00	660.00	660.00	660.00
p3.3.s(_5%)	720.00	720.00	720.00	720.00	670.00	670.00	670.00	670.00
p3.3.t(_5%)	760.00	760.00	760.00	760.00	700.00	700.00	700.00	700.00
p3.4.a(_5%)	20.00	20.00	20.00	20.00	–	–	–	–
p3.4.b(_5%)	30.00	30.00	30.00	30.00	–	–	–	–
p3.4.c(_5%)	90.00	90.00	90.00	90.00	–	–	–	–
p3.4.d(_5%)	40.00	100.00	100.00	100.00	–	–	–	–
p3.4.e(_5%)	56.00	140.00	140.00	140.00	–	–	–	–
p3.4.f(_5%)	190.00	190.00	190.00	190.00	–	–	–	–
p3.4.g(_5%)	220.00	220.00	220.00	220.00	190.00	190.00	190.00	190.00
p3.4.h(_5%)	240.00	240.00	240.00	240.00	220.00	220.00	220.00	220.00
p3.4.i(_5%)	270.00	270.00	270.00	270.00	230.00	230.00	230.00	230.00
p3.4.j(_5%)	310.00	310.00	310.00	310.00	–	–	–	–
p3.4.k(_5%)	350.00	350.00	350.00	350.00	280.00	280.00	280.00	280.00
p3.4.l(_5%)	380.00	380.00	380.00	380.00	–	–	–	–
p3.4.m(_5%)	390.00	390.00	390.00	390.00	340.00	340.00	340.00	340.00
p3.4.n(_5%)	440.00	440.00	440.00	440.00	380.00	380.00	380.00	380.00
p3.4.o(_5%)	500.00	500.00	500.00	500.00	–	–	–	–

Continued on next page

Table A.11 — continued from previous page.

	TOP				STOP			
	FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000	
	Profit sum		Profit sum		Profit sum		Profit sum	
	Avg	Best	Avg	Best	Avg	Best	Avg	Best
p3.4.p(_5%)	560.00	560.00	560.00	560.00	530.00	530.00	530.00	530.00
p3.4.q(_5%)	560.00	560.00	560.00	560.00	500.00	500.00	500.00	500.00
p3.4.r(_5%)	600.00	600.00	600.00	600.00	560.00	560.00	560.00	560.00
p3.4.s(_5%)	670.00	670.00	670.00	670.00	610.00	610.00	610.00	610.00
p3.4.t(_5%)	670.00	670.00	670.00	670.00	630.00	630.00	630.00	630.00
p4.2.a(_5%)	164.80	206.00	206.00	206.00	–	–	–	–
p4.2.b(_5%)	306.90	341.00	341.00	341.00	–	–	–	–
p4.2.c(_5%)	452.00	452.00	452.00	452.00	–	–	–	–
p4.2.d(_5%)	529.10	531.00	527.20	530.00	–	–	–	–
p4.2.e(_5%)	618.00	618.00	618.00	618.00	443.00	443.00	443.00	443.00
p4.2.f(_5%)	677.60	679.00	679.50	687.00	618.00	618.00	618.00	618.00
p4.2.g(_5%)	753.70	757.00	752.10	756.00	574.00	574.00	574.00	574.00
p4.2.h(_5%)	827.10	835.00	827.70	835.00	714.80	718.00	716.40	718.00
p4.2.i(_5%)	912.60	918.00	918.00	918.00	726.80	728.00	726.80	728.00
p4.2.j(_5%)	962.80	964.00	962.80	964.00	869.20	873.00	868.60	873.00
p4.2.k(_5%)	1022.00	1022.00	1022.00	1022.00	952.00	955.00	952.00	955.00
p4.2.l(_5%)	1060.50	1074.00	1073.40	1074.00	1047.90	1049.00	1049.00	1049.00
p4.2.m(_5%)	1129.50	1132.00	1130.10	1132.00	1085.90	1096.00	1095.00	1096.00
p4.2.n(_5%)	1173.20	1174.00	1173.80	1174.00	1090.60	1111.00	1104.50	1111.00
p4.2.o(_5%)	1218.00	1218.00	1218.00	1218.00	1142.50	1149.00	1147.90	1149.00
p4.2.p(_5%)	1241.10	1242.00	1241.20	1242.00	1154.10	1157.00	1154.20	1157.00
p4.2.q(_5%)	1265.60	1268.00	1265.40	1268.00	1198.10	1202.00	1197.60	1201.00
p4.2.r(_5%)	1288.10	1292.00	1289.30	1292.00	1223.30	1225.00	1223.80	1225.00
p4.2.s(_5%)	1304.00	1304.00	1304.00	1304.00	1213.00	1213.00	1213.00	1213.00
p4.2.t(_5%)	1306.00	1306.00	1306.00	1306.00	1243.00	1243.00	1243.00	1243.00
p4.3.a(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p4.3.b(_5%)	38.00	38.00	38.00	38.00	–	–	–	–
p4.3.c(_5%)	193.00	193.00	193.00	193.00	–	–	–	–
p4.3.d(_5%)	335.00	335.00	335.00	335.00	–	–	–	–
p4.3.e(_5%)	468.00	468.00	468.00	468.00	–	–	–	–
p4.3.f(_5%)	579.00	579.00	579.00	579.00	–	–	–	–
p4.3.g(_5%)	652.90	653.00	652.90	653.00	–	–	–	–
p4.3.h(_5%)	726.80	729.00	727.60	729.00	383.30	384.00	383.20	384.00
p4.3.i(_5%)	809.00	809.00	809.00	809.00	742.00	742.00	742.00	742.00
p4.3.j(_5%)	858.70	861.00	860.00	861.00	724.00	724.00	724.00	724.00
p4.3.k(_5%)	918.20	919.00	918.20	919.00	828.20	829.00	828.20	829.00
p4.3.l(_5%)	967.80	970.00	970.60	979.00	887.20	888.00	885.80	888.00
p4.3.m(_5%)	1053.40	1063.00	1053.80	1063.00	969.00	982.00	968.40	982.00
p4.3.n(_5%)	1120.40	1121.00	1121.00	1121.00	1024.90	1028.00	1026.20	1028.00
p4.3.o(_5%)	1167.70	1170.00	1168.90	1171.00	1108.60	1114.00	1107.90	1111.00
p4.3.p(_5%)	1222.00	1222.00	1222.00	1222.00	1150.20	1152.00	1150.70	1152.00
p4.3.q(_5%)	1253.00	1253.00	1252.80	1253.00	1171.50	1177.00	1175.60	1177.00
p4.3.r(_5%)	1268.90	1271.00	1269.60	1273.00	1215.00	1215.00	1215.00	1215.00
p4.3.s(_5%)	1295.00	1295.00	1295.00	1295.00	1258.70	1259.00	1258.10	1259.00
p4.3.t(_5%)	1304.30	1305.00	1304.30	1305.00	1254.00	1254.00	1254.00	1255.00
p4.4.a(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p4.4.b(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p4.4.c(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p4.4.d(_5%)	38.00	38.00	38.00	38.00	–	–	–	–
p4.4.e(_5%)	183.00	183.00	183.00	183.00	–	–	–	–
p4.4.f(_5%)	324.00	324.00	324.00	324.00	–	–	–	–
p4.4.g(_5%)	461.00	461.00	461.00	461.00	–	–	–	–

Continued on next page

Table A.11 — continued from previous page.

	TOP				STOP			
	FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000	
	Profit sum		Profit sum		Profit sum		Profit sum	
	Avg	Best	Avg	Best	Avg	Best	Avg	Best
p4.4.h(_5%)	571.00	571.00	571.00	571.00	–	–	–	–
p4.4.i(_5%)	657.00	657.00	657.00	657.00	–	–	–	–
p4.4.j(_5%)	731.90	732.00	731.60	732.00	573.00	573.00	573.00	573.00
p4.4.k(_5%)	819.90	821.00	819.80	821.00	–	–	–	–
p4.4.l(_5%)	877.70	879.00	877.90	879.00	682.00	682.00	682.00	682.00
p4.4.m(_5%)	915.30	919.00	913.70	916.00	–	–	–	–
p4.4.n(_5%)	967.20	976.00	965.30	971.00	847.30	852.00	847.80	852.00
p4.4.o(_5%)	1050.70	1060.00	1040.60	1051.00	943.89	945.00	944.20	945.00
p4.4.p(_5%)	1121.40	1124.00	1121.30	1124.00	1046.70	1048.00	1046.70	1048.00
p4.4.q(_5%)	1160.20	1161.00	1159.90	1161.00	1076.60	1077.00	1076.20	1078.00
p4.4.r(_5%)	1210.00	1216.00	1207.50	1216.00	1128.50	1137.00	1132.80	1137.00
p4.4.s(_5%)	1257.90	1260.00	1256.70	1260.00	1205.90	1208.00	1206.20	1208.00
p4.4.t(_5%)	1283.10	1285.00	1283.30	1285.00	1241.70	1244.00	1242.10	1244.00
p5.2.a(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p5.2.b(_5%)	20.00	20.00	20.00	20.00	–	–	–	–
p5.2.c(_5%)	50.00	50.00	50.00	50.00	–	–	–	–
p5.2.d(_5%)	80.00	80.00	80.00	80.00	–	–	–	–
p5.2.e(_5%)	180.00	180.00	180.00	180.00	–	–	–	–
p5.2.f(_5%)	216.00	240.00	240.00	240.00	–	–	–	–
p5.2.g(_5%)	320.00	320.00	320.00	320.00	–	–	–	–
p5.2.h(_5%)	410.00	410.00	410.00	410.00	–	–	–	–
p5.2.i(_5%)	480.00	480.00	480.00	480.00	350.00	350.00	350.00	350.00
p5.2.j(_5%)	580.00	580.00	580.00	580.00	460.00	460.00	460.00	460.00
p5.2.k(_5%)	670.00	670.00	670.00	670.00	505.00	505.00	505.00	505.00
p5.2.l(_5%)	800.00	800.00	800.00	800.00	660.00	660.00	660.00	660.00
p5.2.m(_5%)	860.00	860.00	860.00	860.00	730.00	730.00	730.00	730.00
p5.2.n(_5%)	924.00	925.00	924.50	925.00	815.00	815.00	815.00	815.00
p5.2.o(_5%)	1020.00	1020.00	1020.00	1020.00	900.00	900.00	900.00	900.00
p5.2.p(_5%)	1150.00	1150.00	1150.00	1150.00	1030.00	1030.00	1030.00	1030.00
p5.2.q(_5%)	1195.00	1195.00	1195.00	1195.00	1080.00	1080.00	1080.00	1080.00
p5.2.r(_5%)	1260.00	1260.00	1260.00	1260.00	1170.00	1170.00	1170.00	1170.00
p5.2.s(_5%)	1340.00	1340.00	1340.00	1340.00	1250.00	1250.00	1250.00	1250.00
p5.2.t(_5%)	1398.00	1400.00	1398.00	1400.00	1310.00	1310.00	1309.00	1310.00
p5.2.u(_5%)	1460.00	1460.00	1460.00	1460.00	1360.00	1360.00	1360.00	1360.00
p5.2.v(_5%)	1504.50	1505.00	1505.00	1505.00	1395.00	1395.00	1394.50	1395.00
p5.2.w(_5%)	1562.50	1565.00	1561.00	1565.00	1464.00	1465.00	1465.00	1465.00
p5.2.x(_5%)	1610.00	1610.00	1610.00	1610.00	1488.50	1490.00	1490.00	1490.00
p5.2.y(_5%)	1645.00	1645.00	1645.00	1645.00	1535.00	1535.00	1534.50	1535.00
p5.2.z(_5%)	1680.00	1680.00	1680.00	1680.00	1570.00	1570.00	1570.00	1570.00
p5.3.a(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p5.3.b(_5%)	15.00	15.00	15.00	15.00	–	–	–	–
p5.3.c(_5%)	20.00	20.00	20.00	20.00	–	–	–	–
p5.3.d(_5%)	60.00	60.00	60.00	60.00	–	–	–	–
p5.3.e(_5%)	95.00	95.00	95.00	95.00	–	–	–	–
p5.3.f(_5%)	88.00	110.00	110.00	110.00	–	–	–	–
p5.3.g(_5%)	185.00	185.00	185.00	185.00	95.00	95.00	95.00	95.00
p5.3.h(_5%)	260.00	260.00	260.00	260.00	–	–	–	–
p5.3.i(_5%)	335.00	335.00	335.00	335.00	–	–	–	–
p5.3.j(_5%)	470.00	470.00	470.00	470.00	–	–	–	–
p5.3.k(_5%)	495.00	495.00	495.00	495.00	425.00	425.00	425.00	425.00
p5.3.l(_5%)	595.00	595.00	595.00	595.00	490.00	490.00	490.00	490.00
p5.3.m(_5%)	650.00	650.00	650.00	650.00	535.00	535.00	535.00	535.00

Continued on next page

Table A.11 — continued from previous page.

	TOP				STOP			
	FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000	
	Profit sum		Profit sum		Profit sum		Profit sum	
	Avg	Best	Avg	Best	Avg	Best	Avg	Best
p5.3.n(_5%)	755.00	755.00	755.00	755.00	665.00	665.00	665.00	665.00
p5.3.o(_5%)	870.00	870.00	870.00	870.00	740.00	740.00	740.00	740.00
p5.3.p(_5%)	990.00	990.00	990.00	990.00	860.00	860.00	860.00	860.00
p5.3.q(_5%)	1070.00	1070.00	1070.00	1070.00	965.00	965.00	965.00	965.00
p5.3.r(_5%)	1125.00	1125.00	1125.00	1125.00	985.00	985.00	985.00	985.00
p5.3.s(_5%)	1190.00	1190.00	1190.00	1190.00	1090.00	1090.00	1090.00	1090.00
p5.3.t(_5%)	1260.00	1260.00	1260.00	1260.00	1150.00	1150.00	1150.00	1150.00
p5.3.u(_5%)	1345.00	1345.00	1345.00	1345.00	1224.00	1225.00	1224.50	1225.00
p5.3.v(_5%)	1424.00	1425.00	1424.00	1425.00	1314.00	1315.00	1314.00	1315.00
p5.3.w(_5%)	1485.00	1485.00	1485.00	1485.00	1387.50	1390.00	1387.00	1390.00
p5.3.x(_5%)	1549.50	1555.00	1548.00	1555.00	1447.50	1450.00	1447.00	1450.00
p5.3.y(_5%)	1590.00	1590.00	1591.50	1595.00	1500.50	1505.00	1500.50	1505.00
p5.3.z(_5%)	1635.00	1635.00	1635.00	1635.00	1515.00	1515.00	1515.00	1515.00
p5.4.a(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p5.4.b(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p5.4.c(_5%)	20.00	20.00	20.00	20.00	–	–	–	–
p5.4.d(_5%)	20.00	20.00	20.00	20.00	–	–	–	–
p5.4.e(_5%)	20.00	20.00	20.00	20.00	–	–	–	–
p5.4.f(_5%)	80.00	80.00	80.00	80.00	–	–	–	–
p5.4.g(_5%)	140.00	140.00	140.00	140.00	–	–	–	–
p5.4.h(_5%)	140.00	140.00	140.00	140.00	–	–	–	–
p5.4.i(_5%)	240.00	240.00	240.00	240.00	–	–	–	–
p5.4.j(_5%)	340.00	340.00	340.00	340.00	220.00	220.00	220.00	220.00
p5.4.k(_5%)	340.00	340.00	340.00	340.00	–	–	–	–
p5.4.l(_5%)	430.00	430.00	430.00	430.00	–	–	–	–
p5.4.m(_5%)	555.00	555.00	555.00	555.00	–	–	–	–
p5.4.n(_5%)	620.00	620.00	620.00	620.00	440.00	440.00	440.00	440.00
p5.4.o(_5%)	690.00	690.00	690.00	690.00	–	–	–	–
p5.4.p(_5%)	765.00	765.00	765.00	765.00	665.00	665.00	665.00	665.00
p5.4.q(_5%)	860.00	860.00	860.00	860.00	715.00	715.00	715.00	715.00
p5.4.r(_5%)	960.00	960.00	960.00	960.00	850.00	850.00	850.00	850.00
p5.4.s(_5%)	1029.50	1030.00	1030.00	1030.00	935.00	935.00	935.00	935.00
p5.4.t(_5%)	1160.00	1160.00	1160.00	1160.00	1060.00	1060.00	1060.00	1060.00
p5.4.u(_5%)	1300.00	1300.00	1300.00	1300.00	1190.00	1190.00	1190.00	1190.00
p5.4.v(_5%)	1320.00	1320.00	1320.00	1320.00	–	–	–	–
p5.4.w(_5%)	1390.00	1390.00	1389.00	1390.00	1258.00	1260.00	1256.50	1260.00
p5.4.x(_5%)	1449.50	1450.00	1450.00	1450.00	1310.00	1310.00	1310.00	1310.00
p5.4.y(_5%)	1520.00	1520.00	1520.00	1520.00	1397.00	1400.00	1397.00	1400.00
p5.4.z(_5%)	1620.00	1620.00	1620.00	1620.00	1530.00	1530.00	1530.00	1530.00
p6.2.a(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.2.b(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.2.c(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.2.d(_5%)	134.40	192.00	172.80	192.00	–	–	–	–
p6.2.e(_5%)	360.00	360.00	360.00	360.00	282.00	282.00	282.00	282.00
p6.2.f(_5%)	588.00	588.00	588.00	588.00	474.00	474.00	474.00	474.00
p6.2.g(_5%)	660.00	660.00	660.00	660.00	522.00	522.00	522.00	522.00
p6.2.h(_5%)	780.00	780.00	780.00	780.00	708.00	708.00	708.00	708.00
p6.2.i(_5%)	888.00	888.00	888.00	888.00	828.00	828.00	828.00	828.00
p6.2.j(_5%)	948.00	948.00	948.00	948.00	846.00	846.00	846.00	846.00
p6.2.k(_5%)	1032.00	1032.00	1032.00	1032.00	948.00	948.00	948.00	948.00
p6.2.l(_5%)	1116.00	1116.00	1116.00	1116.00	1008.00	1008.00	1008.00	1008.00
p6.2.m(_5%)	1188.00	1188.00	1188.00	1188.00	1086.00	1086.00	1086.00	1086.00

Continued on next page

Table A.11 — continued from previous page.

	TOP				STOP			
	FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000	
	Profit sum		Profit sum		Profit sum		Profit sum	
	Avg	Best	Avg	Best	Avg	Best	Avg	Best
p6.2.n(_5%)	1260.00	1260.00	1260.00	1260.00	1169.40	1170.00	1170.00	1170.00
p6.3.a(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.3.b(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.3.c(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.3.d(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.3.e(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.3.f(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.3.g(_5%)	282.00	282.00	282.00	282.00	–	–	–	–
p6.3.h(_5%)	444.00	444.00	444.00	444.00	–	–	–	–
p6.3.i(_5%)	642.00	642.00	642.00	642.00	582.00	582.00	582.00	582.00
p6.3.j(_5%)	828.00	828.00	828.00	828.00	678.00	678.00	678.00	678.00
p6.3.k(_5%)	894.00	894.00	894.00	894.00	840.00	840.00	840.00	840.00
p6.3.l(_5%)	1002.00	1002.00	1002.00	1002.00	930.00	930.00	930.00	930.00
p6.3.m(_5%)	1080.00	1080.00	1080.00	1080.00	1014.00	1014.00	1014.00	1014.00
p6.3.n(_5%)	1170.00	1170.00	1170.00	1170.00	1092.00	1092.00	1090.80	1092.00
p6.4.a(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.4.b(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.4.c(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.4.d(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.4.e(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.4.f(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.4.g(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.4.h(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.4.i(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p6.4.j(_5%)	366.00	366.00	366.00	366.00	–	–	–	–
p6.4.k(_5%)	528.00	528.00	528.00	528.00	318.00	318.00	318.00	318.00
p6.4.l(_5%)	696.00	696.00	696.00	696.00	660.00	660.00	660.00	660.00
p6.4.m(_5%)	912.00	912.00	912.00	912.00	785.40	786.00	786.00	786.00
p6.4.n(_5%)	1068.00	1068.00	1068.00	1068.00	990.00	990.00	990.00	990.00
p7.2.a(_5%)	30.00	30.00	30.00	30.00	–	–	–	–
p7.2.b(_5%)	64.00	64.00	64.00	64.00	–	–	–	–
p7.2.c(_5%)	30.30	101.00	101.00	101.00	–	–	–	–
p7.2.d(_5%)	190.00	190.00	190.00	190.00	–	–	–	–
p7.2.e(_5%)	231.50	290.00	289.40	290.00	–	–	–	–
p7.2.f(_5%)	387.00	387.00	387.00	387.00	–	–	–	–
p7.2.g(_5%)	459.00	459.00	459.00	459.00	–	–	–	–
p7.2.h(_5%)	521.00	521.00	521.00	521.00	337.00	337.00	337.00	337.00
p7.2.i(_5%)	579.20	580.00	579.60	580.00	–	–	–	–
p7.2.j(_5%)	646.00	646.00	645.80	646.00	–	–	–	–
p7.2.k(_5%)	702.50	705.00	702.20	705.00	–	–	–	–
p7.2.l(_5%)	764.20	767.00	767.00	767.00	636.00	636.00	631.00	631.00
p7.2.m(_5%)	827.00	827.00	825.00	827.00	731.00	731.00	731.00	731.00
p7.2.n(_5%)	888.00	888.00	888.00	888.00	654.20	656.00	656.00	656.00
p7.2.o(_5%)	944.60	945.00	944.60	945.00	802.00	802.00	802.00	802.00
p7.2.p(_5%)	1001.60	1002.00	1001.50	1002.00	843.20	844.00	843.60	844.00
p7.2.q(_5%)	1042.30	1043.00	1042.70	1044.00	880.70	885.00	880.22	884.00
p7.2.r(_5%)	1093.70	1094.00	1093.00	1094.00	968.00	968.00	965.60	968.00
p7.2.s(_5%)	1134.70	1135.00	1127.80	1136.00	1001.20	1007.00	1004.50	1007.00
p7.2.t(_5%)	1172.20	1179.00	1176.30	1179.00	1073.80	1075.00	1068.80	1070.00
p7.3.a(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p7.3.b(_5%)	46.00	46.00	46.00	46.00	–	–	–	–
p7.3.c(_5%)	79.00	79.00	79.00	79.00	–	–	–	–

Continued on next page

Table A.11 — continued from previous page.

	TOP				STOP			
	FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000		FP_cuts_LNS_5000	
	Profit sum		Profit sum		Profit sum		Profit sum	
	Avg	Best	Avg	Best	Avg	Best	Avg	Best
p7.3.d(_5%)	46.80	117.00	117.00	117.00	–	–	–	–
p7.3.e(_5%)	175.00	175.00	175.00	175.00	–	–	–	–
p7.3.f(_5%)	247.00	247.00	247.00	247.00	–	–	–	–
p7.3.g(_5%)	344.00	344.00	344.00	344.00	–	–	–	–
p7.3.h(_5%)	425.00	425.00	425.00	425.00	–	–	–	–
p7.3.i(_5%)	487.00	487.00	487.00	487.00	–	–	–	–
p7.3.j(_5%)	560.40	564.00	563.20	564.00	–	–	–	–
p7.3.k(_5%)	633.00	633.00	633.00	633.00	395.00	395.00	395.00	395.00
p7.3.l(_5%)	683.00	684.00	683.20	684.00	522.00	522.00	522.00	522.00
p7.3.m(_5%)	759.40	762.00	760.90	762.00	–	–	–	–
p7.3.n(_5%)	816.40	820.00	818.30	820.00	–	–	–	–
p7.3.o(_5%)	858.80	860.00	865.00	874.00	600.70	602.00	601.30	602.00
p7.3.p(_5%)	920.50	927.00	920.30	927.00	755.90	757.00	755.90	757.00
p7.3.q(_5%)	975.10	987.00	973.80	987.00	873.20	874.00	873.80	874.00
p7.3.r(_5%)	1020.70	1021.00	1020.50	1023.00	923.00	923.00	923.00	923.00
p7.3.s(_5%)	1077.80	1081.00	1075.40	1081.00	949.00	949.00	–	–
p7.3.t(_5%)	1117.80	1120.00	1117.50	1120.00	1031.10	1034.00	1030.20	1034.00
p7.4.a(_5%)	0.00	0.00	0.00	0.00	–	–	–	–
p7.4.b(_5%)	30.00	30.00	30.00	30.00	–	–	–	–
p7.4.c(_5%)	46.00	46.00	46.00	46.00	–	–	–	–
p7.4.d(_5%)	79.00	79.00	79.00	79.00	–	–	–	–
p7.4.e(_5%)	123.00	123.00	123.00	123.00	–	–	–	–
p7.4.f(_5%)	164.00	164.00	164.00	164.00	–	–	–	–
p7.4.g(_5%)	217.00	217.00	217.00	217.00	–	–	–	–
p7.4.h(_5%)	285.00	285.00	285.00	285.00	–	–	–	–
p7.4.i(_5%)	366.00	366.00	366.00	366.00	–	–	–	–
p7.4.j(_5%)	462.00	462.00	462.00	462.00	–	–	–	–
p7.4.k(_5%)	518.40	520.00	519.20	520.00	–	–	–	–
p7.4.l(_5%)	586.70	590.00	586.70	590.00	–	–	–	–
p7.4.m(_5%)	646.00	646.00	646.00	646.00	–	–	–	–
p7.4.n(_5%)	728.10	730.00	727.70	730.00	–	–	–	–
p7.4.o(_5%)	779.90	781.00	780.50	781.00	537.00	537.00	537.00	537.00
p7.4.p(_5%)	842.80	846.00	844.90	846.00	768.30	771.00	767.40	771.00
p7.4.q(_5%)	908.10	909.00	908.40	909.00	780.80	781.00	780.60	781.00
p7.4.r(_5%)	970.00	970.00	970.00	970.00	794.29	796.00	793.00	794.00
p7.4.s(_5%)	1022.00	1022.00	1021.90	1022.00	857.00	857.00	857.00	857.00
p7.4.t(_5%)	1077.00	1077.00	1076.80	1077.00	939.00	939.00	939.00	939.00

A.6 Warm starting the cutting-plane algorithms with primal heuristics

For completeness, we tested four variations of the hybrid algorithm obtained from warm starting CPA with the solutions provided by the LNS heuristic discussed in Chapter 4 (in particular, the two best variations, namely FP_cuts_LNS_5000 and OFP_cuts_LNS_5000). Without loss of generality, an algorithm that couples the cutting-plane algorithm CPA_1 with the heuristic solution provided by FP_cuts_LNS_5000 is referred to as CPA_1+FP_cuts_LNS_5000.

In our implementation, the heuristic solution is fully and explicitly provided to CPLEX through the *MIPStart* mechanism. Additionally, we also set the paramater *IloCplex::Param::Advance* to 2, in order to guide CPLEX into re-applying its presolve considering the initial solution provided to the solver.

The summary of the results obtained for the variations of the hybrid algorithm tested are given in Tables A.12 and A.13. The first column displays the name of each instance set, and, for each variation of the hybrid algorithm, we give four result values described as follows. The first value corresponds to the number of instances solved at optimality (or to proven infeasibility) out of the complete instance set. The second one is the average wall-clock processing time (in seconds) spent in solving these instances. Note that this entry only takes into account the instances solved at optimality. The last couple of result values provides the average and the standard deviation (only over the unsolved instances in each set) of the relative optimality gaps obtained by the algorithm. These gaps are given by $\frac{UB-LB}{UB}$, where LB and UB are, respectively, the best lower and upper bounds obtained by the corresponding algorithm for a given instance within two hours of execution. If, for a given instance, no feasible solution is found within the time limit and its infeasibility is also not proven, the corresponding optimality gap is assumed to be 100%. Likewise, this gap is set to 0% whenever the instance is proven to be infeasible. The last row gives the overall results considering the complete benchmark of instances. Bold entries highlight, for each instance set, the best algorithm(s) in terms of number of instances solved to optimality.

In Tables A.14 and A.15, we display a per-instance report of the results obtained by our implementations of CPA_2+FP_cuts_LNS_5000 and CPA_2+OFP_cuts_LNS_5000 — the ones that performed batter — at solving, respectively, the original TOP instances and the new benchmark of STOP instances. For both algorithms, we report, for each instance, the best lower and upper bounds obtained within two hours of execution (columns “LB” and “UB”, respectively) and the wall-clock execution time in seconds. For the cases where neither a feasible solution was found nor the infeasibility was proven, the entries “LB” and “UB” were filled with “-inf” and “inf”, respectively. We filled with dashes the entries related to the instances that were proven to be infeasible within the time limit. In addition, we marked in bold the instances for which any of the two algorithms found optimality certificates that could not be found by CPA alone, B-B&C or any other previous exact algorithm in the literature (in the case of the TOP instances).

Table A.12: Comparison of the four variations of the hybrid algorithm tested at solving the original benchmark of TOP instances. Bold entries highlight, for each instance set, the best algorithm(s) in terms of number of instances solved to optimality.

	CPA_1+FP_cuts_LNS_5000				CPA_1+OFP_cuts_LNS_5000			
	solved		*unsolved*		*solved*		*unsolved*	
			Gap (%)				Gap (%)	
Set	#opt/total	Time (s)	Avg	StDev	#opt/total	Time (s)	Avg	StDev
1	**54/54**	1.89	–	–	**54/54**	2.04	–	–
2	**33/33**	0.28	–	–	**33/33**	0.29	–	–
3	**60/60**	94.47	–	–	**60/60**	116.43	–	–
4	**44/60**	753.55	1.66	1.31	42/60	745.34	1.69	1.54
5	**62/78**	359.44	3.01	1.70	61/78	368.47	2.84	1.71
6	**41/42**	135.24	1.37	–	**41/42**	156.92	1.50	–
7	**50/60**	650.80	2.19	0.59	49/60	630.96	2.01	0.79
Total	**344/387**	288.68	2.28	1.44	340/387	288.93	2.18	1.52

	CPA_2+FP_cuts_LNS_5000				CPA_2+OFP_cuts_LNS_5000			
	solved		*unsolved*		*solved*		*unsolved*	
			Gap (%)				Gap (%)	
Set	#opt/total	Time (s)	Avg	StDev	#opt/total	Time (s)	Avg	StDev
1	**54/54**	1.64	–	–	**54/54**	2.00	–	–
2	**33/33**	0.29	–	–	**33/33**	0.32	–	–
3	**60/60**	85.43	–	–	**60/60**	106.83	–	–
4	**46/60**	862.62	2.02	1.48	**46/60**	904.12	2.16	1.65
5	**63/78**	631.18	3.17	1.76	**63/78**	482.03	3.00	1.68
6	**42/42**	188.12	–	–	41/42	102.29	1.62	–
7	47/60	463.84	1.92	0.97	**48/60**	649.07	2.05	0.83
Total	**345/387**	331.51	2.40	1.54	**345/387**	329.95	2.42	1.49

Table A.14: Detailed results for the hybrid algorithms at solving the original TOP instances. We highlight in bold the instances for which the hybrid algorithms found optimality certificates that could not be found by CPA alone, B-B&C or any other previous exact algorithm in the literature of the TOP.

	CPA_2+FP_cuts_LNS_5000			CPA_2+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p1.2.a	0.00	0.00	0.01	0.00	0.00	0.01
p1.2.b	15.00	15.00	0.03	15.00	15.00	0.03
p1.2.c	20.00	20.00	0.06	20.00	20.00	0.06
p1.2.d	30.00	30.00	0.06	30.00	30.00	0.06
p1.2.e	45.00	45.00	0.16	45.00	45.00	0.16

Continued on next page

Table A.14 — continued from previous page.

	CPA$_2$+FP_cuts_LNS_5000			CPA$_2$+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p1.2.f	80.00	80.00	0.27	80.00	80.00	0.29
p1.2.g	90.00	90.00	0.82	90.00	90.00	1.11
p1.2.h	110.00	110.00	1.35	110.00	110.00	1.93
p1.2.i	135.00	135.00	2.15	135.00	135.00	3.15
p1.2.j	155.00	155.00	1.72	155.00	155.00	2.83
p1.2.k	175.00	175.00	1.93	175.00	175.00	3.20
p1.2.l	195.00	195.00	1.97	195.00	195.00	3.69
p1.2.m	215.00	215.00	2.60	215.00	215.00	3.89
p1.2.n	235.00	235.00	2.71	235.00	235.00	3.73
p1.2.o	240.00	240.00	3.46	240.00	240.00	4.41
p1.2.p	250.00	250.00	3.17	250.00	250.00	3.81
p1.2.q	265.00	265.00	3.32	265.00	265.00	3.87
p1.2.r	280.00	280.00	4.68	280.00	280.00	5.37
p1.3.a	0.00	0.00	0.01	0.00	0.00	0.01
p1.3.b	0.00	0.00	0.01	0.00	0.00	0.01
p1.3.c	15.00	15.00	0.04	15.00	15.00	0.04
p1.3.d	15.00	15.00	0.04	15.00	15.00	0.04
p1.3.e	30.00	30.00	0.08	30.00	30.00	0.09
p1.3.f	40.00	40.00	0.10	40.00	40.00	0.10
p1.3.g	50.00	50.00	0.04	50.00	50.00	0.21
p1.3.h	70.00	70.00	0.35	70.00	70.00	0.41
p1.3.i	105.00	105.00	0.48	105.00	105.00	0.63
p1.3.j	115.00	115.00	1.16	115.00	115.00	1.35
p1.3.k	135.00	135.00	2.62	135.00	135.00	3.02
p1.3.l	155.00	155.00	2.02	155.00	155.00	2.46
p1.3.m	175.00	175.00	1.89	175.00	175.00	2.63
p1.3.n	190.00	190.00	9.45	190.00	190.00	8.90
p1.3.o	205.00	205.00	3.59	205.00	205.00	4.37
p1.3.p	220.00	220.00	2.37	220.00	220.00	3.05
p1.3.q	230.00	230.00	6.22	230.00	230.00	6.76
p1.3.r	250.00	250.00	3.44	250.00	250.00	5.22
p1.4.a	0.00	0.00	0.01	0.00	0.00	0.01
p1.4.b	0.00	0.00	0.01	0.00	0.00	0.01
p1.4.c	0.00	0.00	0.01	0.00	0.00	0.01
p1.4.d	15.00	15.00	0.04	15.00	15.00	0.04
p1.4.e	15.00	15.00	0.04	15.00	15.00	0.04
p1.4.f	25.00	25.00	0.08	25.00	25.00	0.08
p1.4.g	35.00	35.00	0.12	35.00	35.00	0.12

Continued on next page

Table A.14 — continued from previous page.

	CPA_2+FP_cuts_LNS_5000			CPA_2+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p1.4.h	45.00	45.00	0.13	45.00	45.00	0.13
p1.4.i	60.00	60.00	0.22	60.00	60.00	0.23
p1.4.j	75.00	75.00	0.31	75.00	75.00	0.33
p1.4.k	100.00	100.00	0.50	100.00	100.00	0.55
p1.4.l	120.00	120.00	0.81	120.00	120.00	0.87
p1.4.m	130.00	130.00	3.67	130.00	130.00	3.82
p1.4.n	155.00	155.00	2.37	155.00	155.00	2.58
p1.4.o	165.00	165.00	4.54	165.00	165.00	5.58
p1.4.p	175.00	175.00	3.09	175.00	175.00	3.45
p1.4.q	190.00	190.00	4.80	190.00	190.00	5.25
p1.4.r	210.00	210.00	3.25	210.00	210.00	3.86
p2.2.a	90.00	90.00	0.11	90.00	90.00	0.11
p2.2.b	120.00	120.00	0.13	120.00	120.00	0.14
p2.2.c	140.00	140.00	0.20	140.00	140.00	0.23
p2.2.d	160.00	160.00	0.19	160.00	160.00	0.32
p2.2.e	190.00	190.00	0.45	190.00	190.00	0.49
p2.2.f	200.00	200.00	0.27	200.00	200.00	0.27
p2.2.g	200.00	200.00	0.28	200.00	200.00	0.32
p2.2.h	230.00	230.00	0.37	230.00	230.00	0.42
p2.2.i	230.00	230.00	0.37	230.00	230.00	0.43
p2.2.j	260.00	260.00	0.40	260.00	260.00	0.44
p2.2.k	275.00	275.00	3.13	275.00	275.00	3.27
p2.3.a	70.00	70.00	0.06	70.00	70.00	0.07
p2.3.b	70.00	70.00	0.08	70.00	70.00	0.08
p2.3.c	105.00	105.00	0.02	105.00	105.00	0.14
p2.3.d	105.00	105.00	0.13	105.00	105.00	0.13
p2.3.e	120.00	120.00	0.15	120.00	120.00	0.18
p2.3.f	120.00	120.00	0.14	120.00	120.00	0.16
p2.3.g	145.00	145.00	0.30	145.00	145.00	0.32
p2.3.h	165.00	165.00	0.28	165.00	165.00	0.32
p2.3.i	200.00	200.00	0.40	200.00	200.00	0.45
p2.3.j	200.00	200.00	0.37	200.00	200.00	0.44
p2.3.k	200.00	200.00	0.29	200.00	200.00	0.34
p2.4.a	10.00	10.00	0.01	10.00	10.00	0.01
p2.4.b	70.00	70.00	0.07	70.00	70.00	0.08
p2.4.c	70.00	70.00	0.09	70.00	70.00	0.09
p2.4.d	70.00	70.00	0.08	70.00	70.00	0.08

Continued on next page

Table A.14 — continued from previous page.

	CPA_2+FP_cuts_LNS_5000			CPA_2+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p2.4.e	70.00	70.00	0.08	70.00	70.00	0.08
p2.4.f	105.00	105.00	0.15	105.00	105.00	0.15
p2.4.g	105.00	105.00	0.14	105.00	105.00	0.16
p2.4.h	120.00	120.00	0.16	120.00	120.00	0.18
p2.4.i	120.00	120.00	0.17	120.00	120.00	0.19
p2.4.j	120.00	120.00	0.15	120.00	120.00	0.17
p2.4.k	180.00	180.00	0.32	180.00	180.00	0.32
p3.2.a	90.00	90.00	0.10	90.00	90.00	0.13
p3.2.b	150.00	150.00	0.18	150.00	150.00	0.19
p3.2.c	180.00	180.00	0.49	180.00	180.00	0.54
p3.2.d	220.00	220.00	0.73	220.00	220.00	0.85
p3.2.e	260.00	260.00	1.21	260.00	260.00	1.29
p3.2.f	300.00	300.00	6.88	300.00	300.00	4.84
p3.2.g	360.00	360.00	2.83	360.00	360.00	4.06
p3.2.h	410.00	410.00	4.01	410.00	410.00	4.87
p3.2.i	460.00	460.00	7.56	460.00	460.00	8.34
p3.2.j	510.00	510.00	11.38	510.00	510.00	12.26
p3.2.k	550.00	550.00	10.28	550.00	550.00	11.41
p3.2.l	590.00	590.00	2.88	590.00	590.00	4.07
p3.2.m	620.00	620.00	5.65	620.00	620.00	6.52
p3.2.n	660.00	660.00	3.79	660.00	660.00	4.85
p3.2.o	690.00	690.00	3.60	690.00	690.00	4.63
p3.2.p	720.00	720.00	4.65	720.00	720.00	5.40
p3.2.q	760.00	760.00	3.68	760.00	760.00	5.13
p3.2.r	790.00	790.00	6.60	790.00	790.00	7.53
p3.2.s	800.00	800.00	0.58	800.00	800.00	1.25
p3.2.t	800.00	800.00	3.77	800.00	800.00	5.12
p3.3.a	30.00	30.00	0.05	30.00	30.00	0.06
p3.3.b	90.00	90.00	0.02	90.00	90.00	0.11
p3.3.c	120.00	120.00	0.15	120.00	120.00	0.16
p3.3.d	170.00	170.00	0.27	170.00	170.00	0.26
p3.3.e	200.00	200.00	0.33	200.00	200.00	0.33
p3.3.f	230.00	230.00	1.01	230.00	230.00	1.13
p3.3.g	270.00	270.00	1.12	270.00	270.00	1.23
p3.3.h	300.00	300.00	1.99	300.00	300.00	2.05
p3.3.i	330.00	330.00	34.32	330.00	330.00	22.52
p3.3.j	380.00	380.00	20.96	380.00	380.00	18.43

Continued on next page

Table A.14 — continued from previous page.

	CPA$_2$+FP_cuts_LNS_5000			CPA$_2$+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p3.3.k	440.00	440.00	26.30	440.00	440.00	26.77
p3.3.l	480.00	480.00	58.23	480.00	480.00	151.02
p3.3.m	520.00	520.00	90.32	520.00	520.00	77.86
p3.3.n	570.00	570.00	12.26	570.00	570.00	13.29
p3.3.o	590.00	590.00	700.73	590.00	590.00	319.75
p3.3.p	640.00	640.00	1810.30	640.00	640.00	1901.46
p3.3.q	680.00	680.00	4.48	680.00	680.00	5.05
p3.3.r	710.00	710.00	2.94	710.00	710.00	3.77
p3.3.s	720.00	720.00	41.69	720.00	720.00	27.14
p3.3.t	760.00	760.00	130.54	760.00	760.00	45.73
p3.4.a	20.00	20.00	0.05	20.00	20.00	0.04
p3.4.b	30.00	30.00	0.06	30.00	30.00	0.06
p3.4.c	90.00	90.00	0.11	90.00	90.00	0.11
p3.4.d	100.00	100.00	0.12	100.00	100.00	0.14
p3.4.e	140.00	140.00	0.21	140.00	140.00	0.22
p3.4.f	190.00	190.00	0.32	190.00	190.00	0.33
p3.4.g	220.00	220.00	0.39	220.00	220.00	0.39
p3.4.h	240.00	240.00	0.81	240.00	240.00	0.84
p3.4.i	270.00	270.00	0.87	270.00	270.00	0.89
p3.4.j	310.00	310.00	1.43	310.00	310.00	1.54
p3.4.k	350.00	350.00	1.07	350.00	350.00	1.17
p3.4.l	380.00	380.00	1.32	380.00	380.00	1.46
p3.4.m	390.00	390.00	513.83	390.00	390.00	856.31
p3.4.n	440.00	440.00	71.30	440.00	440.00	113.65
p3.4.o	500.00	500.00	25.04	500.00	500.00	25.73
p3.4.p	560.00	560.00	2.80	560.00	560.00	3.07
p3.4.q	560.00	560.00	1451.07	560.00	560.00	2656.80
p3.4.r	600.00	600.00	31.56	600.00	600.00	29.44
p3.4.s	670.00	670.00	2.36	670.00	670.00	3.06
p3.4.t	670.00	670.00	2.51	670.00	670.00	2.95
p4.2.a	206.00	206.00	1.12	206.00	206.00	1.31
p4.2.b	341.00	341.00	4.69	341.00	341.00	7.95
p4.2.c	452.00	452.00	10.84	452.00	452.00	20.89
p4.2.d	531.00	531.00	61.24	531.00	531.00	105.83
p4.2.e	618.00	618.00	61.46	618.00	618.00	109.77
p4.2.f	687.00	687.00	337.32	687.00	687.00	387.79
p4.2.g	757.00	757.00	172.49	757.00	757.00	221.99

Continued on next page

Table A.14 — continued from previous page.

	CPA$_2$+FP_cuts_LNS_5000			CPA$_2$+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p4.2.h	835.00	835.00	146.30	835.00	835.00	199.12
p4.2.i	918.00	918.00	152.43	918.00	918.00	225.00
p4.2.j	965.00	965.00	195.68	965.00	965.00	287.35
p4.2.k	1022.00	1022.00	280.31	1022.00	1022.00	372.53
p4.2.l	1074.00	1074.00	368.99	1074.00	1074.00	486.28
p4.2.m	1132.00	1132.00	231.68	1132.00	1132.00	305.02
p4.2.n	1174.00	1174.00	289.82	1174.00	1174.00	351.89
p4.2.o	1218.00	1218.00	198.46	1218.00	1218.00	270.53
p4.2.p	1242.00	1242.00	1081.80	1242.00	1242.00	1031.72
p4.2.q	1268.00	1268.00	2884.50	1268.00	1268.00	2653.58
p4.2.r	1292.00	1292.00	621.60	1292.00	1292.00	832.86
p4.2.s	1304.00	1304.00	5657.63	1304.00	1304.00	6057.61
p4.2.t	1306.00	1306.00	14.52	1306.00	1306.00	300.14
p4.3.a	0.00	0.00	0.03	0.00	0.00	0.03
p4.3.b	38.00	38.00	0.13	38.00	38.00	0.13
p4.3.c	193.00	193.00	1.22	193.00	193.00	1.25
p4.3.d	335.00	335.00	12.23	335.00	335.00	13.29
p4.3.e	468.00	468.00	15.24	468.00	468.00	20.83
p4.3.f	579.00	579.00	13.02	579.00	579.00	19.75
p4.3.g	653.00	653.00	106.99	653.00	653.00	217.72
p4.3.h	729.00	729.00	496.03	729.00	729.00	567.03
p4.3.i	809.00	809.00	113.45	809.00	809.00	147.91
p4.3.j	861.00	861.00	4335.73	861.00	861.00	6602.74
p4.3.k	919.00	924.59	7200.00	919.00	930.13	7200.00
p4.3.l	970.00	1007.88	7200.00	970.00	1007.90	7200.00
p4.3.m	1053.00	1070.99	7200.00	1063.00	1065.69	7200.00
p4.3.n	1121.00	1121.00	717.11	1121.00	1121.00	1078.36
p4.3.o	1172.00	1172.00	5700.81	1172.00	1172.00	5759.42
p4.3.p	1222.00	1222.00	700.18	1222.00	1222.00	762.96
p4.3.q	1253.00	1253.00	979.20	1253.00	1253.00	1011.26
p4.3.r	1269.00	1285.32	7200.00	1271.00	1285.00	7200.00
p4.3.s	1295.00	1297.65	7200.00	1295.00	1299.09	7200.00
p4.3.t	1304.00	1306.00	7200.00	1304.00	1306.00	7200.00
p4.4.a	0.00	0.00	0.03	0.00	0.00	0.04
p4.4.b	0.00	0.00	0.03	0.00	0.00	0.03
p4.4.c	0.00	0.00	0.03	0.00	0.00	0.04
p4.4.d	38.00	38.00	0.14	38.00	38.00	0.14
p4.4.e	183.00	183.00	0.83	183.00	183.00	0.85

Continued on next page

Table A.14 — continued from previous page.

	CPA_2+FP_cuts_LNS_5000			CPA_2+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p4.4.f	324.00	324.00	3.12	324.00	324.00	3.31
p4.4.g	461.00	461.00	17.51	461.00	461.00	18.11
p4.4.h	571.00	571.00	106.47	571.00	571.00	107.01
p4.4.i	657.00	657.00	688.39	657.00	657.00	695.64
p4.4.j	732.00	732.00	5945.60	732.00	732.00	5894.22
p4.4.k	821.00	821.00	2401.94	821.00	821.00	2401.83
p4.4.l	880.00	880.00	4551.98	880.00	880.00	2036.31
p4.4.m	919.00	942.65	7200.00	912.00	944.29	7200.00
p4.4.n	964.00	1015.06	7200.00	968.00	1009.94	7200.00
p4.4.o	1061.00	1085.52	7200.00	1037.00	1092.69	7200.00
p4.4.p	1124.00	1138.82	7200.00	1124.00	1138.81	7200.00
p4.4.q	1161.00	1201.71	7200.00	1161.00	1205.43	7200.00
p4.4.r	1202.00	1248.25	7200.00	1209.00	1248.70	7200.00
p4.4.s	1257.00	1276.44	7200.00	1257.00	1276.44	7200.00
p4.4.t	1285.00	1296.46	7200.00	1282.00	1297.38	7200.00
p5.2.a	0.00	0.00	0.02	0.00	0.00	0.02
p5.2.b	20.00	20.00	0.12	20.00	20.00	0.12
p5.2.c	50.00	50.00	0.24	50.00	50.00	0.24
p5.2.d	80.00	80.00	0.95	80.00	80.00	1.03
p5.2.e	180.00	180.00	1.27	180.00	180.00	1.32
p5.2.f	240.00	240.00	1.44	240.00	240.00	3.00
p5.2.g	320.00	320.00	19.80	320.00	320.00	25.92
p5.2.h	410.00	410.00	7.47	410.00	410.00	11.19
p5.2.i	480.00	480.00	129.62	480.00	480.00	117.75
p5.2.j	580.00	580.00	37.68	580.00	580.00	37.69
p5.2.k	670.00	670.00	210.74	670.00	670.00	185.12
p5.2.l	800.00	800.00	6.73	800.00	800.00	13.42
p5.2.m	860.00	860.00	16.67	860.00	860.00	30.05
p5.2.n	925.00	925.00	238.35	925.00	925.00	321.84
p5.2.o	1020.00	1020.00	83.54	1020.00	1020.00	94.78
p5.2.p	1150.00	1150.00	8.72	1150.00	1150.00	13.80
p5.2.q	1195.00	1195.00	20.83	1195.00	1195.00	27.75
p5.2.r	1260.00	1260.00	57.56	1260.00	1260.00	67.31
p5.2.s	1340.00	1340.00	17.80	1340.00	1340.00	27.92
p5.2.t	1400.00	1400.00	18.58	1400.00	1400.00	31.26
p5.2.u	1460.00	1460.00	24.77	1460.00	1460.00	34.25
p5.2.v	1505.00	1505.00	746.92	1505.00	1505.00	784.16

Continued on next page

Table A.14 — continued from previous page.

	CPA$_2$+FP_cuts_LNS_5000			CPA$_2$+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p5.2.w	1565.00	1565.00	318.34	1565.00	1565.00	267.03
p5.2.x	1610.00	1610.00	26.58	1610.00	1610.00	37.12
p5.2.y	1645.00	1645.00	62.07	1645.00	1645.00	76.55
p5.2.z	1680.00	1680.00	5.26	1680.00	1680.00	16.37
p5.3.a	0.00	0.00	0.02	0.00	0.00	0.02
p5.3.b	15.00	15.00	0.07	15.00	15.00	0.08
p5.3.c	20.00	20.00	0.13	20.00	20.00	0.13
p5.3.d	60.00	60.00	0.23	60.00	60.00	0.23
p5.3.e	95.00	95.00	0.40	95.00	95.00	0.39
p5.3.f	110.00	110.00	8.61	110.00	110.00	10.60
p5.3.g	185.00	185.00	1.41	185.00	185.00	1.53
p5.3.h	260.00	260.00	13.17	260.00	260.00	13.70
p5.3.i	335.00	335.00	5.91	335.00	335.00	9.76
p5.3.j	470.00	470.00	8.13	470.00	470.00	12.70
p5.3.k	495.00	495.00	2466.97	495.00	495.00	2473.85
p5.3.l	595.00	595.00	688.93	595.00	595.00	588.46
p5.3.m	650.00	700.14	7200.00	650.00	697.79	7200.00
p5.3.n	755.00	755.00	5766.31	755.00	755.00	2495.43
p5.3.o	870.00	870.00	343.12	870.00	870.00	217.33
p5.3.p	990.00	990.00	38.63	990.00	990.00	41.90
p5.3.q	1070.00	1070.00	58.77	1070.00	1070.00	80.16
p5.3.r	1125.00	1125.00	6641.90	1125.00	1125.00	3169.01
p5.3.s	1190.00	1222.22	7200.00	1190.00	1220.94	7200.00
p5.3.t	1260.00	1304.99	7200.00	1260.00	1304.11	7200.00
p5.3.u	1345.00	1374.86	7200.00	1345.00	1375.65	7200.00
5.3.v	1425.00	1425.00	6148.33	1425.00	1425.00	4736.31
p5.3.w	1485.00	1503.76	7200.00	1485.00	1501.81	7200.00
p5.3.x	1555.00	1555.00	3288.86	1555.00	1555.00	4846.99
p5.3.y	1590.00	1620.74	7200.00	1590.00	1621.61	7200.00
p5.3.z	1635.00	1659.89	7200.00	1635.00	1655.76	7200.00
p5.4.a	0.00	0.00	0.02	0.00	0.00	0.03
p5.4.b	0.00	0.00	0.02	0.00	0.00	0.03
p5.4.c	20.00	20.00	0.09	20.00	20.00	0.09
p5.4.d	20.00	20.00	0.14	20.00	20.00	0.14
p5.4.e	20.00	20.00	0.15	20.00	20.00	0.15
p5.4.f	80.00	80.00	0.40	80.00	80.00	0.54
p5.4.g	140.00	140.00	0.62	140.00	140.00	0.62
p5.4.h	140.00	140.00	9.00	140.00	140.00	8.77

Continued on next page

Table A.14 — continued from previous page.

	CPA$_2$+FP_cuts_LNS_5000			CPA$_2$+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p5.4.i	240.00	240.00	0.99	240.00	240.00	1.02
p5.4.j	340.00	340.00	1.37	340.00	340.00	1.48
p5.4.k	340.00	340.00	2984.86	340.00	340.00	1614.77
p5.4.l	430.00	430.00	34.75	430.00	430.00	52.87
p5.4.m	555.00	555.00	21.42	555.00	555.00	31.14
p5.4.n	620.00	620.00	3994.26	620.00	620.00	4069.32
p5.4.o	690.00	729.14	7200.00	690.00	714.93	7200.00
p5.4.p	765.00	811.87	7200.00	765.00	816.00	7200.00
p5.4.q	860.00	886.50	7200.00	860.00	889.62	7200.00
p5.4.r	960.00	960.00	4917.73	960.00	960.00	3421.38
p5.4.s	1025.00	1075.09	7200.00	1030.00	1075.59	7200.00
p5.4.t	1160.00	1160.00	214.58	1160.00	1160.00	187.89
p5.4.u	1300.00	1300.00	10.35	1300.00	1300.00	15.05
p5.4.v	1320.00	1350.17	7200.00	1320.00	1346.55	7200.00
p5.4.w	1390.00	1422.61	7200.00	1390.00	1425.03	7200.00
p5.4.x	1450.00	1485.68	7200.00	1450.00	1485.46	7200.00
p5.4.y	1520.00	1546.65	7200.00	1520.00	1544.33	7200.00
p5.4.z	1620.00	1620.00	31.49	1620.00	1620.00	36.78
p6.2.a	0.00	0.00	0.02	0.00	0.00	0.02
p6.2.b	0.00	0.00	0.02	0.00	0.00	0.02
p6.2.c	0.00	0.00	0.02	0.00	0.00	0.02
p6.2.d	192.00	192.00	0.31	192.00	192.00	1.05
p6.2.e	360.00	360.00	4.75	360.00	360.00	6.77
p6.2.f	588.00	588.00	2.13	588.00	588.00	3.63
p6.2.g	660.00	660.00	43.89	660.00	660.00	57.22
p6.2.h	780.00	780.00	16.92	780.00	780.00	24.37
p6.2.i	888.00	888.00	9.96	888.00	888.00	17.30
p6.2.j	948.00	948.00	886.84	948.00	948.00	919.44
p6.2.k	1032.00	1032.00	1861.95	1032.00	1032.00	1393.93
p6.2.l	1116.00	1116.00	207.55	1116.00	1116.00	301.55
p6.2.m	1188.00	1188.00	34.46	1188.00	1188.00	53.48
p6.2.n	1260.00	1260.00	22.03	1260.00	1260.00	28.02
p6.3.a	0.00	0.00	0.02	0.00	0.00	0.02
p6.3.b	0.00	0.00	0.02	0.00	0.00	0.02
p6.3.c	0.00	0.00	0.02	0.00	0.00	0.02
p6.3.d	0.00	0.00	0.02	0.00	0.00	0.02
p6.3.e	0.00	0.00	0.02	0.00	0.00	0.02

Continued on next page

Table A.14 — continued from previous page.

	CPA$_2$+FP_cuts_LNS_5000			CPA$_2$+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p6.3.f	0.00	0.00	0.02	0.00	0.00	0.02
p6.3.g	282.00	282.00	1.52	282.00	282.00	1.66
p6.3.h	444.00	444.00	45.52	444.00	444.00	42.01
p6.3.i	642.00	642.00	4.80	642.00	642.00	7.44
p6.3.j	828.00	828.00	4.35	828.00	828.00	8.87
p6.3.k	894.00	894.00	1465.15	894.00	894.00	575.88
p6.3.l	1002.00	1002.00	153.87	1002.00	1002.00	96.60
p6.3.m	1080.00	1080.00	2453.30	1080.00	1097.75	7200.00
p6.3.n	1170.00	1170.00	507.67	1170.00	1170.00	491.61
p6.4.a	0.00	0.00	0.02	0.00	0.00	0.02
p6.4.b	0.00	0.00	0.02	0.00	0.00	0.02
p6.4.c	0.00	0.00	0.02	0.00	0.00	0.02
p6.4.d	0.00	0.00	0.02	0.00	0.00	0.02
p6.4.e	0.00	0.00	0.02	0.00	0.00	0.02
p6.4.f	0.00	0.00	0.02	0.00	0.00	0.02
p6.4.g	0.00	0.00	0.02	0.00	0.00	0.02
p6.4.h	0.00	0.00	0.02	0.00	0.00	0.02
p6.4.i	0.00	0.00	0.02	0.00	0.00	0.02
p6.4.j	366.00	366.00	2.86	366.00	366.00	2.98
p6.4.k	528.00	528.00	25.67	528.00	528.00	26.60
p6.4.l	696.00	696.00	125.87	696.00	696.00	109.19
p6.4.m	912.00	912.00	8.19	912.00	912.00	11.37
p6.4.n	1068.00	1068.00	10.97	1068.00	1068.00	12.74
p7.2.a	30.00	30.00	0.08	30.00	30.00	0.08
p7.2.b	64.00	64.00	0.16	64.00	64.00	0.15
p7.2.c	101.00	101.00	0.17	101.00	101.00	0.43
p7.2.d	190.00	190.00	1.64	190.00	190.00	1.95
p7.2.e	290.00	290.00	6.68	290.00	290.00	9.71
p7.2.f	387.00	387.00	10.99	387.00	387.00	18.81
p7.2.g	459.00	459.00	30.29	459.00	459.00	58.19
p7.2.h	521.00	521.00	110.72	521.00	521.00	211.42
p7.2.i	580.00	580.00	214.45	580.00	580.00	437.79
p7.2.j	646.00	646.00	388.79	646.00	646.00	741.24
p7.2.k	705.00	705.00	464.80	705.00	705.00	621.57
p7.2.l	767.00	767.00	690.75	767.00	767.00	853.83
p7.2.m	827.00	827.00	654.31	827.00	827.00	849.49
p7.2.n	888.00	888.00	545.74	888.00	888.00	658.61

Continued on next page

Table A.14 — continued from previous page.

	CPA_2+FP_cuts_LNS_5000			CPA_2+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p7.2.o	945.00	945.00	516.78	945.00	945.00	602.49
p7.2.p	1002.00	1002.00	370.58	1002.00	1002.00	423.40
p7.2.q	1044.00	1044.00	940.74	1044.00	1044.00	802.17
p7.2.r	1094.00	1094.00	383.07	1094.00	1094.00	437.08
p7.2.s	1136.00	1136.00	346.38	1136.00	1136.00	508.65
p7.2.t	1179.00	1179.00	338.49	1179.00	1179.00	393.41
p7.3.a	0.00	0.00	0.04	0.00	0.00	0.04
p7.3.b	46.00	46.00	0.11	46.00	46.00	0.11
p7.3.c	79.00	79.00	0.21	79.00	79.00	0.21
p7.3.d	117.00	117.00	0.09	117.00	117.00	0.43
p7.3.e	175.00	175.00	1.41	175.00	175.00	1.39
p7.3.f	247.00	247.00	9.93	247.00	247.00	6.88
p7.3.g	344.00	344.00	14.95	344.00	344.00	22.57
p7.3.h	425.00	425.00	389.16	425.00	425.00	361.49
p7.3.i	487.00	487.00	1461.20	487.00	487.00	1703.66
p7.3.j	564.00	564.00	185.12	564.00	564.00	405.45
p7.3.k	633.00	633.00	142.20	633.00	633.00	234.01
p7.3.l	684.00	684.00	6248.22	684.00	684.00	6435.07
p7.3.m	762.00	762.00	454.50	762.00	762.00	467.12
p7.3.n	820.00	820.00	501.09	820.00	820.00	601.16
p7.3.o	874.00	874.00	1077.76	874.00	874.00	1011.32
p7.3.p	929.00	929.00	4574.07	929.00	929.00	4776.17
p7.3.q	987.00	999.03	7200.00	987.00	996.06	7200.00
p7.3.r	1021.00	1056.01	7200.00	1021.00	1056.32	7200.00
p7.3.s	1081.00	1095.49	7200.00	1081.00	1096.73	7200.00
p7.3.t	1120.00	1136.32	7200.00	1118.00	1137.38	7200.00
p7.4.a	0.00	0.00	0.04	0.00	0.00	0.04
p7.4.b	30.00	30.00	0.10	30.00	30.00	0.10
p7.4.c	46.00	46.00	0.15	46.00	46.00	0.15
p7.4.d	79.00	79.00	0.24	79.00	79.00	0.24
p7.4.e	123.00	123.00	0.41	123.00	123.00	0.40
p7.4.f	164.00	164.00	0.84	164.00	164.00	0.89
p7.4.g	217.00	217.00	3.74	217.00	217.00	4.97
p7.4.h	285.00	285.00	30.40	285.00	285.00	30.57
p7.4.i	366.00	366.00	11.24	366.00	366.00	9.66
p7.4.j	462.00	462.00	362.21	462.00	462.00	280.79
p7.4.k	520.00	532.13	7200.00	520.00	532.94	7200.00
p7.4.l	590.00	592.87	7200.00	590.00	595.67	7200.00

Continued on next page

Table A.14 — continued from previous page.

	CPA$_2$+FP_cuts_LNS_5000			CPA$_2$+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p7.4.m	646.00	671.79	7200.00	646.00	670.31	7200.00
p7.4.n	730.00	730.00	315.59	730.00	730.00	656.64
p7.4.o	781.00	792.77	7200.00	781.00	794.69	7200.00
p7.4.p	846.00	865.20	7200.00	846.00	863.45	7200.00
p7.4.q	909.00	931.53	7200.00	909.00	930.48	7200.00
p7.4.r	970.00	994.12	7200.00	970.00	991.84	7200.00
p7.4.s	1022.00	1041.97	7200.00	1022.00	1041.91	7200.00
p7.4.t	1077.00	1084.08	7200.00	1077.00	1077.00	6513.17

Table A.15: Detailed results for the hybrid algorithms at solving the new benchmark of STOP instances. We highlight in bold the instances for which the hybrid algorithms found optimality certificates that could not be found by CPA alone and B-B&C.

	CPA$_2$+FP_cuts_LNS_5000			CPA$_2$+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p1.2.a_5%	–	–	0.00	–	–	0.00
p1.2.b_5%	–	–	0.00	–	–	0.00
p1.2.c_5%	–	–	0.00	–	–	0.00
p1.2.d_5%	15.00	15.00	0.05	15.00	15.00	0.04
p1.2.e_5%	–	–	0.00	–	–	0.00
p1.2.f_5%	–	–	0.00	–	–	0.00
p1.2.g_5%	–	–	0.00	–	–	0.00
p1.2.h_5%	45.00	45.00	0.63	45.00	45.00	1.06
p1.2.i_5%	110.00	110.00	2.16	110.00	110.00	2.92
p1.2.j_5%	140.00	140.00	1.71	140.00	140.00	2.64
p1.2.k_5%	150.00	150.00	2.15	150.00	150.00	3.19
p1.2.l_5%	185.00	185.00	2.26	185.00	185.00	4.27
p1.2.m_5%	190.00	190.00	2.30	190.00	190.00	3.70
p1.2.n_5%	215.00	215.00	2.41	215.00	215.00	3.23
p1.2.o_5%	220.00	220.00	2.71	220.00	220.00	3.89
p1.2.p_5%	225.00	225.00	2.85	225.00	225.00	3.73
p1.2.q_5%	245.00	245.00	3.17	245.00	245.00	3.73
p1.2.r_5%	265.00	265.00	4.16	265.00	265.00	4.72
p1.3.a_5%	–	–	0.00	–	–	0.00
p1.3.b_5%	–	–	0.00	–	–	0.00
p1.3.c_5%	–	–	0.00	–	–	0.00

Continued on next page

Table A.15 — continued from previous page.

	CPA_2+FP_cuts_LNS_5000			CPA_2+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p1.3.d_5%	–	–	0.00	–	–	0.00
p1.3.e_5%	–	–	0.00	–	–	0.00
p1.3.f_5%	–	–	0.00	–	–	0.00
p1.3.g_5%	–	–	0.00	–	–	0.00
p1.3.h_5%	–	–	0.00	–	–	0.00
p1.3.i_5%	90.00	90.00	0.44	90.00	90.00	0.51
p1.3.j_5%	80.00	80.00	1.03	80.00	80.00	1.21
p1.3.k_5%	115.00	115.00	1.52	115.00	115.00	1.78
p1.3.l_5%	140.00	140.00	1.71	140.00	140.00	2.34
p1.3.m_5%	130.00	130.00	42.71	130.00	130.00	43.11
p1.3.n_5%	165.00	165.00	6.63	165.00	165.00	7.11
p1.3.o_5%	180.00	180.00	2.65	180.00	180.00	3.52
p1.3.p_5%	200.00	200.00	2.18	200.00	200.00	2.86
p1.3.q_5%	210.00	210.00	8.73	210.00	210.00	6.62
p1.3.r_5%	225.00	225.00	3.06	225.00	225.00	4.22
p1.4.a_5%	–	–	0.00	–	–	0.00
p1.4.b_5%	–	–	0.00	–	–	0.00
p1.4.c_5%	–	–	0.00	–	–	0.00
p1.4.d_5%	–	–	0.00	–	–	0.00
p1.4.e_5%	–	–	0.00	–	–	0.00
p1.4.f_5%	–	–	0.00	–	–	0.00
p1.4.g_5%	–	–	0.00	–	–	0.00
p1.4.h_5%	–	–	0.00	–	–	0.00
p1.4.i_5%	–	–	0.00	–	–	0.00
p1.4.j_5%	–	–	0.00	–	–	0.00
p1.4.k_5%	–	–	0.00	–	–	0.00
p1.4.l_5%	–	–	0.00	–	–	0.00
p1.4.m_5%	115.00	115.00	3.99	115.00	115.00	4.06
p1.4.n_5%	135.00	135.00	2.02	135.00	135.00	2.07
p1.4.o_5%	–	–	0.00	–	–	0.00
p1.4.p_5%	150.00	150.00	5.36	150.00	150.00	5.38
p1.4.q_5%	165.00	165.00	5.70	165.00	165.00	6.06
p1.4.r_5%	195.00	195.00	3.89	195.00	195.00	4.63
p2.2.a_5%	–	–	0.00	–	–	0.00
p2.2.b_5%	–	–	0.00	–	–	0.00
p2.2.c_5%	–	–	0.00	–	–	0.00
p2.2.d_5%	–	–	0.00	–	–	0.00

Continued on next page

Table A.15 — continued from previous page.

	CPA$_2$+FP_cuts_LNS_5000			CPA$_2$+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p2.2.e_5%	–	–	0.00	–	–	0.00
p2.2.f_5%	–	–	0.00	–	–	0.00
p2.2.g_5%	–	–	0.00	–	–	0.00
p2.2.h_5%	200.00	200.00	0.41	200.00	200.00	0.47
p2.2.i_5%	–	–	0.00	–	–	0.00
p2.2.j_5%	220.00	220.00	0.34	220.00	220.00	0.41
p2.2.k_5%	210.00	210.00	0.28	210.00	210.00	0.42
p2.3.a_5%	–	–	0.00	–	–	0.00
p2.3.b_5%	–	–	0.00	–	–	0.00
p2.3.c_5%	–	–	0.00	–	–	0.00
p2.3.d_5%	–	–	0.00	–	–	0.00
p2.3.e_5%	–	–	0.00	–	–	0.00
p2.3.f_5%	–	–	0.00	–	–	0.00
p2.3.g_5%	–	–	0.00	–	–	0.00
p2.3.h_5%	–	–	0.00	–	–	0.00
p2.3.i_5%	170.00	170.00	0.39	170.00	170.00	0.45
p2.3.j_5%	160.00	160.00	0.35	160.00	160.00	0.39
p2.3.k_5%	170.00	170.00	0.28	170.00	170.00	0.33
p2.4.a_5%	–	–	0.00	–	–	0.00
p2.4.b_5%	–	–	0.00	–	–	0.00
p2.4.c_5%	–	–	0.00	–	–	0.00
p2.4.d_5%	–	–	0.00	–	–	0.00
p2.4.e_5%	–	–	0.00	–	–	0.00
p2.4.f_5%	–	–	0.00	–	–	0.00
p2.4.g_5%	–	–	0.00	–	–	0.00
p2.4.h_5%	95.00	95.00	0.14	95.00	95.00	0.16
p2.4.i_5%	–	–	0.00	–	–	0.00
p2.4.j_5%	–	–	0.00	–	–	0.00
p2.4.k_5%	145.00	145.00	0.25	145.00	145.00	0.26
p3.2.a_5%	–	–	0.00	–	–	0.00
p3.2.b_5%	–	–	0.00	–	–	0.00
p3.2.c_5%	–	–	0.00	–	–	0.00
p3.2.d_5%	–	–	0.00	–	–	0.00
p3.2.e_5%	220.00	220.00	1.94	220.00	220.00	2.24
p3.2.f_5%	–	–	0.00	–	–	0.00
p3.2.g_5%	290.00	290.00	9.18	290.00	290.00	9.70
p3.2.h_5%	360.00	360.00	1.86	360.00	360.00	2.54

Continued on next page

Table A.15 — continued from previous page.

	CPA$_2$+FP_cuts_LNS_5000			CPA$_2$+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p3.2.i_5%	400.00	400.00	7.53	400.00	400.00	8.36
p3.2.j_5%	470.00	470.00	11.19	470.00	470.00	12.03
p3.2.k_5%	510.00	510.00	13.87	510.00	510.00	14.88
p3.2.l_5%	540.00	540.00	2.90	540.00	540.00	3.92
p3.2.m_5%	560.00	560.00	5.41	560.00	560.00	6.57
p3.2.n_5%	610.00	610.00	4.41	610.00	610.00	4.99
p3.2.o_5%	660.00	660.00	3.32	660.00	660.00	4.30
p3.2.p_5%	690.00	690.00	5.70	690.00	690.00	6.95
p3.2.q_5%	710.00	710.00	4.59	710.00	710.00	5.53
p3.2.r_5%	740.00	740.00	6.73	740.00	740.00	7.48
p3.2.s_5%	730.00	730.00	0.31	730.00	730.00	4.33
p3.2.t_5%	700.00	700.00	3.87	700.00	700.00	4.93
p3.3.a_5%	–	–	0.00	–	–	0.00
p3.3.b_5%	–	–	0.00	–	–	0.00
p3.3.c_5%	–	–	0.00	–	–	0.00
p3.3.d_5%	150.00	150.00	0.24	150.00	150.00	0.24
p3.3.e_5%	–	–	0.00	–	–	0.00
p3.3.f_5%	160.00	160.00	0.84	160.00	160.00	0.85
p3.3.g_5%	–	–	0.00	–	–	0.00
p3.3.h_5%	–	–	0.00	–	–	0.00
p3.3.i_5%	–	–	0.00	–	–	0.00
p3.3.j_5%	360.00	360.00	15.62	360.00	360.00	15.92
p3.3.k_5%	400.00	400.00	26.99	400.00	400.00	26.67
p3.3.l_5%	–	–	0.00	–	–	0.00
p3.3.m_5%	470.00	470.00	36.50	470.00	470.00	77.54
p3.3.n_5%	500.00	500.00	16.68	500.00	500.00	17.40
p3.3.o_5%	560.00	560.00	337.80	560.00	560.00	334.96
p3.3.p_5%	570.00	570.00	3587.90	570.00	570.00	3203.12
p3.3.q_5%	650.00	650.00	6.47	650.00	650.00	6.96
p3.3.r_5%	660.00	660.00	3.36	660.00	660.00	4.18
p3.3.s_5%	670.00	670.00	22.76	670.00	670.00	29.09
p3.3.t_5%	700.00	700.00	41.60	700.00	700.00	42.93
p3.4.a_5%	–	–	0.00	–	–	0.00
p3.4.b_5%	–	–	0.00	–	–	0.00
p3.4.c_5%	–	–	0.00	–	–	0.00
p3.4.d_5%	–	–	0.00	–	–	0.00
p3.4.e_5%	–	–	0.00	–	–	0.00
p3.4.f_5%	–	–	0.00	–	–	0.00

Continued on next page

Table A.15 — continued from previous page.

	CPA$_2$+FP_cuts_LNS_5000			CPA$_2$+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p3.4.g_5%	190.00	190.00	0.35	190.00	190.00	0.37
p3.4.h_5%	220.00	220.00	0.66	220.00	220.00	0.69
p3.4.i_5%	230.00	230.00	0.94	230.00	230.00	1.01
p3.4.j_5%	–	–	0.00	–	–	0.00
p3.4.k_5%	280.00	280.00	3.06	280.00	280.00	3.33
p3.4.l_5%	–	–	0.00	–	–	0.00
p3.4.m_5%	340.00	340.00	557.48	340.00	340.00	474.10
p3.4.n_5%	380.00	380.00	40.16	380.00	380.00	40.71
p3.4.o_5%	–	–	0.00	–	–	0.00
p3.4.p_5%	530.00	530.00	2.54	530.00	530.00	3.08
p3.4.q_5%	500.00	500.00	2304.71	500.00	500.00	4016.78
p3.4.r_5%	560.00	560.00	32.88	560.00	560.00	33.19
p3.4.s_5%	610.00	610.00	2.27	610.00	610.00	2.88
p3.4.t_5%	630.00	630.00	2.52	630.00	630.00	3.04
p4.2.a_5%	–	–	0.00	–	–	0.00
p4.2.b_5%	–	–	0.01	–	–	0.01
p4.2.c_5%	–	–	0.01	–	–	0.01
p4.2.d_5%	–	–	1635.29	–	–	1075.85
p4.2.e_5%	443.00	443.00	761.59	443.00	443.00	850.04
p4.2.f_5%	618.00	618.00	103.01	618.00	618.00	160.12
p4.2.g_5%	574.00	574.00	865.72	574.00	574.00	888.54
p4.2.h_5%	718.00	718.00	487.22	718.00	718.00	396.10
p4.2.i_5%	728.00	728.00	447.40	728.00	728.00	596.27
p4.2.j_5%	873.00	873.00	250.53	873.00	873.00	312.77
p4.2.k_5%	955.00	955.00	336.00	955.00	955.00	408.87
p4.2.l_5%	1049.00	1049.00	180.60	1049.00	1049.00	263.28
p4.2.m_5%	1096.00	1096.00	272.60	1096.00	1096.00	336.91
p4.2.n_5%	1111.00	1111.00	508.81	1111.00	1111.00	1105.25
p4.2.o_5%	1149.00	1149.00	298.27	1149.00	1149.00	275.79
p4.2.p_5%	1157.00	1157.00	1918.14	1157.00	1157.00	1318.11
p4.2.q_5%	1202.00	1202.00	2316.38	1202.00	1202.00	3962.04
p4.2.r_5%	1225.00	1225.00	650.59	1225.00	1225.00	695.41
p4.2.s_5%	1213.00	1213.00	5011.03	1213.00	1213.00	5110.12
p4.2.t_5%	1243.00	1243.00	303.49	1243.00	1243.00	281.73
p4.3.a_5%	–	–	0.00	–	–	0.00
p4.3.b_5%	–	–	0.00	–	–	0.00
p4.3.c_5%	–	–	0.00	–	–	0.00

Continued on next page

Table A.15 — continued from previous page.

	CPA$_2$+FP_cuts_LNS_5000			CPA$_2$+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p4.3.d_5%	–	–	0.01	–	–	0.01
p4.3.e_5%	–	–	0.01	–	–	0.01
p4.3.f_5%	–	–	0.01	–	–	0.01
p4.3.g_5%	–	–	0.01	–	–	0.01
p4.3.h_5%	384.00	416.95	7200.00	383.00	414.95	7200.00
p4.3.i_5%	742.00	742.00	432.14	742.00	742.00	396.42
p4.3.j_5%	724.00	724.00	1335.19	724.00	724.00	1459.80
p4.3.k_5%	829.00	829.00	938.27	829.00	829.00	1574.56
p4.3.l_5%	888.00	904.97	7200.00	885.00	906.29	7200.00
p4.3.m_5%	977.00	1003.89	7200.00	982.00	998.15	7200.00
p4.3.n_5%	1030.00	1030.00	2409.08	1030.00	1030.00	2219.47
p4.3.o_5%	1114.00	1123.99	7200.00	1105.00	1131.08	7200.00
p4.3.p_5%	1152.00	1152.00	349.75	1152.00	1152.00	318.76
p4.3.q_5%	1169.00	1183.64	7200.00	1177.00	1177.00	2646.79
p4.3.r_5%	1215.00	1221.83	7200.00	1215.00	1221.60	7200.00
p4.3.s_5%	1259.00	1262.14	7200.00	1259.00	1262.03	7200.00
p4.3.t_5%	1254.00	1256.00	7200.00	1254.00	1256.00	7200.00
p4.4.a_5%	–	–	0.00	–	–	0.00
p4.4.b_5%	–	–	0.00	–	–	0.00
p4.4.c_5%	–	–	0.00	–	–	0.00
p4.4.d_5%	–	–	0.00	–	–	0.00
p4.4.e_5%	–	–	0.00	–	–	0.00
p4.4.f_5%	–	–	0.00	–	–	0.00
p4.4.g_5%	–	–	0.01	–	–	0.01
p4.4.h_5%	–	–	0.01	–	–	0.01
p4.4.i_5%	–	–	0.01	–	–	0.01
p4.4.j_5%	573.00	573.00	137.62	573.00	573.00	162.62
p4.4.k_5%	–	–	0.02	–	–	0.02
p4.4.l_5%	682.00	682.00	728.62	682.00	682.00	784.22
p4.4.m_5%	–	–	0.02	–	–	0.02
p4.4.n_5%	846.00	903.23	7200.00	848.00	905.66	7200.00
p4.4.o_5%	945.00	982.33	7200.00	945.00	982.27	7200.00
p4.4.p_5%	1048.00	1058.92	7200.00	1048.00	1058.86	7200.00
p4.4.q_5%	1078.00	1123.01	7200.00	1078.00	1122.99	7200.00
p4.4.r_5%	1130.00	1171.22	7200.00	1137.00	1171.51	7200.00
p4.4.s_5%	1204.00	1224.46	7200.00	1208.00	1222.56	7200.00
p4.4.t_5%	1244.00	1257.73	7200.00	1244.00	1257.64	7200.00

Continued on next page

Table A.15 — continued from previous page.

	CPA_2+FP_cuts_LNS_5000			CPA_2+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p5.2.a_5%	–	–	0.00	–	–	0.00
p5.2.b_5%	–	–	0.00	–	–	0.00
p5.2.c_5%	–	–	0.00	–	–	0.00
p5.2.d_5%	–	–	0.00	–	–	0.00
p5.2.e_5%	–	–	0.01	–	–	0.00
p5.2.f_5%	–	–	0.01	–	–	0.01
p5.2.g_5%	–	–	0.01	–	–	0.02
p5.2.h_5%	–	–	2.33	–	–	2.21
p5.2.i_5%	350.00	350.00	129.16	350.00	350.00	129.55
p5.2.j_5%	460.00	460.00	15.19	460.00	460.00	28.36
p5.2.k_5%	505.00	505.00	40.32	505.00	505.00	54.81
p5.2.l_5%	660.00	660.00	14.20	660.00	660.00	24.48
p5.2.m_5%	730.00	730.00	9.63	730.00	730.00	20.09
p5.2.n_5%	815.00	815.00	90.87	815.00	815.00	98.25
p5.2.o_5%	900.00	900.00	26.53	900.00	900.00	48.88
p5.2.p_5%	1030.00	1030.00	9.07	1030.00	1030.00	14.12
p5.2.q_5%	1080.00	1080.00	16.95	1080.00	1080.00	22.61
p5.2.r_5%	1170.00	1170.00	27.90	1170.00	1170.00	34.12
p5.2.s_5%	1250.00	1250.00	14.95	1250.00	1250.00	21.93
p5.2.t_5%	1310.00	1310.00	20.67	1310.00	1310.00	28.05
p5.2.u_5%	1360.00	1360.00	25.32	1360.00	1360.00	33.62
p5.2.v_5%	1395.00	1395.00	663.67	1395.00	1395.00	682.19
p5.2.w_5%	1465.00	1465.00	39.00	1465.00	1465.00	49.33
p5.2.x_5%	1490.00	1490.00	30.17	1490.00	1490.00	40.64
p5.2.y_5%	1535.00	1535.00	64.76	1535.00	1535.00	85.93
p5.2.z_5%	1570.00	1570.00	7.55	1570.00	1570.00	16.36
p5.3.a_5%	–	–	0.00	–	–	0.00
p5.3.b_5%	–	–	0.00	–	–	0.00
p5.3.c_5%	–	–	0.00	–	–	0.00
p5.3.d_5%	–	–	0.00	–	–	0.00
p5.3.e_5%	–	–	0.00	–	–	0.00
p5.3.f_5%	–	–	0.00	–	–	0.00
p5.3.g_5%	95.00	95.00	0.87	95.00	95.00	1.05
p5.3.h_5%	–	–	0.00	–	–	0.01
p5.3.i_5%	–	–	0.01	–	–	0.01
p5.3.j_5%	–	–	0.01	–	–	0.01
p5.3.k_5%	425.00	425.00	56.76	425.00	425.00	68.71
p5.3.l_5%	490.00	490.00	239.69	490.00	490.00	247.43

Continued on next page

Table A.15 — continued from previous page.

	CPA$_2$+FP_cuts_LNS_5000			CPA$_2$+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p5.3.m_5%	535.00	535.00	2308.63	535.00	535.00	927.30
p5.3.n_5%	665.00	665.00	374.30	665.00	665.00	352.40
p5.3.o_5%	740.00	740.00	368.91	740.00	740.00	520.20
p5.3.p_5%	860.00	860.00	9.73	860.00	860.00	16.52
p5.3.q_5%	965.00	965.00	21.01	965.00	965.00	26.27
p5.3.r_5%	985.00	985.00	748.67	985.00	985.00	521.46
p5.3.s_5%	1090.00	1113.25	7200.00	1090.00	1112.76	7200.00
p5.3.t_5%	1150.00	1191.71	7200.00	1150.00	1193.02	7200.00
p5.3.u_5%	1225.00	1250.96	7200.00	1225.00	1253.11	7200.00
p5.3.v_5%	1315.00	1315.00	3785.79	1315.00	1315.00	4085.58
p5.3.w_5%	1390.00	1401.60	7200.00	1390.00	1390.00	3657.10
p5.3.x_5%	1455.00	1455.00	4177.00	1455.00	1455.00	3062.75
p5.3.y_5%	1500.00	1530.67	7200.00	1500.00	1531.04	7200.00
p5.3.z_5%	1515.00	1540.00	7200.00	1515.00	1540.00	7200.00
p5.4.a_5%	–	–	0.00	–	–	0.00
p5.4.b_5%	–	–	0.00	–	–	0.00
p5.4.c_5%	–	–	0.00	–	–	0.00
p5.4.d_5%	–	–	0.00	–	–	0.00
p5.4.e_5%	–	–	0.00	–	–	0.00
p5.4.f_5%	–	–	0.00	–	–	0.00
p5.4.g_5%	–	–	0.00	–	–	0.00
p5.4.h_5%	–	–	0.00	–	–	0.00
p5.4.i_5%	–	–	0.00	–	–	0.00
p5.4.j_5%	220.00	220.00	1.91	220.00	220.00	2.04
p5.4.k_5%	–	–	0.00	–	–	0.00
p5.4.l_5%	–	–	0.01	–	–	0.01
p5.4.m_5%	–	–	0.01	–	–	0.01
p5.4.n_5%	440.00	440.00	22.78	440.00	440.00	27.93
p5.4.o_5%	–	–	0.02	–	–	0.02
p5.4.p_5%	665.00	697.80	7200.00	665.00	694.70	7200.00
p5.4.q_5%	715.00	715.00	6483.63	715.00	722.66	7200.00
p5.4.r_5%	850.00	850.00	602.81	850.00	850.00	392.47
p5.4.s_5%	935.00	952.75	7200.00	935.00	956.82	7200.00
p5.4.t_5%	1060.00	1060.00	4846.70	1060.00	1060.00	3478.27
p5.4.u_5%	1190.00	1190.00	9.79	1190.00	1190.00	14.70
p5.4.v_5%	1230.00	1251.52	7200.00	1230.00	1251.63	7200.00
p5.4.w_5%	1260.00	1284.63	7200.00	1260.00	1284.80	7200.00
p5.4.x_5%	1310.00	1345.19	7200.00	1310.00	1348.19	7200.00

Continued on next page

Table A.15 — continued from previous page.

	CPA$_2$+FP_cuts_LNS_5000			CPA$_2$+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p5.4.y_5%	1400.00	1422.23	7200.00	1400.00	1424.63	7200.00
p5.4.z_5%	1530.00	1530.00	23.79	1530.00	1530.00	32.76
p6.2.a_5%	–	–	0.00	–	–	0.00
p6.2.b_5%	–	–	0.00	–	–	0.00
p6.2.c_5%	–	–	0.00	–	–	0.00
p6.2.d_5%	–	–	0.00	–	–	0.00
p6.2.e_5%	282.00	282.00	1.14	282.00	282.00	3.14
p6.2.f_5%	474.00	474.00	2.31	474.00	474.00	5.59
p6.2.g_5%	522.00	522.00	16.03	522.00	522.00	24.26
p6.2.h_5%	708.00	708.00	5.80	708.00	708.00	11.58
p6.2.i_5%	828.00	828.00	9.21	828.00	828.00	16.51
p6.2.j_5%	846.00	846.00	1453.75	846.00	846.00	1481.26
p6.2.k_5%	948.00	948.00	1013.62	948.00	948.00	617.78
p6.2.l_5%	1008.00	1008.00	914.78	1008.00	1008.00	371.34
p6.2.m_5%	1086.00	1086.00	1167.78	1086.00	1086.00	1657.98
p6.2.n_5%	1170.00	1170.00	21.18	1170.00	1170.00	29.38
p6.3.a_5%	–	–	0.00	–	–	0.00
p6.3.b_5%	–	–	0.00	–	–	0.00
p6.3.c_5%	–	–	0.00	–	–	0.00
p6.3.d_5%	–	–	0.00	–	–	0.00
p6.3.e_5%	–	–	0.00	–	–	0.00
p6.3.f_5%	–	–	0.00	–	–	0.00
p6.3.g_5%	–	–	0.00	–	–	0.00
p6.3.h_5%	–	–	0.01	–	–	0.00
p6.3.i_5%	582.00	582.00	3.22	582.00	582.00	6.06
p6.3.j_5%	678.00	678.00	16.15	678.00	678.00	16.52
p6.3.k_5%	840.00	840.00	89.27	840.00	840.00	90.94
p6.3.l_5%	930.00	930.00	41.82	930.00	930.00	47.23
p6.3.m_5%	1014.00	1023.74	7200.00	1014.00	1014.00	4456.22
p6.3.n_5%	1092.00	1092.00	499.31	1092.00	1092.00	428.76
p6.4.a_5%	–	–	0.00	–	–	0.00
p6.4.b_5%	–	–	0.00	–	–	0.00
p6.4.c_5%	–	–	0.00	–	–	0.00
p6.4.d_5%	–	–	0.00	–	–	0.00
p6.4.e_5%	–	–	0.00	–	–	0.00
p6.4.f_5%	–	–	0.00	–	–	0.00
p6.4.g_5%	–	–	0.00	–	–	0.00

Continued on next page

Table A.15 — continued from previous page.

	CPA$_2$+FP_cuts_LNS_5000			CPA$_2$+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p6.4.h_5%	–	–	0.00	–	–	0.00
p6.4.i_5%	–	–	0.00	–	–	0.00
p6.4.j_5%	–	–	0.00	–	–	0.00
p6.4.k_5%	318.00	318.00	8.01	318.00	318.00	8.95
p6.4.l_5%	660.00	660.00	113.84	660.00	660.00	169.34
p6.4.m_5%	786.00	786.00	36.10	786.00	786.00	39.86
p6.4.n_5%	990.00	990.00	9.69	990.00	990.00	14.00
p7.2.a_5%	–	–	0.00	–	–	0.00
p7.2.b_5%	–	–	0.00	–	–	0.00
p7.2.c_5%	–	–	0.00	–	–	0.00
p7.2.d_5%	–	–	0.00	–	–	0.00
p7.2.e_5%	–	–	0.01	–	–	0.01
p7.2.f_5%	–	–	0.04	–	–	0.04
p7.2.g_5%	–	–	2.40	–	–	2.33
p7.2.h_5%	337.00	337.00	61.80	337.00	337.00	99.59
p7.2.i_5%	–	–	0.04	–	–	0.04
p7.2.j_5%	–	–	7200.00	–	–	7200.00
p7.2.k_5%	–	–	7200.00	–	–	7200.00
p7.2.l_5%	636.00	636.00	746.21	636.00	636.00	927.43
p7.2.m_5%	731.00	731.00	583.88	731.00	731.00	673.26
p7.2.n_5%	656.00	656.00	975.25	656.00	656.00	1093.89
p7.2.o_5%	802.00	802.00	461.66	802.00	802.00	570.78
p7.2.p_5%	844.00	844.00	2882.32	844.00	844.00	2994.84
p7.2.q_5%	885.00	885.00	1443.13	885.00	885.00	3142.25
p7.2.r_5%	968.00	968.00	358.42	968.00	968.00	425.41
p7.2.s_5%	1007.00	1007.00	478.26	1007.00	1007.00	490.02
p7.2.t_5%	1075.00	1075.00	323.46	1075.00	1075.00	494.54
p7.3.a_5%	–	–	0.00	–	–	0.00
p7.3.b_5%	–	–	0.00	–	–	0.00
p7.3.c_5%	–	–	0.00	–	–	0.00
p7.3.d_5%	–	–	0.00	–	–	0.00
p7.3.e_5%	–	–	0.00	–	–	0.00
p7.3.f_5%	–	–	0.00	–	–	0.00
p7.3.g_5%	–	–	0.01	–	–	0.01
p7.3.h_5%	–	–	0.02	–	–	0.02
p7.3.i_5%	–	–	0.04	–	–	0.04
p7.3.j_5%	-inf	inf	7200.00	-inf	inf	7200.00

Continued on next page

Table A.15 — continued from previous page.

	CPA_2+FP_cuts_LNS_5000			CPA_2+OFP_cuts_LNS_5000		
Instance	LB	UB	Time (s)	LB	UB	Time (s)
p7.3.k_5%	395.00	395.00	123.22	395.00	395.00	164.72
p7.3.l_5%	522.00	522.00	195.70	522.00	522.00	572.04
p7.3.m_5%	–	–	0.04	–	–	0.04
p7.3.n_5%	–	–	182.70	–	–	180.95
p7.3.o_5%	602.00	602.00	234.72	602.00	602.00	381.97
p7.3.p_5%	757.00	757.00	3657.93	757.00	757.00	4464.31
p7.3.q_5%	874.00	874.00	1637.33	874.00	874.00	2141.15
p7.3.r_5%	923.00	923.00	3593.14	923.00	923.00	3779.55
p7.3.s_5%	949.00	971.81	7200.00	944.00	981.09	7200.00
p7.3.t_5%	1034.00	1043.60	7200.00	1034.00	1044.30	7200.00
p7.4.a_5%	–	–	0.00	–	–	0.00
p7.4.b_5%	–	–	0.00	–	–	0.00
p7.4.c_5%	–	–	0.00	–	–	0.00
p7.4.d_5%	–	–	0.00	–	–	0.00
p7.4.e_5%	–	–	0.00	–	–	0.00
p7.4.f_5%	–	–	0.00	–	–	0.00
p7.4.g_5%	–	–	0.00	–	–	0.00
p7.4.h_5%	–	–	0.00	–	–	0.00
p7.4.i_5%	–	–	0.01	–	–	0.01
p7.4.j_5%	–	–	0.01	–	–	0.01
p7.4.k_5%	–	–	0.02	–	–	0.02
p7.4.l_5%	–	–	0.04	–	–	0.04
p7.4.m_5%	–	–	0.06	–	–	0.06
p7.4.n_5%	–	–	0.09	–	–	0.09
p7.4.o_5%	537.00	537.00	1182.18	537.00	537.00	1278.87
p7.4.p_5%	771.00	771.00	4971.00	771.00	771.00	3656.41
p7.4.q_5%	781.00	794.85	7200.00	781.00	797.28	7200.00
p7.4.r_5%	775.00	843.52	7200.00	794.00	825.42	7200.00
p7.4.s_5%	857.00	887.61	7200.00	857.00	886.77	7200.00
p7.4.t_5%	939.00	958.84	7200.00	939.00	960.11	7200.00

Table A.13: Comparison of the four variations of the hybrid algorithm tested at solving the new benchmark of STOP instances. Bold entries highlight, for each instance set, the best algorithm(s) in terms of number of instances solved to optimality.

	CPA_1+FP_cuts_LNS_5000				CPA_1+OFP_cuts_LNS_5000			
	solved		*unsolved*		*solved*		*unsolved*	
			Gap (%)				Gap (%)	
Set	#opt/total	Time (s)	Avg	StDev	#opt/total	Time (s)	Avg	StDev
1_5%	**54/54**	1.55	–	–	**54/54**	1.93	–	–
2_5%	**33/33**	0.07	–	–	**33/33**	0.09	–	–
3_5%	**60/60**	92.34	–	–	**60/60**	77.13	–	–
4_5%	44/60	852.38	1.98	1.62	**45/60**	709.68	1.91	1.75
5_5%	**65/78**	305.20	2.41	0.91	64/78	185.42	2.35	1.14
6_5%	**42/42**	284.87	–	–	**42/42**	284.35	–	–
7_5%	**53/60**	890.07	16.15	36.99	51/60	679.23	12.83	32.71
Total	**351/387**	347.89	4.89	16.35	349/387	272.55	4.66	15.95

	CPA_2+FP_cuts_LNS_5000				CPA_2+OFP_cuts_LNS_5000			
	solved		*unsolved*		*solved*		*unsolved*	
			Gap (%)				Gap (%)	
Set	#opt/total	Time (s)	Avg	StDev	#opt/total	Time (s)	Avg	StDev
1_5%	**54/54**	2.19	–	–	**54/54**	2.46	–	–
2_5%	**33/33**	0.07	–	–	**33/33**	0.09	–	–
3_5%	**60/60**	118.79	–	–	**60/60**	140.63	–	–
4_5%	45/60	503.94	2.47	2.28	**46/60**	600.00	2.52	2.27
5_5%	**66/78**	383.80	2.21	1.01	**66/78**	285.92	2.28	0.90
6_5%	41/42	132.27	0.95	–	**42/42**	226.11	–	–
7_5%	**53/60**	726.48	16.95	36.70	**53/60**	791.37	16.60	36.79
Total	352/387	281.77	5.23	16.60	**354/387**	300.80	5.42	17.06