A branch-cut-and-price approach for the single-trip and multi-trip two-echelon vehicle routing problem with time windows

Guillaume Marques, Ruslan Sadykov, Jean-Christophe Deschamps, Rémy Dupas

A branch-cut-and-price approach for the single-trip and multi-trip two-echelon vehicle routing problem with time windows

Guillaume Marques[*1,2,3], Ruslan Sadykov[†2,3], Jean-Christophe Deschamps[‡1], and Rémy Dupas[§1]

[1]IMS, Univ. of Bordeaux, Bordeaux INP, CNRS (UMR 5218) , 351 Cours de la libération, 33405 Talence cedex, France
[2]Inria Bordeaux – Sud-Ouest , 200 Avenue de la Vieille Tour, 33405 Talence cedex, France
[3]IMB, Univ. of Bordeaux, CNRS (UMR 5251), Bordeaux INP, 351 Cours de la libération, 33405 Talence cedex, France

30 November 2020

Abstract

The paper studies the two-echelon capacitated vehicle routing problem with time windows, in which delivery of freight from depots to customers is performed using intermediate facilities called satellites. We consider the variant of the problem with precedence constraints for unloading and loading freight at satellites. This variant allows for storage and consolidation of freight at satellites. Thus, the total transportation cost may decrease in comparison with the alternative variant with exact freight synchronization at satellites. We suggest a mixed integer programming formulation for the problem with an exponential number of route variables and an exponential number of precedence constraints which link first-echelon and second-echelon routes. Routes at the second echelon connecting satellites and clients may consist of multiple trips and visit several satellites. A branch-cut-and-price algorithm is proposed to solve efficiently the problem. This is the first exact algorithm in the literature for the multi-trip variant of the problem. We also present a post-processing procedure to check whether the solution can be transformed to avoid freight consolidation and storage without increasing its transportation cost. Our algorithm significantly outperforms another recent one for the single-trip variant of the problem. We also show that all single-trip literature instances solved to optimality admit optimal solutions of the same cost for both variants of the problem either with precedence constraints or with exact synchronization constraints.

1 Introduction

The strong growth of home delivery services and e-commerce leads to a massive flow of goods to the city centers. This tends to bring trucks within cities, while the latter restrict or ban the use of polluting and large freight vehicles in city centers. In order to find alternative solutions to direct deliveries from distribution centers to customers, multi-echelon distribution networks were

[*]guillaume.marques@inria.fr
[†]ruslan.sadykov@inria.fr
[‡]jean-christophe.deschamps@ims-bordeaux.fr
[§]remy.dupas@ims-bordeaux.fr

proposed by Crainic et al. (2009). The two-echelon distribution systems is the simplest structure of these distribution networks. The trucks circulates in the first level outside the city center while small and clean vehicles are used in the second level for last-mile delivery. Light electric freight vehicles or cargo bikes as commonly used at this level since they are agile, quiet, emission-free, and takes up less space than vans or trucks. The connection between the two levels is ensured by intermediate warehouses such as Urban Consolidation Centers (UCC) (Allen et al., 2012), which provide temporary storage and consolidate the parcels flow in the last mile (McDermott, 1975). As the costs of these UCCs are high (Holguín-Veras et al., 2018), an alternative is to use intermediate warehouses with limited storage space or no storage at all. These warehouses called satellites are commonly based on existing infrastructure such as car parks, bus depots, or some street sidewalks. In this context, synchronization of flows at intermediate warehouses is therefore an essential feature in urban freight transport: exact synchronization constraints are encountered in satellites, and precedence constraints are encountered in UCCs. Time windows at customer sites are also commonly used in practice.

In this paper, we study the two-echelon vehicle routing problem with time windows (2E-VRPTW), which consists in determining the number of goods to be shipped from the distribution centers to the satellites and from satellites to customers, together with the optimal routes connecting entities in each level, such that vehicle capacities are not exceeded, customer demands are satisfied, customers are delivered within their time windows, and first-level routes precede second-level routes to do transfers at satellites. The goal is to minimize operational and transportation costs.

Two-echelon vehicle routing with time windows has received little attention in the literature so far. Usually in the variants of the problem considered in the literature, exact synchronization is required, and one forbids freight consolidation, i.e. the loading to a city freighter from several urban trucks. Such constraints are imposed, for example, in the papers by Grangier et al. (2016) and Dellaert et al. (2019). In contrast to these papers, our problem variant allows for consolidation and does not require exact synchronization for transfers because we use precedence constraints at satellites. Our case is thus suited for the practical situations with UCCs, i.e. satellites with relatively large storage capacities. In this work, we propose the first exact algorithm for this variant of the problem. The algorithm is based on the branch-cut-and-price (BCP) approach.

We would like to underline that our algorithm is useful both for the case with precedence constraints and for the case with exact synchronization. Indeed, exact algorithms are generally used in practice to obtain valid lower bounds on the value of an optimum solution of the problem. These bounds are then used to estimate the quality of heuristic algorithms. As the variant of the 2E-VRPTW with precedence constraints and consolidation is a relaxation for the variant with exact synchronization, the lower bounds obtained by our algorithm are valid for both cases. Moreover, we provide a post-processing procedure which allows one to minimize the usage of storage in a given solution without increasing its transportation cost.

We now summarize the main contributions of our work.

- We introduce a new mixed-integer programming (MIP) formulation for the 2E-VRPTW with precedence constraints and freight consolidation. This formulation involves an exponential number of route variables and an exponential number of precedence constraints. Our formulation does not involve variables which explicitly model freight transfer at satellites. This fact greatly simplifies the following approach to solve the formulation.
- To solve the introduced formulation to optimality, we propose a branch-cut-and-price algorithm, which combines column and cut generation with strong branching and an enumeration procedure for elementary routes (Baldacci et al., 2008). Our algorithm incorporates many advanced techniques proposed recently for tackling classic vehicle routing problems. It includes an original separation algorithm which generates violated precedence constraints. We also show how precedence constraints can be taken into account when solving the pricing problem in column generation.

- We show how to adapt our BCP algorithm for a more practically relevant variant of the problem, in which city freighters can perform multiple trips.
- We perform extensive computational evaluation of our algorithm using literature instances introduced by Grangier et al. (2016) and Dellaert et al. (2019). Experiments reveal that it can solve to optimality single-trip instances with up to 6 distribution centers, 5 satellites and 100 customers, and multi-trip instances with up to 8 satellites and 100 customers. Moreover, we show that: (i) virtually all instances proposed by Dellaert et al. (2019) have optimum solutions with the same transportation cost for the both variants of the problem with precedence constraints and with exact synchronization; (ii) our algorithm solves to optimality 54 open instances of the single-trip 2E-VRPTW; (iii) it outperforms significantly the algorithm proposed by Dellaert et al. (2019) on their instances.

The remaining of the paper is organized as follows. Section 2 reviews the literature. MIP formulations of the problem are introduced in Section 3. In Section 4, we describe the proposed branch-cut-and-price algorithm. In Section 5, we present and discuss the computational results. Section 6 concludes the paper and presents future research directions. In the appendices, we give the detailed results of the BCP algorithm for each tested instance of the problem.

2 Literature review

The 2E-VRPTW is a generalization of the quite well-studied two-echelon capacitated vehicle routing problem (2E-CVRP). Several exact approaches have been proposed for the 2E-CVRP. Branch-and-cut algorithms were suggested by Gonzalez-Feliu et al. (2007); Perboli et al. (2011); Jepsen et al. (2013); and Contardo et al. (2012). An exact method based on the enumeration of first-level solutions was proposed by Baldacci et al. (2013). The first branch-cut-and-price algorithm was developed by Santos et al. (2015). Recently, Marques et al. (2020) published an improved branch-cut-and-price algorithm which outperforms other exact methods in the literature. Optimum solutions can now be consistently obtained for instances with up to 200 customers and 10 satellites.

Several heuristic approaches for the 2E-CVRP have been proposed in the literature. A large neighbourhood search based method has been suggested by Hemmelmayr et al. (2012) and by Breunig et al. (2016). Zeng et al. (2014) proposed a hybrid heuristic that combines greedy randomized adaptive search procedure and a variable neighborhood descent. Matheuristics that combine local search to build routes and a route-based MIP to derive complete solutions were employed by Wang et al. (2017) and Amarouche et al. (2018). The latter two algorithms are the best heuristics for the problem available in the literature until today. More details on the 2E-CVRP can be found in the survey paper by Cuda et al. (2015).

The 2E-VRPTW and its variants are less studied than the 2E-CVRP, although the former is more relevant in practice. Grangier et al. (2016) suggested a mathematical formulation of the variant of the 2E-VRPTW with multiple trips and exact synchronization (MT-2E-VRPTW-ES), in which freight consolidation is forbidden. A city freighter thus receives products from only one urban truck. The authors proposed an adaptive large neighbourhood search heuristic which embeds customized destroy and repair procedures. Their objective function successively minimizes the number of urban trucks used, the number of city freighters used, and the total distance travelled. They experimented with instances involving 8 satellites and 100 customers and searched for feasible solutions within two hours.

Dellaert et al. (2019) suggested three MIP formulations for the single-trip 2E-VRPTW with exact synchronization (ST-2E-VRPTW-ES), in which freight consolidation is also forbidden. First, they introduced an arc-based formulation optimized using a commercial MIP solver. This approach could only solve instances with 15 customers within one hour. Secondly, they proposed a "tour-tree" based formulation, in which a tour-tree is a combination of a first-level route and the second-level routes loaded by this first-level route. A branch-and-price algorithm was devised to

tackle this formulation. Again, only instances with up to 15 customers could be solved. Finally, the authors proposed a route based formulation and an enumeration-based algorithm similar to the one by Baldacci et al. (2013) to solve the ST-2E-VRPTW-ES. The method generates collections of first-level routes, then iteratively fixes the first-level routes by choosing the most promising collection according to a lower bound, and finally optimizes the second-level problem as a multi-depot vehicle routing problem with time windows using a branch-and-price algorithm. This method could solve most instances with up to 50 customers and some instances with 100 customers.

Li et al. (2016) considered a variant of the 2E-VRPTW for linehaul delivery systems with exact synchronization. They suggested a MIP formulation and tackled the problem with combination of an initial Clarke-and-Wright savings heuristic and a local search. Li et al. (2020b) studied another variant of the 2E-VRPTW with mobile satellites. The first echelon involves vans and the second echelon involves unmanned aerial vehicles (UAV). The vans parked at some customer locations are used as mobile satellites from which drones deliver other customers. They proposed a vehicle-flow formulation which involves non-exact synchronization constraints at mobile satellites and time windows at customer locations. Given the specific nature of the distribution with UAVs at the second level, their model is not dedicated to achieve freight consolidation at satellites. An adaptive large neighborhood search (ALNS) heuristic has also been proposed, which were tested on instances with up to 100 customers derived from the standard VRPTW benchmark. Li et al. (2020a) considered a city logistics distribution system with on-street satellites and time windows at customer locations, which has some similarities with the 2E-VRPTW. Synchronization is performed at satellites where freight consolidation takes place. The problem is formulated as a MIP, and it is optimized with a variable neighbourhood search (VNS) heuristic. The latter was tested on instances with up to 30 on-street-satellites and 900 customers. Nolz et al. (2020) considered two-echelon urban distribution systems with a single capacitated city hub and exact synchronization between echelons. For this setting, they proposed a three-phase heuristic method which uses population-based meta-heuristics and integer programs.

Related works include models developed for certain specific application areas. Wang and Wen (2020) focused on a variant of the 2E-VRPTW with soft time windows and an heterogeneous fleet of vehicles for the cold chain logistics. They proposed an adaptive genetic algorithm, and optimized small instances with 2 distribution centers, 15 customers and 3 satellites. He and Li (2019) proposed a memetic algorithm for a multi-trip variant of the 2E-VRPTW arising in agriculture, where second-level harvesting machines have to visit many farmlands and unload the crop at one or many of them (i.e. satellites) into first-level trucks. In this problem, the satellite location is changed continuously during the working day and a non-exact synchronization between the two levels is defined by a time windows at satellite. There is no consolidation in this model. The authors used a set of instances with up to 250 customers.

To conclude, several variants of the 2E-VRPTW have been studied in the literature in recent years. Nevertheless, to our knowledge, only one exact method has been proposed so far by Dellaert et al. (2019) to the 2E-VRPTW. In this work, we address the lack of exact methods by proposing an algorithm which has a broader applicability than the existing one.

3 Problem definition and formulation

We now formally define the problem. At first level, a fleet $\mathcal{U}$ of homogeneous urban trucks ships goods from a set $\mathcal{D}$ of distribution centers to a set $\mathcal{S}$ of satellites. The capacity of an urban truck is Q_1 items. The tour of an urban truck starts at a distribution center, delivers freight to some satellites, and ends at the same distribution center. The second level involves a set $\mathcal{F}$ of homogeneous city freighters that ship freight from satellites to a set $\mathcal{C}$ of customers. The capacity of a city freighter is Q_2 items. Each customer $c \in \mathcal{C}$ has an integer demand d_c and must be visited by one city freighter. The latter must arrive at the location of customer $c \in \mathcal{C}$ within a time window starting at time l_c and ending at time u_c (waiting is possible for an early arrival).

Once arrived, the city freighter needs σ_c time units to serve the customer. In the single-trip variant, a city freighter starts its tour from a satellite, visits some customers, and ends at the same satellite. In the multi-trip variant, a city freighter starts from an unique depot, goes to a satellite, delivers some customers, and goes empty to a satellite to start another trip or ends at the depot.

Transfers of freight from urban trucks to city freighters take place at satellites. Vehicles can arrive at satellite $s \in \mathcal{S}$ within a time window $[l_s, u_s]$. In our variant, the freight consolidation is allowed. This means that a city freighter can receive freight from several urban trucks. Moreover, exact synchronization of an urban truck and a city freighter at a satellite is not required. A transfer at satellite $s \in \mathcal{S}$ consists of the following steps. An urban truck arrives at the satellite, possibly waits until the beginning of time window l_s, stays during service time σ_s, and then leaves. A city freighter arrives at the satellite, possibly waits until the start of service time of an urban truck from which the city freighter gets its freight, stays during service time σ_s, and then leaves. Figure 1 depicts examples of feasible transfers.

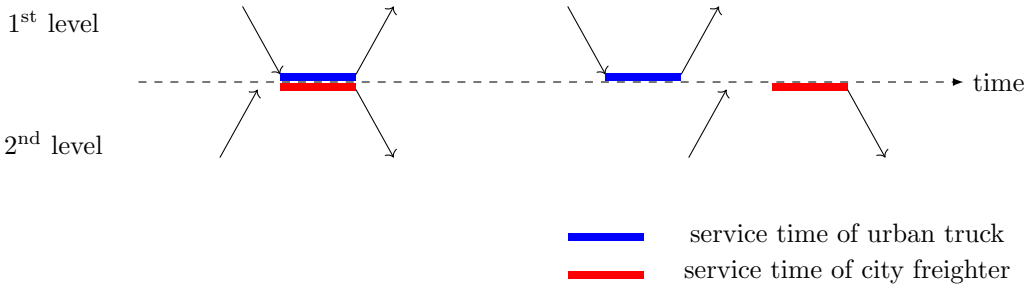


Figure 1: Examples of transfers at satellite

For the sake of clarity, we now focus on the single-trip variant of the problem. Specificities of the multi-trip variant are discussed in Section 3.4.

The first-level problem is similar to the split-delivery CVRP because several urban trucks can supply a satellite. However, the amount of freight delivered to each satellite is not fixed. The second level problem is similar to the multi-depot CVRP with time windows, in which satellites take the role of depots.

The distribution system is represented by two graphs. Directed graph $G_1 = (V_1, A_1)$ with $V_1 = \mathcal{D} \cup \mathcal{S}$ and $A_1 = \mathcal{D} \times \mathcal{S} \cup \{(s, s') \in \mathcal{S}^2 : s \neq s'\} \cup \mathcal{S} \times \mathcal{D}$ represents the first level of the distribution system. Directed graph $G_2 = (V_2, A_2)$ with $V_2 = \mathcal{S} \cup \mathcal{C}$ and $A_2 = \mathcal{S} \times \mathcal{C} \cup \{(c, c') \in \mathcal{C}^2 : c \neq c'\} \cup \mathcal{C} \times \mathcal{S}$ represents the second level of the distribution system. For each arc $a \in A_1 \cup A_2$, travelling cost f_a and travel time t_a are given.

We denote as P the set of feasible first-level routes. A route $p \in P$ is an elementary cycle $(v_0^p, v_1^p, \ldots, v_{n(p)}^p)$ in G_1, in which $v_0^p = v_{n(p)}^p \in \mathcal{D}$, and $v_k^p \in \mathcal{S}$, $1 \leq k < n(p)$. We denote as S_p the set of satellites visited by route $p \in P$: $S_p = \{v_1^p, \ldots, v_{n(p)-1}^p\} \subseteq \mathcal{S}$. Since our variant allows for storage of items at satellites, there exists an optimal solution in which each first-level route visits each satellite at most once and departs from each node as early as possible. Let $\tilde{f}^p$ denote the cost of route $p \in P$, which includes the total travel cost and the fixed cost of using an urban truck. Also let $\tilde{t}_k^p$ denote the departure time of route $p \in P$ from node v_k^p, $0 \leq k \leq n(p)$. Without loss of generality, each value $\tilde{t}_k^p$ can be fixed to the earliest departure time:

$$\tilde{t}_k^p = \begin{cases} \sigma_{v_k^p}, & k = 0, \\ \max\left\{\tilde{t}_{k-1}^p + t_{(v_{k-1}^p, v_k^p)},\ l_{v_k^p}\right\} + \sigma_{v_k^p}, & 1 \leq k \leq n(p). \end{cases}$$

A first-level route p is feasible if $\tilde{t}_k^p \leq u_{v_k^p} + \sigma_{v_k^p}$, $1 \leq k \leq n(p)$. For a pair (s, t), where $s \in \mathcal{S}$ and $0 \leq t \leq u_s$, we denote as P_{st}, the set of first-level routes which visit satellite s and depart from it strictly before time moment t: $P_{st} = \{p \in P : \exists k, 1 \leq k < n(p), v_k^p = s, \tilde{t}_k^p < t\}$.

We denote as R_s the set of feasible second-level routes starting from satellite s. Let also $R = \bigcup_{s\in\mathcal{S}} R_s$. A route $r \in R_s$ is an elementary cycle $(v_0^r, v_1^r, \ldots, v_{n(r)}^r)$ in G_2, in which $v_0^r = v_{n(r)}^r = s$, and $v_k^r \in \mathcal{C}$, $1 \le k < n(r)$. Again, since our variant allows for storage of items at satellites, there exists an optimal solution in which each second-level route departs from each node as late as possible. Let $\tilde{z}_c^r$ be equal to 1 if route $r \in R$ serves customer $c \in \mathcal{C}$, and 0 otherwise. Let $\tilde{d}^r$ be the total amount of freight delivered by route $r \in R$: $\tilde{d}^r = \sum_{c\in\mathcal{C}} d_c \tilde{z}_c^r \le Q_2$. Let $\tilde{f}^r$ denote the cost of route $r \in R$, which includes the total travel cost and the fixed cost of using a city freighter. Also let $\tilde{t}_k^r$ denote the departure time of route $r \in R$ from node v_k^p, $0 \le k \le n(p)$. Without loss of generality, each value $\tilde{t}_k^r$ can be fixed to the latest departure time:

$$\tilde{t}_k^r = \begin{cases} u_{v_k^r}, & k = n(r), \\ \min\left\{\tilde{t}_{k+1}^r - \sigma_{v_{k+1}^r} - t_{(v_k^r, v_{k+1}^r)},\ u_{v_k^r} + \sigma_{v_k^r}\right\}, & 0 \le k < n(r). \end{cases}$$

A second-level route r is feasible if $\tilde{t}_k^r \ge l_{v_k^r} + \sigma_{v_k^r}$, $0 \le k < n(r)$. For a pair (s,t), where $s \in \mathcal{S}$ and $0 \le t \le u_s$, we denote as R_{st} the set of second-level routes in R_s which depart from satellite s strictly before time moment t: $R_{st} = \{r \in R_s :\ \tilde{t}_0^r < t\}$.

A feasible solution to the problem consists of a set of feasible first-level and second-level routes satisfying the following partitioning, precedence, and capacity constraints:

(C1) each customer is visited by exactly one second-level route,

(C2) for each satellite $s \in S$ and each time moment $0 \le t \le u_s$, the total amount of freight, delivered to s by first-level routes in P_{st}, is not smaller than the total amount of freight delivered by second-level routes in R_{st},

(C3) the total amount of freight delivered by every first-level route does not exceed Q_1.

The objective function is the same as the one used by Dellaert et al. (2019): we need to minimize the sum of the total travelling cost and the total fixed cost of vehicles usage, i.e. the total routes cost.

3.1 Standard formulation

Let integer variable λ_p, $p \in P$, be equal to the number of urban trucks which follow first-level route p. Let binary variable μ_r, $r \in R$, takes value 1 if a city freighter follows second-level route r, and 0 otherwise. Let continuous variable w_{ps}, $p \in P$, $s \in S_p$, be equal to the amount of freight that first-level route p delivers to satellite s. Then our problem can be formulated as follows.

$$\text{(F1)}\quad \min \quad \sum_{p\in P} \tilde{f}^p \lambda_p + \sum_{r\in R} \tilde{f}^r \mu_r \tag{1}$$

$$\text{s.t.} \quad \sum_{r\in R} \tilde{z}_c^r \mu_r = 1 \qquad c \in \mathcal{C} \tag{2}$$

$$\sum_{p\in P_{st}} w_{ps} - \sum_{r\in R_{st}} \tilde{d}^r \mu_r \ge 0 \qquad s \in \mathcal{S},\ l_s < t \le u_s, \tag{3}$$

$$\sum_{s\in S_p} w_{ps} \le Q_1 \lambda_p \quad p \in P \tag{4}$$

$$\lambda_p \in \mathbb{Z}_+ \qquad p \in P \tag{5}$$

$$\mu_r \in \{0,1\} \quad r \in R \tag{6}$$

$$w_{ps} \ge 0 \qquad p \in P,\ s \in S_p \tag{7}$$

The objective function (1) minimizes the total routes cost. Partitioning constraints (2), precedence constraints (3), and capacity constraints (4) correspond to constraints (C1), (C2), and (C3) respectively. Constraints (5), (6) and (7) define the domains of variables. The number

of precedence constraints (3) can be reduced to a finite number while keeping the formulation valid. We define as T_s the set of all time moments at which first-level routes leave satellite s: $T_s = \{\tilde{t}_k^p : p \in P,\ 1 \le k < n(p),\ v_k^p = s\}$. Then it suffices to keep only constraints (3) for pairs (s,t) such that $s \in \mathcal{S}$ and $t \in T_s$.

Formulation (F1) cannot be solved directly in practice as the number of variables and constraints is very large. Even the standard column and cut generation approach is not suited to solve its linear programming (LP) relaxation. This is because for every newly generated variable λ_p, $p \in P$, one should also generate variables w_{ps}, $s \in S_p$, and the corresponding constraint (4).

3.2 Modified formulation

In this section, we modify formulation (F1) so that dynamic generation of route variables does not require simultaneous generation of constraints. For the modified formulation, we are able to compute the current reduced cost of route variables. The standard column generation procedure then can be used once the restricted set of precedence constraints is fixed. Therefore, precedence constraints separation procedure may alternate with the column generation procedure. The overall columns and cut generation procedure stops when no negative reduced columns are found and no precedence constraints are violated.

The main idea of the modified formulation is to merge constraints (3) and (4) to remove variables w. First we need to introduce some notation. Given a time vector $\tau = (\tau_s)_{s\in\mathcal{S}}$, let $P(\tau)$ be the set of first-level routes which depart from a satellite $s \in \mathcal{S}$ before time τ_s: $P(\tau) = \{p \in P : \exists k,\ 1 \le k < n(p),\ \tilde{t}_k^p < \tau_{v_k^p}\}$. Analogously, let $R(\tau)$ be the set of second-level routes which depart from a satellite $s \in \mathcal{S}$ before time τ_s: $R(\tau) = \{r \in R : \tilde{t}_0^r < \tau_{v_0^r}\}$. Also, let $\mathcal{T}$ be the cartesian product of all sets T_s, $s \in \mathcal{S}$, extended by value 0, i.e. $\mathcal{T} = \bigtimes_{s\in\mathcal{S}}(T_s \cup \{0\})$. The modified formulation is then the following.

$$\text{(F2)}\quad \min \quad \sum_{p\in P} \tilde{f}^p \lambda_p + \sum_{r\in R} \tilde{f}^r \mu_r \tag{8}$$

$$\text{s.t.} \quad \sum_{r\in R} \tilde{z}_c^r \mu_r = 1 \qquad c \in \mathcal{C} \tag{9}$$

$$\sum_{p\in P(\tau)} Q_1 \lambda_p - \sum_{r\in R(\tau)} \tilde{d}^r \mu_r \ge 0 \qquad \tau \in \mathcal{T} \tag{10}$$

$$\lambda_p \in \mathbb{Z}_+ \qquad p \in P \tag{11}$$

$$\mu_r \in \{0,1\} \qquad r \in R \tag{12}$$

We call constraints (10) two-level precedence constraints (TLPC). They replace constraints (3) and (4). Before showing that formulation (F2) is equivalent to (F1), we give an example of a violated TLPC.

Example 1. *Consider an instance of 2E-VRPTW with one distribution center, three satellites* $\mathcal{S} = \{s_1, s_2, s_3\}$*, and a set of customers that has a total demand of* 55 *items. Capacities of vehicles are* $Q_1 = 20$ *and* $Q_2 = 13$*. Suppose we are given the solution depicted by Figure 2. Urban truck following route* p_1 *takes* 20 *units of freight from the distribution center and delivers them to satellite* s_2 *at time* 5 *and satellite* s_1 *at time* 15*. Urban truck taking route* p_2 *delivers* 20 *items to* s_2 *at time* 55*. Urban truck taking route* p_3 *delivers* 15 *items to* s_1 *at time* 75*. City freighters taking routes* r_1 *and* r_5 *start from satellite* s_1 *at time moments* 27 *and* 105 *with loads of* 10 *and* 13 *items respectively. City freighters taking routes* r_2*,* r_3*, and* r_4 *start from* s_2 *at time moments* 40*,* 63*, and* 85 *with loads of* 12*,* 7*,* 13 *items respectively.*

We now consider the TLPC characterized by time vector $\tau = (70, 50, 0)$*. This TLPC involves routes* p_1*,* r_1*, and* r_2 *arriving and leaving satellites in the gray area in Figure 2. Since* p_1 *delivers* 20 *items and* r_1*, and* r_2 *cover* 22 *items of demand, the solution violates this TLPC.*

We now prove that (F2) is a projection of (F1). The proof is illustrated in Figure 3.

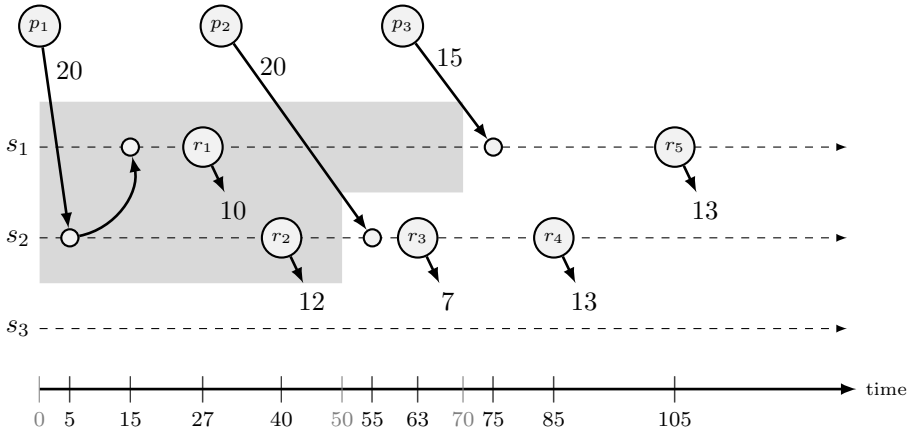


Figure 2: Example of a solution and a violated TLPC characterized by vector $\tau = (70, 50, 0)$

Proposition 1. *A solution $(\bar{\lambda}, \bar{\mu})$ is feasible to the LP relaxation (LF2) of formulation (F2) if and only if there exists a feasible solution $(\bar{\lambda}, \bar{\mu}, \bar{w})$ to the LP relaxation (LF1) of formulation (F1).*

Proof. To prove sufficiency ("if" part), we need to show that constraints (10) are valid for (LF1). Let fix a feasible solution $(\bar{\lambda}, \bar{\mu}, \bar{w})$ to (LF1). For an arbitrary time vector $\tau \in \mathcal{T}$, we have

$$\sum_{r \in R(\tau)} \tilde{d}^r \bar{\mu}_r = \sum_{s \in \mathcal{S}} \sum_{r \in R_{s,\tau_s}} \tilde{d}^r \bar{\mu}_r \overset{(3)}{\leq} \sum_{s \in \mathcal{S}} \sum_{p \in P_{s,\tau_s}} \bar{w}_{ps} \leq \sum_{p \in P(\tau)} \sum_{s \in S_p} \bar{w}_{ps} \overset{(4)}{\leq} \sum_{p \in P(\tau)} Q_1 \bar{\lambda}_p$$

Thus, $(\bar{\lambda}, \bar{\mu})$ is feasible to (LF2).

We now prove necessity, i.e. "only if" part. Consider a feasible solution $(\bar{\lambda}, \bar{\mu})$ to (LF2). We denote as $\bar{P}$ and $\bar{R}$ the sets of first-level and second-level routes participating in the solution: $\bar{P} = \{p \in P : \bar{\lambda}_p > 0\}$ and $\bar{R} = \{r \in R : \bar{\mu}_r > 0\}$.

We build the directed graph $\bar{G} = (\bar{V}, \bar{A})$. The set of nodes is $\bar{V} = \{\bar{s}, \bar{t}\} \cup \bar{P} \cup \bar{R}$, where $\bar{s}$ is the source, and $\bar{t}$ is the sink. Set $\bar{A}$ of arcs consists of three subsets. Subset $\bar{A}_1$ contains, for each $p \in \bar{P}$, arc $(\bar{s}, p)$ with capacity $Q_1 \bar{\lambda}_p$. Subset $\bar{A}_2$ contains arc $(p, r) \in \bar{P} \times \bar{R}$ if and only if first level route p leaves satellite $s = v_0^r$ before or at the same time as second-level route r. Every arc in $\bar{A}_2$ has infinite capacity. Subset $\bar{A}_3$ contains, for each $r \in \bar{R}$, arc $(r, \bar{t})$ with capacity $\tilde{d}^r \bar{\mu}_r$.

Let us now prove by contradiction that the maximum flow value from $\bar{s}$ to $\bar{t}$ in $\bar{G}$ is equal to $\sum_{c \in \mathcal{C}} d_c = d(\mathcal{C})$. Assume that the maximum flow value is strictly less than $d(\mathcal{C})$. Let $\bar{V}'$ be the subset of $\bar{V}$ obtained from a minimum $\bar{s}$-$\bar{t}$ cut in $\bar{G}$, where $\bar{s} \in \bar{V}'$. Let $\bar{P}' = \bar{P} \setminus \bar{V}'$ and $\bar{R}' = \bar{R} \setminus \bar{V}'$. Let $\bar{A}'$ be the set of arcs that cross the cut: $\bar{A}' = \{(i, j) \in \bar{A} : \ i \in \bar{V}', \ j \notin \bar{V}'\}$. From the assumption and the max-flow-min-cut theorem, it follows that the total capacity of arcs in $\bar{A}'$ is less than $d(\mathcal{C})$. Thus $\bar{A}'$ does not contain all arcs in $\bar{A}_3$ and $\bar{A}'$ contains at least one arc in $\bar{A}_1$. Therefore, the total capacity of arcs in $\bar{A}_1 \cap \bar{A}'$ is strictly less than the total capacity of arcs in $\bar{A}_3 \setminus \bar{A}'$:

$$\sum_{p \in \bar{P}'} Q_1 \bar{\lambda}_p < \sum_{r \in \bar{R}'} \tilde{d}^r \bar{\mu}_r. \tag{13}$$

Consider now time vector $\bar{\tau}$ such that

$$\bar{\tau}_s = \begin{cases} \epsilon + \max_{r \in \bar{R}' \cap R_s} \{\tilde{t}_0^r\}, & \bar{R}' \cap R_s \neq \emptyset, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall s \in \mathcal{S}. \tag{14}$$

Here ϵ is a very small positive value. No first-level route $\bar{P} \setminus \bar{P}'$ can serve second-level route in $\bar{R}'$, as otherwise the minumum $\bar{s}$-$\bar{t}$ cut would have infinite value. Thus, set $P(\bar{\tau})$ does not contain

any route in $\bar{P} \setminus \bar{P}'$, and

$$\sum_{p \in P(\bar{\tau})} Q_1 \bar{\lambda}_p = \sum_{p \in \bar{P}'} Q_1 \bar{\lambda}_p \overset{(13)}{<} \sum_{r \in \bar{R}'} \tilde{d}^r \bar{\mu}_r \leq \sum_{r \in R(\bar{\tau})} \tilde{d}^r \bar{\mu}_r.$$

Thus, solution $(\bar{\lambda}, \bar{\mu})$ violates the precedence constraint (10) characterized by time vector $\bar{\tau}$ and that contradicts the fact that this solution is feasible to (LF2). Then, our assumption about the maximum flow value is wrong, and this value is equal to $d(\mathcal{C})$. We now set each value $\bar{w}_{ps}$, $p \in P$, $s \in \mathcal{S}$, equal to the total flow value along all arcs (p, r) in $\bar{A}_2$ such that $r \in R_s$, and to 0 if there are no such arcs. By construction of graph $\bar{G}$, constraints (3) and (4) are satisfied by solution $(\bar{\lambda}, \bar{\mu}, \bar{w})$, and the latter is feasible to (LF1).

□

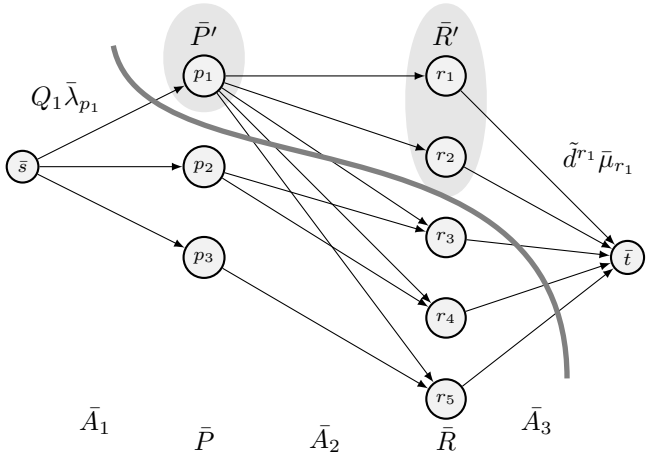


Figure 3: Minimum cut in graph $\bar{G}$ based on the solution in Example 1.

The linear relaxation (LF2) is solved by the column and cut generation procedure described in Section 4. To improve the lower bound for the 2E-VRPTW obtained by this procedure, we use four families of valid inequalities, described in the next section.

3.3 Valid inequalities

To simplify presentation, we use additional auxiliary variables. Let $\tilde{x}_a^p$ be equal to 1 if route $p \in P$ uses arc $a \in A_1$, and 0 otherwise. Let also integer variable x_a, $a \in A_1$, be equal to the number of times urban trucks use arc a: $x_a = \sum_{p \in P} \tilde{x}_a^p \lambda_p$. Let $\tilde{y}_a^r$ be equal to 1 if route $r \in R$ uses arc $a \in A_2$, and 0 otherwise. Let also binary variable y_a, $a \in A_2$, be equal to 1 if a city freighter uses arc a: $y_a = \sum_{r \in R} \tilde{y}_a^r \mu_r$. Finally, let integer variable ν_S, $S \subseteq \mathcal{S}$, be equal to the number of urban trucks visiting at least one satellite in S: $\nu_S = \sum_{p \in P:\, S_p \cap S \neq \emptyset} \lambda_p$.

Rounded capacity cuts were introduced by Laporte and Nobert (1983) for the CVRP. Given a subset $C \subseteq \mathcal{C}$ of customers, value $\lceil \sum_{c \in C} d_c / Q_2 \rceil$ is a lower bound on the number of city freighters required to visit at least one customer in C. Therefore, the rounded capacity cuts (RCC)

$$\sum_{s \in \mathcal{S}} \sum_{a \in \delta_2^-(C)} y_a \geq \left\lceil \frac{\sum_{c \in C} d_c}{Q_2} \right\rceil, \quad C \subseteq \mathcal{C}, \tag{15}$$

are valid for the 2E-VRPTW. Here $\delta_2^-(C)$ is the set of arcs in A_2 incoming to C, i.e. $\delta_2^-(C) = \{(i, j) \in A_2 : i \notin C, j \in C\}$. Constraints (15) are separated using the CVRPSEP package (Lysgaard, 2018) which implements a heuristic by Lysgaard et al. (2004). At most 100 RCCs are added in each cut separation round.

The next family of cuts is obtained by the Chvátal-Gomory rounding of set-partitioning constraints (9), relaxed to $\leq$ inequalities. For a non-negative vector $\alpha \in \mathbb{Q}_+^{|\mathcal{C}|}$ of multipliers, the following rank-1 cut (R1C) is valid:

$$\sum_{r \in R} \left\lfloor \sum_{c \in \mathcal{C}} \alpha_c \tilde{z}_c^r \right\rfloor \mu_r \leq \left\lfloor \sum_{c \in \mathcal{C}} \alpha_c \right\rfloor . \tag{16}$$

An inequality (16) obtained using a vector of multipliers with k positive components is called a k-row rank-1 cut. If all positive components of α are the same, the corresponding inequality is called a subset-row cut. Jepsen et al. (2008) first introduced 3-row subset-row cuts. Pecin et al. (2017a) used k-row subset-row cuts with $k \leq 5$. General k-row rank-1 cuts with $k \leq 5$ were considered by Pecin et al. (2017b). They determined all dominant vectors of multipliers for such cuts: if a k-row rank-1 cut with $k \leq 5$ is violated, at least one rank-1 cut obtained using a dominant vector of multipliers is violated.

Similarly to Sadykov et al. (2020), separation of k-row rank-1 cuts with $k \leq 5$ is performed using a local search heuristic independently for every dominant vector of multipliers. At most 450 R1Cs are added in each cut separation round. We employ the limited memory technique by Pecin et al. (2017a) to reduce the impact of rank-1 cuts on the solution time of the pricing problem.

Let binary variable y_{sa}, $s \in \mathcal{S}$, $a \in A_2$, be equal to 1 if a city freighter starting from satellite s uses arc a: $y_{sa} = \sum_{r \in R_s} \tilde{y}_a^r \mu_r$. The idea of satellite supply inequalities (SSI) introduced by Marques et al. (2020) is to bound from below the number of city freighters started from a satellite outside a subset $S \subseteq \mathcal{S}$ which should visit a subset $C \subseteq \mathcal{C}$ of clients. The right-hand-side of inequality

$$\sum_{s \in \mathcal{S} \setminus S} \sum_{a \in \delta_2^-(C)} y_{sa} \geq \left\lceil \frac{\sum_{c \in C} d_c - Q_1 \lfloor \nu_S \rfloor}{Q_2} \right\rceil \tag{17}$$

is equal to such a lower bound. Constraints (17) are clearly non-linear. Marques et al. (2020) showed how to replace one non-linear SSI by a set of linear constraints. We separate SSI using the heuristic described in this paper. At most 150 SSIs are added in each cut separation round.

Given a satellite $s \in \mathcal{S}$ and a customer $c \in \mathcal{C}$, the visited satellite inequality (VSI) states that there is at least one urban truck visiting satellite s if customer c is delivered by a city freighter coming from s:

$$\sum_{a \in \delta_2^-(\{c\})} y_{sa} \leq \nu_{\{s\}}, \quad s \in \mathcal{S}, \quad c \in \mathcal{C}. \tag{18}$$

These inequalities are first used for the 2ECVRP by Marques et al. (2020). They can be separated by enumerating all pairs $(s, c) \in \mathcal{S} \times \mathcal{C}$. At most 50 VCIs are added in each cut separation round.

3.4 Multi-trip variant

In the multi-trip variant, the first level of the distribution system stays the same but the second level changes because city freighters can perform several trips and can visit more than one satellite. The second level problem becomes the multi-depot multi-trip CVRP with time windowsand it is similar to the multi-depot VRP with interdepot routes considered by (Muter et al., 2014). We consider two graph representations of the second level. In representation (R1) below, we do not keep track of the satellite from which the current trip started. In representation (R2), we keep track of the latest visited satellite. In the former, the second-level graph is smaller, but some valid inequalities described in Section 3.3 cannot be used.

(R1) The second level of the distribution system is represented by directed graph $G_2' = (V_2', A_2')$ where $V_2' = \{0\} \cup \mathcal{S} \cup \mathcal{C}$, $A_2' = A_2 \cup \{0\} \times \mathcal{S} \cup \mathcal{C} \times \{0\}$, and node 0 is the depot of city freighters. A trip in graph G_2' starts in a satellite $s \in \mathcal{S}$, visits some customers in $\mathcal{C}$, and goes empty

to a satellite $s' \in \mathcal{S}$ (possibly $s = s'$) or to depot 0. For each arc $a \in \{0\} \times \mathcal{S} \cup \mathcal{C} \times \{0\}$, its travelling cost f_a and travel time t_a are given.

(R2) In this representation, each customer $c \in \mathcal{C}$ is represented by $|\mathcal{S}|$ nodes, one per satellite, instead of one. Let $\mathcal{C}_s$ be the set of customer nodes for satellite $s \in \mathcal{S}$, and let $\hat{\mathcal{C}} = \bigcup_{s \in \mathcal{S}} \mathcal{C}_s$. The second level is thus represented by directed graph $G_2'' = (V_2'', A_2'')$, where $V_2'' = \{0\} \cup \mathcal{S} \cup \hat{\mathcal{C}}$, $A_2'' = (\bigcup_{s \in \mathcal{S}} A_{2s}'') \cup \{0\} \times \mathcal{S}$, and $A_{2s}'' = \{s\} \times \mathcal{C}_s \cup \{(c, c') \in \mathcal{C}_s^2 : c \neq c'\} \cup \mathcal{C}_s \times \mathcal{S} \cup \mathcal{C}_s \times \{0\}$. We say that an arc $a'' \in A_2''$ projects to an arc $a' \in A_2'$ if their tails and heads correspond to the same satellite or customer. For an arc $a' \in A_2'$, Let $A''(a')$, be the set of arcs in A_2'' projecting to arc a'. A trip in graph G_2'' starts in a satellite $s \in \mathcal{S}$, visits some customers in $\mathcal{C}_s$, and goes empty to a satellite $s' \in \mathcal{S}$ (possibly $s = s'$) or to depot 0.

Let R' be the set of feasible multi-trip second-level routes. Each route consists of the first arc going from depot 0 to a satellite $s \in \mathcal{S}$, and one or several consecutive trips such that the first trip starts at s, each other trip starts at the satellite at which the previous trip has ended, and the last trip finishes at node 0. Let I_r, $r \in R'$, be the set of trips of a multi-trip route. As in the single-trip case, there exists an optimal solution in which each second-level route departs from each node as late as possible. We use the same notation R' for both graph representations (R1) and (R2), as there is a bijection between feasible routes in graphs G_2' and G_2''.

Let $\tilde{z}_c^i$ be equal to 1 if trip $i \in I_r$, $r \in R'$, serves customer $c \in \mathcal{C}$, and 0 otherwise. We have $\tilde{z}_c^r = \sum_{i \in I_r} \tilde{z}_c^i$ for a route $r \in R'$. Let $\tilde{y}_{a'}^r$ be equal to 1 if route $r \in R'$ uses arc $a' \in A_2'$, if representation (R1) is used, or uses an arc in $A''(a')$ if representation (R2) is used. Let $\tilde{d}^i$ be the total amount of freight delivered by trip $i \in I_r$, $r \in R'$: $\tilde{d}^i = \sum_{c \in \mathcal{C}} d_c \tilde{z}_c^i \leq Q_2$. Let $\tilde{t}^i$ and $\tilde{s}^i$ be the departure time of trip $i \in I_r$, and the satellite from which this trip departs.

We denote $I_{rs}(t)$ as the set of trips of route $r \in R'$ starting from satellite s before time t: $I_{rs}(t) = \{i \in I_r : \tilde{s}^i = s, \tilde{t}^i < t\}$. Given a time vector $\tau = (\tau_s)_{s \in \mathcal{S}}$ and a route $r \in R'$, let $I_r(\tau) = \cup_{s \in \mathcal{S}} I_{rs}(\tau_s)$. Then, two-level precedence constraints (10) can be rewritten for the multi-depot variant of the problem:

$$\sum_{p \in P(\tau)} Q_1 \lambda_p - \sum_{r \in R'} \sum_{i \in I_r(\tau)} \tilde{d}^i \mu_r \geq 0, \qquad \tau \in \mathcal{T}. \tag{19}$$

As values $\tilde{z}_c^r$, $c \in \mathcal{C}$, and $\tilde{y}_a^r$, $a \in A_2$, are defined for all multi-trip second-level routes, valid inequalities (15) and (16) can directly be used in the multi-trip case.

The branch-cut-and-price algorithm presented in the next section has two variants for the multi-trip case, depending on the graph representation used. If representation (R2) is used, for a given triple (r, s, a), $r \in R'$, $s \in \mathcal{S}$, $a \in A_2$, we are able to determine value $\tilde{y}_{sa}^r$, which is equal to 1 if an arc $a'' \in A_{2s}''$ projecting to a is used by route r. Then, variables $y_{sa} = \sum_{r \in R'} \tilde{y}_{sa}^r \mu_r$ are available, and we can use valid inequalities (17) and (18).

4 Branch-Cut-and-Price algorithm

The LP relaxation (LF2) of formulation (F2) together with valid inequalities (15), (16), (17), and (18) is solved by a column and cut generation approach. The first-level and second-level route variables are generated by solving the pricing problems which we describe in Section 4.1. We also show how two-level precedence constraints (10) affect the structure of the pricing problems. In Section 4.2, we introduce a separation algorithm for TLPC (10). We give a brief description of the remaining components of the branch-cut-and-price algorithm in Section 4.3. Finally, in Section 4.4, we present the post-processing procedure that tries to exactly synchronize urban trucks and city freighters.

4.1 Pricing problems

Consider formulation (LF2) with a restricted number of variables and constraints. We denote it as (RLF2). Let $(\bar{\pi}, \bar{\zeta}, \bar{\rho}, \bar{\xi}, \bar{\eta}, \bar{\theta})$ be an optimal dual solution of (RLF2), corresponding to constraints (9), (10), (15), (16), (17), and (18) respectively. We say that a constraint is active if its value is non-zero in the dual solution. Let E be the set of active TLPC, and τ^e defines cut $e \in E$ with dual value $\bar{\zeta}_e$. Let N be the set of active RCC, and C^n defines cut $n \in N$ with dual value $\bar{\rho}_n$. Let M be the set of active R1C, and α^m defines cut $m \in M$ with dual value $\bar{\xi}_m$. Let H be the set of active SSI, and (S^h, C^h, β^h) defines cut $h \in H$ with dual value $\bar{\eta}_h$, where β^h is the coefficient of variable ν_S in this SSI.

4.1.1 First level pricing problem

The reduced cost of a first-level route $p \in P$ is equal to

$$\sum_{a \in A_1} f_a \tilde{x}^p_a - \sum_{e \in E:\, p \in P(\tau^e)} Q_1 \bar{\zeta}_e + \sum_{h \in H:\, S_p \cap S^h \neq \emptyset} \beta^h \bar{\eta}_h + \sum_{s \in S_p} \sum_{c \in \mathcal{C}} \bar{\theta}_{sc}. \tag{20}$$

We cannot express reduced cost (20) as a linear combination of reduced costs on arcs in A_1. Thus, solving the first-level pricing problem as a standard resource constrained shortest path problem (RCSPP) is not possible. Here, we take advantage of the fact that the number of depots and satellites in the literature instances is not large. We enumerate all feasible first-level routes before starting the column and cut generation. However, we cannot include all corresponding first-level route variables λ to (LF2), as their number can exceed 100,000. Instead, as proposed by Contardo and Martinelli (2014), we solve the first-level pricing problem by inspection of enumerated routes. The reduced cost of every enumerated route is updated based on the current dual solution, and routes with the smallest reduced costs are selected.

4.1.2 Second-level single-trip pricing problem

This problem can be decomposed in $|\mathcal{S}|$ independent subproblems, one per satellite. Given a satellite $s \in \mathcal{S}$, the reduced cost of a second-level single-trip route $r \in R_s$ is calculated as

$$\begin{aligned} &\sum_{a \in A_2} f_a \tilde{y}^r_a - \sum_{c \in \mathcal{C}} \sum_{a \in \delta_2^-(\{c\})} \bar{\pi}_c \tilde{y}^r_a - \sum_{n \in N} \sum_{a \in \delta_2^-(C^n)} \bar{\rho}_n \tilde{y}^r_a + \sum_{c \in \mathcal{C}} \sum_{a \in \delta_2^-(\{c\})} \bar{\theta}_{sc} \tilde{y}^r_a \\ &\quad - \sum_{h \in H:\, s \notin S^h} \sum_{a \in \delta_2^-(C^h)} \bar{\eta}_h \tilde{y}^r_a + \sum_{e \in E:\, r \in R(\tau^e)} \bar{\zeta}_e \tilde{d}^r + \sum_{m \in M} \left\lfloor \sum_{c \in \mathcal{C}} \alpha^m_c \tilde{z}^r_c \right\rfloor \bar{\xi}_m. \end{aligned} \tag{21}$$

Consider first the case without active R1Cs, i.e. $M = \emptyset$. Reduced cost (21) cannot be expressed as a linear combination of reduced costs on arcs in A_2 because of the term coming from active TLPCs. Indeed, for each $e \in E$, the reduced cost of path $r \in R_s$ is increased by $\bar{\zeta}_e \tilde{d}^r$ if r departs from satellite s strictly before time moment τ^e_s. Therefore, we define a graph $\mathcal{G}^s$, which is extended from graph G_2, to express the contribution of TLPC to (21) as a linear combination of reduced costs on arcs in $\mathcal{G}^s$. Then, the second-level pricing subproblem corresponding to satellite s can be formulated as a standard RCSPP in extended graph $\mathcal{G}^s$.

Let $\bar{T}^s = (\bar{t}^s_0, \bar{t}^s_1, \ldots, \bar{t}^s_{\bar{n}(s)})$, $\bar{t}^s_0 = l_s + \sigma_s$, be the ordered set of different time moments τ^e_s for all $e \in E$, augmented by value $l_s + \sigma_s$ if necessary. All values τ^e_s which are less than $l_s + \sigma_s$ are ignored. Set of nodes in $\mathcal{G}^s$ is defined as $\mathcal{V}^s \cup \mathcal{V}^{\mathcal{C}} \cup \{v_{\text{source}}, v_{\text{sink}}\}$, where node $v^s_k \in \mathcal{V}^s$, $0 \leq k \leq \bar{n}(s)$, corresponds to the situation in which city freighter is available at time $\bar{t}^s_k$ at satellite s, and node $v^{\mathcal{C}}_{cq} \in \mathcal{V}^{\mathcal{C}}$, $c \in \mathcal{C}$, $d_c \leq q \leq Q_2$, corresponds to the situation in which vehicle is coming to customer c with load q. Set of arcs in $\mathcal{G}^s$ is defined as $\{(v_{\text{source}}, v^s_0)\} \cup \mathcal{A}^s \cup \mathcal{A}^{s \to \mathcal{C}} \cup \mathcal{A}^{\mathcal{C}} \cup \mathcal{A}^{\text{sink}}$. Arcs in $\mathcal{A}^s = \{(v^s_{k-1}, v^s_k)\}_{1 \leq k \leq \bar{n}(s)}$ connect consecutive nodes in $\mathcal{V}^s$. Arcs in $\mathcal{A}^{s \to \mathcal{C}} = \{(v, v')\}_{v \in \mathcal{V}^s, v' \in \mathcal{V}^{\mathcal{C}}}$ connect all satellite nodes to all customer nodes. Arcs

in $\mathcal{A}^{\mathcal{C}} = \{(v^{\mathcal{C}}_{c,q}, v^{\mathcal{C}}_{c',q-d_c})\}_{c,c' \in \mathcal{C},\, c \neq c',\, d_c + d_{c'} \leq q \leq Q_2}$ interconnect customer nodes. Finally, arcs in $\mathcal{A}^{\text{sink}} = \{(v^{\mathcal{C}}_{c,d_c}, v_{\text{sink}})\}_{c \in \mathcal{C}}$ connect customer nodes to the sink. Each arc in $\mathcal{A}^{s \to \mathcal{C}}$ project into the corresponding arc in A_2 between satellite s and a customer. Each arc in $\mathcal{A}^{\mathcal{C}}$ projects into the corresponding arc in A_2 between two customers. Each arc in $\mathcal{A}^{\text{sink}}$ projects into the corresponding arc in A_2 between a customer and satellite s.

We now formulate the pricing problem as a RCSPP in graph $\mathcal{G}^s$. Time is the only resource. The time consumption of arc a in graph $\mathcal{G}^s$ is equal to the sum of travel time $t_{a'}$ and the service time of the satellite or customer corresponding to the tail of arc $a' \in A_2$ to which a projects. If a does not project to an arc in A_2, the time consumption is zero. Bounds on the accumulated time consumption are given on nodes. These bounds are $[0, 0]$ for v_{source}, $[\bar{t}^s_k, u_s]$ for $v^s_k \in \mathcal{V}^s$, $[l_c, u_c]$ for nodes $v^{\mathcal{C}}_{cq} \in \mathcal{V}^{\mathcal{C}}$, and $[l_s + \sigma_s, u_s]$ for v_{sink}. The time resource is disposable, as defined by Pessoa et al. (2020): accumulated time consumption of a path in $\mathcal{G}^s$ at a node v is adjusted to the lower bound on the accumulated time consumption at v, if the former is smaller than the latter. Figure 4 depicts an example of an extended graph $\mathcal{G}^s$. The reduced cost of each arc a in graph $\mathcal{G}^s$ is equal to the sum of the travelling cost $f_{a'}$ of arc $a' \in A_2$ to which a projects, the total coefficient of $\tilde{y}^r_a$ in (21), and the contribution of TLPCs. The reduced cost of an arc a is zero if a does not project to an arc in A_2. Contribution of active TLPCs to the reduced cost of each arc $(v^s_k, v^{\mathcal{C}}_{c,q}) \in \mathcal{A}^{s \to \mathcal{C}}$, $0 \leq k \leq \bar{n}(s)$, $c \in \mathcal{C}$, $d_c \leq q \leq Q_2$, is equal to $q \cdot \sum_{e \in E:\, \tau^e_s > \bar{t}_k} \bar{\zeta}_e$. Contribution of active TLPC to arcs which are not in $\mathcal{A}^{s \to \mathcal{C}}$ is zero.

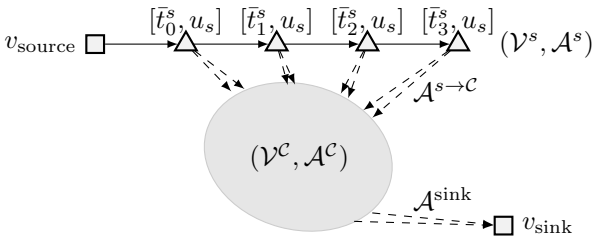


Figure 4: Example of extended graph $\mathcal{G}^s$ for pricing second-level single-trip routes

The RCSPP just formulated is solved by the bucket graph based labeling algorithm proposed by Sadykov et al. (2020). The modification of this algorithm for the case with active limited-memory rank-1 cuts is also described in the latter paper by following ideas from (Jepsen et al., 2008; Pecin et al., 2017a).

4.1.3 Second-level multi-trip pricing problem

This problem cannot be decomposed in subproblems. It is thus solved in one run. The reduced cost of a second-level multi-trip route $r \in R'$ is calculated as

$$\begin{aligned} &\sum_{a \in A'_2} f_a \tilde{y}^r_a - \sum_{c \in \mathcal{C}} \sum_{a \in \delta^-_2(\{c\})} \bar{\pi}_c \tilde{y}^r_a - \sum_{n \in N} \sum_{a \in \delta^-_2(C^n)} \bar{\rho}_n \tilde{y}^r_a + \sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{S}} \sum_{a \in \delta^-_2(\{c\})} \bar{\theta}_{sc} \tilde{y}^r_{sa} \\ &- \sum_{h \in H} \sum_{s \in \mathcal{S} \setminus S^h} \sum_{a \in \delta^-_2(C^h)} \bar{\eta}_h \tilde{y}^r_{sa} + \sum_{e \in E} \sum_{i \in I_r(\tau^e)} \bar{\zeta}_e \tilde{d}^i + \sum_{m \in M} \left\lfloor \sum_{c \in \mathcal{C}} \alpha^m_c \tilde{z}^r_c \right\rfloor \bar{\xi}_m, \end{aligned} \tag{22}$$

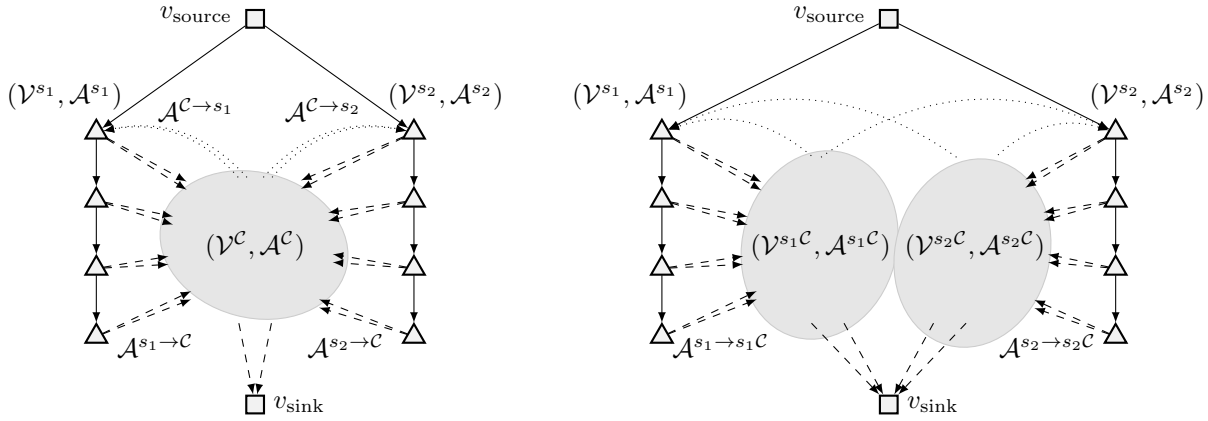
If the problem is solved using graph representation (R1), values $\tilde{y}^r_{sa}$ are not available. Thus, SSIs and VSIs cannot be used and the corresponding terms in (22) are skipped.

Similarly to the single-trip case and depending on the representation used, we extend graph G'_2 or graph G''_2 to $\mathcal{G}'$ or $\mathcal{G}''$ respectively. We first describe graph $\mathcal{G}'$ extended from representation (R1). Set of nodes in $\mathcal{G}'$ is defined as $\bigcup_{s \in \mathcal{S}} \mathcal{V}^s \cup \mathcal{V}^{\mathcal{C}} \cup \{v_{\text{source}}, v_{\text{sink}}\}$, where $\mathcal{V}^s$, $s \in \mathcal{S}$, and $\mathcal{V}^{\mathcal{C}}$ are defined as in the single-trip case. Set of arcs in $\mathcal{G}'$ is defined as

$$\bigcup_{s \in \mathcal{S}} \left(\mathcal{A}^s \cup \mathcal{A}^{s \to \mathcal{C}} \cup \mathcal{A}^{\mathcal{C} \to s}\right) \cup \mathcal{A}^{\mathcal{C}} \cup \mathcal{A}^{\text{source}} \cup \mathcal{A}^{\text{sink}},$$

where $\mathcal{A}^s$, $\mathcal{A}^{s\to\mathcal{C}}$, $s \in \mathcal{S}$, $\mathcal{A}^{\mathcal{C}}$, and $\mathcal{A}^{\text{sink}}$ are defined as in the single-trip case. Given satellite $s \in \mathcal{S}$, arcs in $\mathcal{A}^{\mathcal{C}\to s} = \{(v^{\mathcal{C}}_{c,d_c}, v^s_0)\}_{c\in\mathcal{C}}$ connect **some** customer nodes to the initial satellite s node, . Arcs in $\mathcal{A}^{\text{source}} = \{(v_{\text{source}}, v^s_0)\}_{s\in\mathcal{S}}$ connect the source to the initial satellite nodes. Projection of arcs in $\mathcal{A}^{s\to\mathcal{C}}$ and in $\mathcal{A}^{\mathcal{C}}$ is the same as in the single-trip case. Each arc in $\mathcal{A}^{\text{source}}$ projects into the corresponding arc in A'_2 between the depot and a satellite. Each arc in $\mathcal{A}^{\text{sink}}$ projects into the corresponding arc in A'_2 between a customer and the depot. Figure 5a depicts the structure of graph $\mathcal{G}'$.

The formulation of the RCSPP in graph $\mathcal{G}'$ is similar as the one in graph $\mathcal{G}^s$. Bounds on the accumulated time consumption are the same for nodes in $\bigcup_{s\in\mathcal{S}} \mathcal{V}^s \cup \mathcal{V}^{\mathcal{C}}$. Bounds for nodes $\{v_{\text{source}}, v_{\text{sink}}\}$ correspond to time window when the depot is open. The resource consumption of arc a in graph $\mathcal{G}'$ is equal to the sum of travel time $t_{a'}$ and the service time of the satellite or customer corresponding to the tail of arc $a' \in A'_2$ to which a projects. The reduced cost of each arc a in graph $\mathcal{G}'$ is equal to the sum of the travelling cost $f_{a'}$ of arc $a' \in A'_2$ to which a projects, the total coefficient of $\tilde{y}^r_a$ in (22), and the contribution of TLPCs. Contribution of active TLPCs to arcs in $\mathcal{A}^{s\to\mathcal{C}}$ is the same as in the single-trip case. Contribution of active TLPCs to other arcs is zero.



(a) Structure of graph $\mathcal{G}'$

(b) Structure of graph $\mathcal{G}''$

Figure 5: Examples of extended graphs for pricing second-level multi-trip routes

We now describe graph $\mathcal{G}''$ extended from representation (R2). The set of nodes in $\mathcal{G}''$ is the same as in $\mathcal{G}'$, except that customer nodes are duplicated for each satellite: $\bigcup_{s\in\mathcal{S}} \left(\mathcal{V}^s \cup \mathcal{V}^{s\mathcal{C}}\right) \cup \{v_{\text{source}}, v_{\text{sink}}\}$. Each node $v^{s\mathcal{C}}_{cq} \in \mathcal{V}^{s\mathcal{C}}$, $s \in \mathcal{S}$, $c \in \mathcal{C}$, $d_c \le q \le Q_2$, corresponds to the situation in which a city freighter is coming to customer c with load q, and the last visited satellite is s. Set of arcs in $\mathcal{G}''$ is defined as

$$\bigcup_{s\in\mathcal{S}} \left(\bigcup_{s'\in\mathcal{S}} \mathcal{A}^{s\mathcal{C}\to s'} \cup \mathcal{A}^s \cup \mathcal{A}^{s\to s\mathcal{C}} \cup \mathcal{A}^{s\mathcal{C}} \cup \mathcal{A}^{s\to\text{sink}} \right) \cup \mathcal{A}^{\text{source}},$$

where $\mathcal{A}^s$, $s \in \mathcal{S}$, and $\mathcal{A}^{\text{source}}$ are defined as for graph $\mathcal{G}'$. Given satellites $s, s' \in \mathcal{S}$, arcs in $\mathcal{A}^{s\mathcal{C}\to s'} = \{(v^{s\mathcal{C}}_{c,d_c}, v^{s'}_0)\}_{c\in\mathcal{C}}$ connect some customer nodes associated to satellite s to the initial satellite s' nodes. Arcs in $\mathcal{A}^{s\to s\mathcal{C}} = \{(v, v')\}_{v\in\mathcal{V}^s, v'\in\mathcal{V}^{s\mathcal{C}}}$, $s \in \mathcal{S}$, connect all nodes of satellite s to all customer nodes associated to s. Arcs in $\mathcal{A}^{s\mathcal{C}} = \{(v^{s\mathcal{C}}_{c,q}, v^{s\mathcal{C}}_{c',q-d_c})\}_{c,c'\in\mathcal{C},\, c\neq c',\, d_c+d_{c'}\le q\le Q_2}$, $s \in \mathcal{S}$, interconnect customer nodes associated to the same satellite s. Finally, arcs in $\mathcal{A}^{s\to\text{sink}} = \{(v^{s\mathcal{C}}_{c,d_c}, v_{\text{sink}})\}_{c\in\mathcal{C}}$, $s \in \mathcal{S}$, connect customer nodes associated to s to the sink. Projection of arcs in graph $\mathcal{G}''$ to arcs in A'_2 is similar to the projection of arcs in graph $\mathcal{G}'$. Figure 5b depicts the structure of graph $\mathcal{G}'$.

The formulation of the RCSPP in graph $\mathcal{G}'$ is similar to the one in graph $\mathcal{G}''$. Bounds on the accumulated time consumption are the same for nodes in $\bigcup_{s\in\mathcal{S}} \mathcal{V}^s \cup \{v_{\text{source}}, v_{\text{sink}}\}$. Bounds for each customer node in $\mathcal{V}^{s\mathcal{C}}$, $s \in \mathcal{S}$, are equal to the start and the end of time window of

the corresponding customer. The time consumption of arc a in graph $\mathcal{G}''$ is equal to the sum of travel time $t_{a'}$ and the service time of the satellite or customer corresponding to the tail of arc $a' \in A'_2$ to which a projects. The reduced cost of each arc a in graph $\mathcal{G}''$ is equal to the sum of the travelling cost $f_{a'}$ of arc $a' \in A'_2$ to which a projects plus the total coefficient of $\tilde{y}^r_{a'}$ in (22), the contribution of TLPC , and the contribution of SSI and VSI. Contribution of active TLPC to arcs in $\mathcal{A}^{s \to s\mathcal{C}}$, $s \in \mathcal{S}$, is the same as in the single-trip case. Contribution of active TLPC to other arcs is zero. Contribution of active SSI and VSI to arc $a \in \mathcal{A}^{s\mathcal{C}}$, $s \in \mathcal{S}$, is equal to the total coefficient of $\tilde{y}^r_{a's}$, where a' the arc in A'_2 to which a projects.

4.2 TLPC separation algorithm

Given a solution to formulation (RLF2), the TLPC separation algorithm searches for violated TLPCs. These constraints are essential to the formulation. Thus, the separation algorithm should find a violated constraint when it exists. Our algorithm first finds the most violated constraint, and then it tries heuristically to obtain other violated constraints. We now present the algorithm for the single-trip case. Extension to the multi-trip case is obvious after replacing second-level routes by second-level trips.

Our separation algorithm is based on the proof of Proposition 1. Given fractional or integer solution $(\bar{\lambda}, \bar{\mu})$, we construct graph $\bar{G}$, as described in the proof. We then find a minimum cut in this graph. If the value of this cut is equal to $d(\mathcal{C})$, then no TLPC violated by $(\bar{\lambda}, \bar{\mu})$ exists, and the algorithm stops. If the value of the cut is strictly smaller than $d(\mathcal{C})$, we obtain set $\bar{P}'$ of first-level routes and set $\bar{R}'$ of second-level routes as defined in the proof. Vector $\bar{\tau}$ characterising the most violated constraint is then calculated according to formula (14).

If a violated TLPC is found, we try to obtain other violated constraints. For this, we define the directed graph $\vec{G} = (\vec{V}, \vec{A})$ that represents precedence relations between first-level routes. Remember that $\bar{P} = \{p \in P : \bar{\lambda}_p > 0\}$ and $\bar{R} = \{r \in R : \bar{\mu}_r > 0\}$ are the sets of first-level and second-level routes participating in the solution. Let also $\vec{T}^s = (\vec{t}^s_1, \vec{t}^s_2, \ldots, \vec{t}^s_{\vec{n}(s)})$ be the ordered set of different time moments at which first-level routes in $\bar{P}$ depart from satellite $s \in \mathcal{S}$. We have $\vec{V} = \vec{V}^P \cup \left(\cup_{s\in\mathcal{S}} \vec{V}^s\right)$, where node $\vec{v}^P_p \in \vec{V}^P$, $p \in \bar{P}$, corresponds to a first-level route participating in the solution, and node $\vec{v}^s_k$, $1 \le k \le \vec{n}(s)$, corresponds to a visit of a first-level route in the solution to satellite s. Set of arcs $\vec{A}$ is defined as $\left(\cup_{p\in\bar{P}} \vec{A}^p\right) \cup \left(\cup_{s\in\mathcal{S}} \vec{A}^s\right)$. Subset $\vec{A}^p$ of arcs connects node $\vec{v}^P_p$ with the corresponding visits or route p to satellites: $\vec{A}^p = \left\{(\vec{v}^P_p, \vec{v}^s_{k(p,s)}), (\vec{v}^s_{k(p,s)}, \vec{v}^P_p)\right\}_{s\in S_p}$, where $k(p,s)$ is the index in $\vec{T}^s$ of the time moment when route p departs from satellite s. Subset $\vec{A}^s$ of arcs connects consecutive nodes corresponding to visits of routes to satellite s in the reverse chronological order: $\vec{A}^s = \{\vec{v}^s_{k+1}, \vec{v}^s_k\}_{1\le k<\vec{n}(s)}$. As an example, graph $\vec{G}$ corresponding to Example 1 is depicted in Figure 6.

Using graph $\vec{G}$, for each first-level route $p \in \bar{P}'$, we find set $\vec{P}^p$ of first-level routes in $\bar{P}$ which "precede" p. Set $\vec{P}^p$ corresponds to all nodes in $\vec{v}^P_p$ which are reachable from node $\vec{v}^P_p$. In the example in Figure 6, we have $\vec{P}^{p_1} = \{p_1\}$, $\vec{P}^{p_2} = \{p_1, p_2\}$, $\vec{P}^{p_3} = \{p_1, p_3\}$, and $\vec{P}^{p_4} = \{p_1, p_2, p_4\}$. For each set $\vec{P}^p$, $p \in \bar{P}'$, we then find the vector τ^p such that $\bar{P} \cap P(\tau^p) = \vec{P}^p$ and set $\bar{R} \cap R(\tau^p)$ is as large as possible so that the violation of the corresponding TLPC is maximized. Component τ^p_s, $s \in \mathcal{S}$, of such vector τ^p, $p \in \bar{P}'$, is calculated as $\tau^p_s = \min\left\{u_s, \min_{p\in\bar{P}\setminus\bar{P}':\, s\in S_p}\{\vec{t}^s_{k(p,s)}\}\right\}$. For each $p \in \bar{P}'$ we verify whether constraint (10) characterized by τ^p is violated. All violated TLPCs are then added to formulation (RLF2).

4.3 The overall algorithm

The structure of the overall branch-cut-and-price (BCP) algorithm we use is similar to the one of the BCP algorithm described in (Sadykov et al., 2020). Thus, we only present the main components used. We give more details only on branching strategies, which are problem-specific.

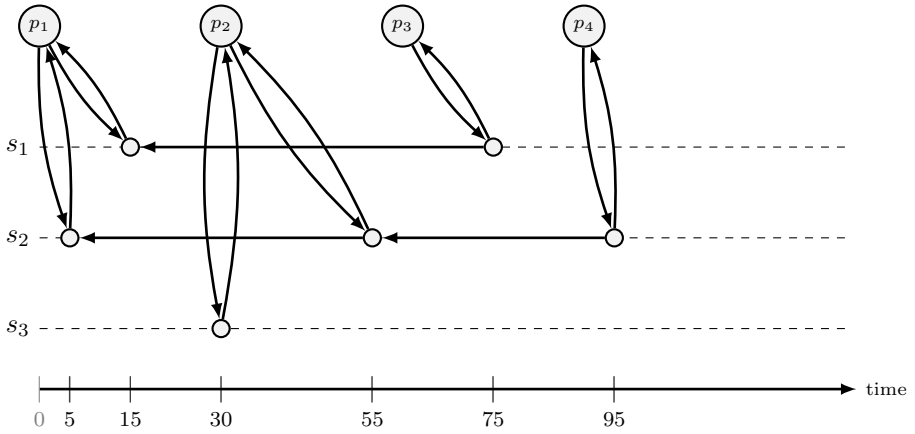


Figure 6: Graph $\vec{G}$ corresponding to Example 1

As it was mentioned before, the pricing problems, which are RCSPPs, are solved by the bucket graph based labeling algorithm, proposed by (Sadykov et al., 2020). We also use the same heuristic variants of this algorithm. Automatic dual pricing smoothing stabilization, proposed by Pessoa et al. (2018), improves the convergence of column generation. The bucket arc elimination procedure, proposed by (Sadykov et al., 2020), decreases the size of graphs for RCSP pricing problems using the current primal-dual gap. For each pricing problem, the elementary route enumeration technique , proposed by Baldacci et al. (2008), tries to enumerate all elementary routes with reduced cost smaller than the current primal-dual gap. If the enumeration succeeds, the pricing subproblem is solved by inspection in future column generation iterations, similarly to Contardo and Martinelli (2014). If the total number of enumerated first-level and second-level routes becomes small, all the routes are added to formulation (F2) and the latter is solved by the MIP solver. The parameterisation of the BCP algorithm is the same as described in (Sadykov et al., 2020).

We now describe how the branching is performed. Suppose that the solution $(\bar{\lambda}, \bar{\mu})$ obtained by column and cut generation at a node of the branch-and-bound tree is fractional. As described in Section 3.3, values $\bar{x}$, $\bar{y}$, and $\bar{\nu}$ are computed based on $\bar{\lambda}$ and $\bar{\mu}$. We branch on (i) the number of urban trucks $\sum_{p \in P} \bar{\lambda}_p$; (ii) the number of city freighters $\sum_{r \in R} \bar{\mu}_r$ or $\sum_{r \in R'} \bar{\mu}_r$; (iii) the number of urban trucks visiting a subset of satellites ν_S, $S \subseteq \mathcal{S}$; (iv) the use of first-level arcs $\bar{x}_a$, $a \in A_1$, by urban trucks; and (v) the use of second-level arcs $\bar{y}_a$, $a \in A_2$ or $a \in A_2'$, by city freighters. In the single-trip case, we also branch on the number of city freighters starting from a satellite $\sum_{r \in R:\, \check{s}_r = s} \bar{\mu}_r$, $s \in \mathcal{S}$. The multi-phase strong branching procedure, described in Sadykov et al. (2020), selects the branching candidate.

At each node in the branch-and-bound tree, an heuristic based on an artificial primal bound and the elementary route enumeration, similar to the one used by Pessoa et al. (2009) and Marques et al. (2020), looks for an integer solution. In an iterative procedure, it decreases the artificial primal bound in order to divide the primal-dual gap by two in each iteration. Then, it performs elementary route enumeration for each pricing subproblem. The iterative procedure stops when the enumeration succeeds for all pricing subproblems. Afterward, it picks 5'000 elementary routes with the smallest reduced cost and add them to the master problem. Finally, IBM CPLEX MIP solver to solve the resulting problem with the time limit of 10 seconds. The polishing heuristic (Rothberg, 2007) implemented in CPLEX is activated.

4.4 Post-processing

The post-processing phase seeks to synchronize the arrival of urban trucks and city freighters at satellites, if possible. In other words, given an optimal solution (λ^*, μ^*) to (F2), it modifies the arrival and departure times of routes such that the storage usage while transferring the

freight from urban trucks to city freighters is minimized. For the sake of brevity, we focus on the multi-trip variant. Adjustments for the single-trip variant are straightforward.

Let $P^* = \{p \in P : \lambda_p^* \geq 1\}$ and $R^* = \{r \in R' : \mu_r^* = 1\}$. Let also P_i^* be the set of routes in P^* which can serve trip $i \in I_r$, $r \in R^*$: $P_i^* = \{p \in P^* : \exists k, 1 \leq k < n(p),\ v_k^p = \tilde{s}^i,\ \tilde{t}_k^p \leq \tilde{t}^i\}$. We denote as $k^1(p,i)$ the index number of visit to satellite $\tilde{s}^i$, $i \in I_r$, $r \in R^*$, in route $p \in P_i^*$. We also denote as $k^2(r,i)$ the index number of visit to satellite $\tilde{s}^i$, $i \in I_r$, in route $r \in R^*$ when starting trip i.

Let binary variable χ_{pji}, $i \in I_r$, $r \in R^*$, $p \in P_i^*$, $1 \leq j \leq \lambda_p^*$, be equal to one if trip i is served by the j-th vehicle following first-level route p. Let variable γ_{pji}, $i \in I_r$, $r \in R^*$, $p \in P_i^*$, $1 \leq j \leq \lambda_p^*$, be equal to the fraction of the load of trip i served by the j-th vehicle following first-level route p. Let variable Δ_{pji}, $i \in I_r$, $r \in R^*$, $p \in P_i^*$, $1 \leq j \leq \lambda_p^*$, be equal to the time elapsed between the departure of the j-th vehicle on route p and the departure of trip i in satellite $\tilde{s}^i$, if trip i is served by this vehicle. If route r leaves satellite $\tilde{s}^i$ before departure of the j-th vehicle on route p or trip i is not served by this vehicle, then $\Delta_{pji} = 0$. Let variables ϕ_{pjk}^{1-} and ϕ_{pjk}^{1+} be equal to the arrival and departure times of the j-th first-level vehicle on route $p \in P^*$ at node $v_k^p \in \{0\} \cup \mathcal{S}$, $0 \leq k \leq n(p)$. Let variable ϕ_{rk}^{2+} be the departure time of second-level route $r \in R^*$ at node $v_k^r \in \mathcal{S} \cup \mathcal{C}$, $0 \leq k \leq n(r)$. The following mixed integer linear program minimizes the total time during which satellites store freight.

$$(PP) \equiv \min \quad \sum_{r \in R^*} \sum_{i \in I_r} \sum_{p \in P_i^*} \sum_{j=1}^{\lambda_p^*} \Delta_{pji} \tag{23}$$

$$\text{s.t.} \quad \gamma_{pji} \leq \chi_{pji} \qquad r \in R^*,\ i \in I_r,\ p \in P_i^*,\ 1 \leq j \leq \lambda_p^* \tag{24}$$

$$\sum_{r \in R^*} \sum_{i \in I_r:\ p \in P_i^*} \tilde{d}^i \gamma_{pji} \leq Q_1 \qquad p \in P^*,\ 1 \leq j \leq \lambda_p^* \tag{25}$$

$$\sum_{p \in P_i^*} \sum_{j=1}^{\lambda_p^*} \gamma_{pji} = 1 \qquad r \in R^*,\ i \in I_r \tag{26}$$

$$\Delta_{pji} - \phi_{r,k^2(r,i)}^{2+} + \phi_{p,j,k^1(p,i)}^{1+} + (u_{\tilde{s}^i} - l_{\tilde{s}^i}) \cdot (1 - \chi_{pji}) \geq 0 \qquad r \in R^*,\ i \in I_r,\ p \in P_i^*,\ 1 \leq j \leq \lambda_p^* \tag{27}$$

$$\phi_{r,k^2(r,i)}^{2+} - \phi_{p,j,k^1(p,i)}^{1-} - \sigma_{\tilde{s}^i} + (u_{\tilde{s}^i} - l_{\tilde{s}^i}) \cdot (1 - \chi_{pji}) \geq 0 \qquad r \in R^*,\ i \in I_r,\ p \in P_i^*,\ 1 \leq j \leq \lambda_p^* \tag{28}$$

$$\phi_{p,j,k-1}^{1+} + t_{(v_{k-1}^p, v_k^p)} \leq \phi_{p,j,k}^{1-} \qquad p \in P^*,\ 1 \leq k \leq n(p),\ 1 \leq j \leq \lambda_p^* \tag{29}$$

$$\phi_{p,j,k}^{1-} + \sigma_{v_k^p} \leq \phi_{p,j,k}^{1+} \qquad p \in P^*,\ 1 \leq k < n(p),\ 1 \leq j \leq \lambda_p^* \tag{30}$$

$$\phi_{r,k-1}^{2+} + t_{(v_{k-1}^r, v_k^r)} + \sigma_{v_k^r} \leq \phi_{r,k}^{2+} \qquad r \in R^*,\ 1 \leq k \leq n(r) \tag{31}$$

$$l_{v_k^p} \leq \phi_{p,j,k}^{1-} \leq u_{v_k^p} \qquad p \in P^*,\ 0 \leq k \leq n(p),\ 1 \leq j \leq \lambda_p^* \tag{32}$$

$$l_{v_k^r} \leq \phi_{p,k}^{2+} - \sigma_{v_k^r} \leq u_{v_k^r} \qquad r \in R^*,\ 0 \leq k \leq n(r) \tag{33}$$

$$\chi_{pji} \in \{0,1\} \qquad r \in R^*,\ i \in I_r,\ p \in P_i^*,\ 1 \leq j \leq \lambda_p^* \tag{34}$$

$$0 \leq \gamma_{pji} \leq 1 \qquad r \in R^*,\ i \in I_r,\ p \in P_i^*,\ 1 \leq j \leq \lambda_p^* \tag{35}$$

$$\Delta_{pji} \geq 0 \qquad r \in R^*,\ i \in I_r,\ p \in P_i^*,\ 1 \leq j \leq \lambda_p^* \tag{36}$$

The objective function (23) minimises the total number of time units during which freight is stored at satellites. Constraints (24) link variables χ and γ. Constraints (25) ensure that the capacity of the first-level vehicles is satisfied. Constraints (26) ensures that each trip of city freighters receives the desired amount of freight. Constraints (27) compute the values of variables Δ: if a trip i is served by the j-th first-level vehicle on path p, then Δ_{pji} is not smaller than the difference between departure times of trip i and the vehicle from satellite $\tilde{s}^i$. In these and in the next constraints, expression $(u_{\tilde{s}^i} - l_{\tilde{s}^i})$ acts as a big-M value. Constraints (28) ensures that each first-level route p arrives and completes its service time before all departures of the second-level trip it serves. Constraints (29)–(31) guarantee that arrival and departure times of first-level and second-level vehicles are compatible with visited order of nodes. Constraints (32) and (33) ensure that all time windows are satisfied. If the optimal solution value is zero, then we can synchronize urban trucks and city freighters, and solution (λ^*, μ^*) is feasible and optimal for the case in which satellites do not have storage. Otherwise, the value of solution (λ^*, μ^*) provides a lower bound for the case without storage.

If storage is not needed, then we can further check if freight consolidation can be avoided, i.e. if each second-level trip can be served by only one first-level vehicle. To do it, we should verify if there exists a feasible solution to formulation (25)–(34), in which variables Δ are fixed to 0 in constraints (27), and variables γ_{pji} are replaced by variables χ_{pji} in constraints (25)–(26).

5 Computational results

The implementation of the proposed algorithm was done in C++ language. We used the following third-party libraries and codes:

- BaPCod C++ library (Vanderbeck et al., 2019) which implements the BCP framework;
- C++ code, developed by Sadykov et al. (2020), which implements the bucket graph based labeling algorithm, bucket arc elimination procedure, elementary route enumeration, and the separation algorithm for R1Cs;
- CVRPSEP C++ library (Lysgaard, 2018) which implements heuristic separation of RCCs;
- IBM CPLEX Optimizer version 12.10 as the LP solver in column generation, as the solver for the enumerated MIPs, and as the solver for the MIP post-processing.

Experiments were run on a 2 Dodeca-core Haswell Intel Xeon E5-2680 v3 servers at 2.5 GHz. Every server has 128 Go of RAM. Each instance is solved on a single thread.

5.1 Literature Instances

In this paper, we experiment our algorithm on single-trip instances proposed by Dellaert et al. (2019) and multi-trip instances proposed by Grangier et al. (2016). Three main characteristics allow us to estimate the difficulty of an instance: the size (number of customers, satellites and depots), the capacity of city freighters, and the size of time windows relative to the time horizon. Indeed, a large second-level vehicle capacity results in large extended graphs $\mathcal{G}^s$, $\mathcal{G}'$, and $\mathcal{G}''$, used in the pricing problems. Also, wide time windows lead to a larger number of labels in the labelling algorithm.

Instances by Dellaert et al. (2019) are divided in four classes Ca, Cb, Cc, and Cd. They have narrow time windows: the widest time window is from 7% to 20% of the time horizon, depending on the instance class. Instances in classes Ca, Cb, and Cd have customers with demands 10 or 20 with city freighter capacity equal to 50. Thus, capacity can be divided by 10. Instances in class Cc has customers with integer demands from 5 to 25, and city freighter capacity equal to 50. We put all instances by Dellaert et al. (2019) in set D.

Instances by Grangier et al. (2016) were adapted from famous Solomon instances for the VRPTW. We split these instances into three sets depending on their difficulty. Set G contains 9 difficult instances in class `c1`, `c102`, ..., `c109`, which have small city freighter capacity (the original capacity can also be divided by 10) and tight time windows. Set H contains very difficult instances in classes `c2`, `r1`, and `rc1`, which have either wide time windows or large city freighter capacity. Set I contains "intractable" instances in classes `r2` and `rc2` with wide time windows and large city freighter capacity.

Table 1: Sets of instances from the literature used for experiments

Set	#	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{C}\rvert$	Difficulty	Authors
D	237	2,3,6	3,4,5	15,30,50,100	easy-difficult	Dellaert et al. (2019)
G	9	1	8	100	difficult	Grangier et al. (2016)
H	28	1	8	100	very difficult	Grangier et al. (2016)
I	19	1	8	100	intractable	Grangier et al. (2016)

Table 1 gives an overview of these instances. It contains, for each set, the number of instances, the number of distribution centers, the number of satellites, the number of customers, difficulty estimation, and the authors. The number of feasible first-level routes starting from a distribution center is bounded from above by $\sum_{k=1}^{|\mathcal{S}|} \binom{|\mathcal{S}|}{k} k!$, where $|\mathcal{S}|$ is the number of satellites. Thus instances in set D have at most 1950 feasible first-level routes. Instances in other sets G, H, and I may have up to $109,600$ feasible first-level routes. It is important to notice that in all literature instances, the capacity of an urban truck is a multiple of the capacity of a city freighter.

5.2 Results for literature instances

We first experiment our algorithm on literature instances. We use two variants of our BCP algorithm :

- BCP_{base} — the variant without separation of valid inequalities SSI and VCI (thus, smaller graph $\mathcal{G}'$ is used when solving pricing problems for multi-trip instances)
- $\text{BCP}_{\text{complete}}$ — the variant with separation of all families of valid inequalities

5.2.1 Results for single-trip instances

We run our BCP algorithm on instances of set D with the time limit of 10 hours. On each server, we optimize in parallel 12 instances that share 128 Go of RAM. Table 2 compares two variants of our algorithm with the one proposed by Dellaert et al. (2019). For a fair comparison, the solution time of the latter is multiplied by 1.2 because of the difference in speed between the computers used. In the table, we give average values for instances with the same number of distribution centers, satellites and customers: the average root gap (RG), the geometric mean of the number of branch-and-bound nodes (Nds), the geometric mean of total solution time in seconds (ST), and the number of instances solved to optimality within 3 hours. For unsolved instances, the solution time is set to 3 hours.

We first discuss the comparison between two variants of our BCP algorithm. It is clear that separation of SSIs and VCIs makes the algorithm more efficient. Indeed, these cuts improve dramatically the root gap and significantly decrease the number of nodes in the branch-and-bound tree. Four more instances could be solved to optimality in 3 hours, and the average solution time is several times smaller. The complete variant of our BCP algorithm solves all but four instances within three hours. Two additional instances are solved in 10 hours, and two instances remain open. Detailed results of $\text{BCP}_{\text{complete}}$ for single-trip literature instances are given in Appendix A.

Table 2: Comparison with the algorithm by Dellaert et al. (2019)

Instance			BCP_{base}				$BCP_{complete}$				Literature	
$\|\mathcal{D}\|$	$\|\mathcal{S}\|$	$\|\mathcal{C}\|$	RG(%)	Nds	ST(s)	Solved	RG(%)	Nds	ST(s)	Solved	ST(s)	Solved
2	3	15	0.49	1.1	1	20/20	0.00	1.0	1	20/20	0	20/20
2	3	30	2.31	3.9	6	20/20	0.12	1.7	3	20/20	14	20/20
2	3	50	0.87	4.1	17	20/20	0.27	2.5	14	20/20	715	14/20
2	3	100	0.46	51.0	532	16/20	0.33	13.1	195	18/20	7780	6/20
3	5	15	3.10	1.7	6	20/20	0.05	1.1	4	20/20	2	20/20
3	5	30	3.93	6.5	35	20/20	0.23	1.7	13	20/20	50	20/20
3	5	50	2.34	22.8	232	20/20	0.37	2.8	51	20/20	862	19/20
3	5	100	0.66	42.4	986	17/20	0.34	12.2	449	19/20	10152	3/20
6	4	15	1.19	1.1	3	17/17	0.00	1.0	3	17/17	0	17/17
6	4	30	3.15	4.4	21	20/20	0.17	1.6	9	20/20	17	20/20
6	4	50	0.89	5.3	49	20/20	0.30	2.9	33	20/20	586	18/20
6	4	100	0.39	12.4	283	19/20	0.27	7.2	215	19/20	10715	2/20

Both variants of our BCP algorithm outperform significantly the algorithm by Dellaert et al. (2019). Even though we solve a relaxation of the problem solved by Dellaert et al. (2019), our post-processing procedure shows that all our optimal solutions can be transformed to satisfy the exact synchronization and avoid freight consolidation without increasing the transportation cost. Thus, all our optimal solutions are also optimal for the variant of the problem considered by Dellaert et al. (2019). We solve 54 open instances to optimality for the first time. The best solutions we found for two instances not solved to optimality require freight consolidation. Thus, these solutions are not feasible for the variant considered by Dellaert et al. (2019).

5.2.2 Multi-trip variant

We run our BCP algorithm on multi-trip instances with the time limit of 10 hours. On each server, we optimize 2 instances of set G that share 128 Go using $BCP_{complete}$, and only one instance of set H using BCP_{base}. Variant $BCP_{complete}$ is not suitable for instances in set H, because graph $\mathcal{G}''$ in the pricing problem becomes very large, and the pricing problem becomes intractable. The same happens for instances in set I: even graph $\mathcal{G}'$ used in variant BCP_{base} becomes too large. We fix the sizes of the fleets of urban trucks and city freighters to sizes in the best solutions found by Grangier et al. (2016).

Table 3 gives an overview of our results. For each set, it contains the number of instances solved to optimality, the number of instances for which the algorithm finds a feasible solution without proving optimality, the number of instances for which the algorithm does not find any solution, and the number of instances on which the algorithm fails, i.e. the column generation does not finish when solving formulation (LF2) and a lower bound cannot be obtained.

Table 3: Overview of results for multi-trip instances

Set	Algorithm	Optimal	Feasible	No solution	Failure	Total
G	$BCP_{complete}$	5	2	2	0	9
H	BCP_{base}	3	5	17	3	28
I	BCP_{base}	0	0	0	19	19

Since our algorithm finds optimal solutions to half of the instances of set G, it can handle multi-trip instances with small city freighter capacity and tight time windows. Other instances are much more difficult for our algorithm. Indeed, BCP_{base} finds only 3 optimal solutions for instances in set H and fails to optimize the root node for 3 of them. For the instances in set I, the model does not fit in the server memory.

Table 4 lists the multi-trip instances solved to optimality by our BCP algorithm. In this

Table 4: Overview of experiments on multi-trip instances

Instance	Set	Δ	Consolidation	BCP val	Grangier et al. (2016)	Gap(%)
c101	G	1478	false	1969.3	2022.4	2.70
c105	G	662	false	1873.3	1934.0	3.24
c106	G	997	false	1903.0	1945.0	2.21
c107	G	557	false	1846.4	1888.9	2.30
c108	G	756	false	1825.9	1875.3	2.71
c201	H	5763	false	1277.5	1389.3	8.75
r101	H	0	false	**2298.7**	2333.5	1.51
r102	H	0	false	**2109.3**	2136.8	1.30

table, we give the name of the instance, the set to which belongs the instance, the objective value of the post-processing MIP, the necessity of consolidation, the optimal value found by our algorithm, the best solution value found by Grangier et al. (2016), and the relative gap between these two values. Only for instances `r101` and `r102`, the optimal solutions do not require any storage. Thus, these solutions are also optimal for the variant, considered by Grangier et al. (2016). Detailed results are given in Appendix B.

Table 4 shows that the heuristic from Grangier et al. (2016) seems to be of a good quality. Indeed, the total distance travelled in the optimal solutions after relaxation of exact synchronization is generally 2–3% lower than the one in the heuristic solutions. However, further progress in exact solution of the 2E-VRPTW is needed to be able to estimate the quality of this heuristic on a larger set of instances. We also note that for instance `c201`, the gap between solutions with and without synchronisation is sufficiently large to consider the possibility to have storage at satellites, i.e. to replace satellites by UCCs in practice.

5.3 Results for new single-trip instances

As almost all single-trip literature instances were solved to optimality, here we generate more difficult instances. These instances are based on the multi-trip instances in sets G and H. The fleet size is unlimited, but the cost of using an urban truck is set to 50 and the cost of using a city freighter is set to 25, as in the instances in set D. The city freighters start and finish from a satellite, as in set D instances.

We run our algorithm with the time limit of 10 hours. On one server, two instances in set G on one instance in set H are optimized in parallel.

Table 5: Overview of experiments on new single-trip instances

	Set G		Set H		
	Multi-trip	Single-trip	Multi-trip	Single-trip	
Status	$\text{BCP}_{\text{complete}}$	$\text{BCP}_{\text{complete}}$	BCP_{base}	BCP_{base}	$\text{BCP}_{\text{complete}}$
Optimal	5	9	3	10	15
Feasible	2	0	5	17	10
No solution	2	0	17	1	3
Failure	0	0	3	0	0

In Table 5, we report the overview of results for new single-trip instances. For comparison purposes, results for multi-trip instances are also recalled. This experiment shows that the multi-trip variant is more difficult than the single-trip one. Moreover, new single-trip instances are more difficult that instances in set D, as our algorithm solved to optimality only half of instances in set H. Detailed results are available in Appendix C.

5.4 Results for smaller multi-trip instances

We derive new multi-trip instances from ones in sets G, H, and I, originally based on Solomon instances for the VRPTW. Positions of the distribution center and the satellites follow the procedure described by Grangier et al. (2016) that we recall now. They introduce an X/Y/M/N notation, where X and Y give the position of the distribution center expressed as a percentage of the size map, M and N are the number of rows and columns of a grid cutting the map in rectangles of equal sizes. Satellites are positioned at each intersection in the grid. In our new instances, we keep the distribution center in the same location but we change the number of customers and the number of satellites. We have :

- 25 customers with a 50/150/2/2 configuration (4 satellites)
- 50 customers with a 50/150/2/2 configuration (4 satellites)
- 75 customers with a 50/150/2/3 configuration (6 satellites)

The size of the vehicle fleet is unlimited. We set the cost of using an urban truck to 50 and the cost of using a city freighter to 25.

We run the variant $\text{BCP}_{\text{complete}}$ of our algorithm with the time limit of 10 hours. On each server, we simultaneously optimize at most 24 instances with 25 customers on a server, 12 instances with 50 customers, and 4 instances with 75 customers.

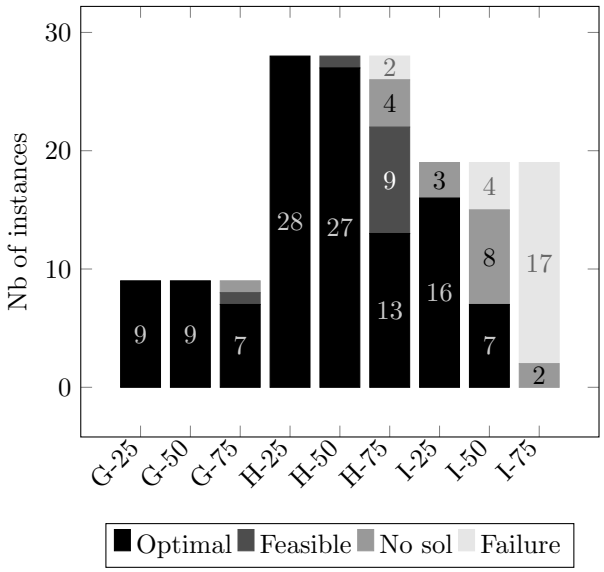


Figure 7: Overview of the results for multi-trip instances with 25, 50, and 75 customers

Figure 7 presents an overview of the results. The columns give the results for different instance classes denoted by the instance set and the number of customers. Our algorithm can solve the absolute majority of multi-trip instances with up to 50 customers. Beyond that size, the efficiency of the algorithm degrades significantly. Unsurprisingly, instances in set I become quickly intractable, even lower bounds for some instances in this set with 50 customers cannot be found. Detailed results are given in Appendices D, E, and F.

5.5 Results for instances with modified vehicle capacity

In all instances considered above, the capacity of an urban truck is a multiple of the city freighter capacity. In this special case, freight consolidation at satellites is not likely to happen. Our

experiments confirm this, as only for two instances freight consolidation is required in the best found solutions.

In this experiment, we verify whether change of vehicle capacity increases freight consolidation. We consider modified instances based on ones in sets D, G-75, and H-75. For instances in set D, the capacity of urban trucks is reduced from 200 to 180, and the capacity of city freighters remains 50. Thus an urban truck has 3.6 times more capacity than a city freighter. For instances in sets G and H, we change the ratios for urban truck capacity / city freighter capacity, defined by Grangier et al. (2016). The ratio become 3.75/0.5 for instances in classes `r1`, `c1`, and `rc1` (i.e. an urban truck has 7.5 times more capacity than a city freighter). The ratio becomes 2/0.35 for instances in classes `r2`, `c2`, and `rc2` (i.e. an urban truck has 5.7 times more capacity than a city freighter). These new instances are run in the same way as the original instances. We report an overview of results in Tables 6 and 7.

Table 6: Overview of results for instances with original vehicle capacity

Variant	Synchronization	No consolidation	Total Optimal
Single-trip	235	235	235
Multi-Trip	2	18	20

Table 7: Overview of results for instances with modified vehicle capacity

Variant	Synchronization	No consolidation	Total Optimal
Single-trip	217	149	217
Multi-Trip	3	16	20

The first result is that about 10% of single trip instances are not solved to optimality. Thus, new instances are more difficult. Moreover, more than 25% of optimal single-trip solutions involve freight consolidation. This consolidation happens mostly for large instances with 100 customers. For multi-trip instances, it is difficult to draw any conclusions. Consolidation is required for 4 optimal solutions instead of 2, but this consolidation increase is small. Detailed results are given in Appendix G.

6 Conclusions

In this paper, we proposed an exact approach for the two-echelon capacitated vehicle routing problem with time windows, in which freight storage and consolidation are allowed at satellites. Our approach can tackle two variants of the problem: (i) when city freighters do a single trip from a single satellite; (ii) and when city freighters can do multiple trips visiting several satellites. Our problem variant is a relaxation of the variant considered in the literature with exact synchronisation of first-level and second-level vehicles. Our solution approach is a branch-cut-and-price algorithm that is based upon recent techniques for classic vehicle routing problems including the two-echelon capacitated vehicle routing problem without time consideration. Our algorithm also uses new problem-specific contributions such as an original MIP formulation with an exponential number of variables and constraints, as well as the separation procedure for constraints linking two levels, and a way to handle these constraints in the pricing problem. We showed that our algorithm is efficient for the single-trip literature instances and some multi-trip literature instances with 100 customers. It outperforms significantly the only existing exact algorithm for the single-trip variant of the problem. It is also the first exact algorithm for the multi-trip variant of the problem.

We experimentally showed that our "precedence" relaxation of the exact synchronization variant is very tight: it is exact for all single-trip literature instances solved to optimality, and it has generally only few percentage relative gap for multi-trip literature instances solved to optimality.

We have generated new single-trip instances which are more difficult that the literature ones. We have also experimentally showed that the vehicle capacity has a large impact on the freight consolidation in the single-trip case.

The first perspective research direction is to improve efficiency of our algorithm. Currently, its applicability is limited by the fact that we use discretisation of vehicle capacity in the pricing problem in order to take into account constraints linking two distribution levels. Thus, instances with large city freighter capacity cannot be solved efficiently or sometimes even cannot be fit into memory. In order to get rid of the discretisation approach, we need to modify the labelling algorithm to solve the pricing problem. This algorithm should be able to work with arcs, for which the resource consumption is variable and the reduced cost depends on this consumption. An approach by Ioachim et al. (1998) can be useful here.

Our current algorithm relies on enumeration of first-level routes. For all literature instances and many practical ones, this enumeration is possible due to a small number of distribution centers and satellites. However, if this number is larger, our approach will fail. In order to extend our algorithm to this case, first-level routes should be generated by solving a pricing problem similarly to the second level. However, a first-level route has coefficient one in the constraints linking two levels if and only if it visits at least one satellite in a certain set. Thus, these constraints resemble strong capacity constraints introduced in (Baldacci et al., 2008), and they modify the structure of the pricing problem for the first level.

One could also think of extending our approach for the variant of the problem with exact synchronization of two distribution levels. This would require a considerable work, as the timing (times of vehicle arrival and departure at nodes of the network) cannot be fixed anymore for a fixed route. One could think of an approach which dynamically generates constraints, which are necessary to ensure such exact synchronization. Another approach is to improve efficiency of the algorithm by Dellaert et al. (2019). However, the variant with exact synchronization seems to be significantly more difficult to treat. Thus, our approach to focus on exactly solving a relaxation can be seen as good trade-off between the quality of the valid bound obtained and the computational effort.

In this work, our effort is mainly to obtain tight valid lower bounds for the problem, and not to obtain feasible solutions. Thus, another research direction is to focus on the latter. The most efficient heuristics for the two-echelon capacitated vehicle routing problem are matheuristics (Wang et al., 2017; Amarouche et al., 2018). Therefore, it seems to be promising to develop matheuristics for our problem, which are based on column generation or on branch-cut-and-price, both for the exact synchronization variant and for the variant with storage and freight consolidation. Our results showed that although the current state-of-art-heuristics like the one by Grangier et al. (2016) are of a good quality, there is still a room for improvement.

Acknowledgments

We are grateful to François Vanderbeck whose efforts contributed to obtaining the Ph.D. studentship for the first author.

We thank Idex Bordeaux and Eduardo Uchoa for financing and implementing a 3 months stay at Federal Fluminense University (Brazil) for the first author.

We thank Nico Dellaert and Phillipe Grangier for kindly sending us some of their solutions.

Experiments presented in this paper were carried out using the PlaFRIM (Federative Platform for Research in Computer Science and Mathematics), created under the Inria PlaFRIM development action with support from Bordeaux INP, LABRI and IMB and other entities: Conseil Régional d'Aquitaine, Université de Bordeaux, CNRS and ANR in accordance to the "Programme d'Investissements d'Avenir".

References

Julian Allen, Michael Browne, Allan Woodburn, and Jacques Leonardi. The Role of Urban Consolidation Centres in Sustainable Freight Transport. *Transport Reviews*, 32(4):473–490, July 2012. doi: 10.1080/01441647.2012.688074.

Youcef Amarouche, Rym N. Guibadj, and Aziz Moukrim. A Neighborhood Search and Set Cover Hybrid Heuristic for the Two-Echelon Vehicle Routing Problem. In Ralf Borndörfer and Sabine Storandt, editors, *18th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2018)*, volume 65 of *OpenAccess Series in Informatics (OASIcs)*, pages 11:1–11:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Roberto Baldacci, Nicos Christofides, and Aristide Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115:351–385, 2008.

Roberto Baldacci, Aristide Mingozzi, Roberto Roberti, and Roberto Wolfler Calvo. An exact algorithm for the two-echelon capacitated vehicle routing problem. *Operations Research*, 61 (2):298–314, 2013.

U. Breunig, V. Schmid, R.F. Hartl, and T. Vidal. A large neighbourhood based heuristic for two-echelon routing problems. *Computers and Operations Research*, 76:208 – 225, 2016.

Claudio Contardo and Rafael Martinelli. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12:129 – 146, 2014.

Claudio Contardo, Vera Hemmelmayr, and Teodor Gabriel Crainic. Lower and upper bounds for the two-echelon capacitated location-routing problem. *Computers & Operations Research*, 39(12):3185 – 3199, 2012.

Teodor Gabriel Crainic, Nicoletta Ricciardi, and Giovanni Storchi. Models for evaluating and planning city logistics systems. *Transportation science*, 43(4):432–454, 2009.

R. Cuda, G. Guastaroba, and M.G. Speranza. A survey on two-echelon routing problems. *Computers & Operations Research*, 55:185 – 199, 2015.

Nico Dellaert, Fardin Dashty Saridarq, Tom Van Woensel, and Teodor Gabriel Crainic. Branch and price based algorithms for the two-echelon vehicle routing problem with time windows. *Transportation Science*, 53(2):463–479, 2019.

Jesus Gonzalez-Feliu, Guido Perboli, Roberto Tadei, and Daniele Vigo. The two-echelon capacitated vehicle routing problem. Technical Report DEIS OR.INGCE 2007/2, Department of Electronics, Computer Science, and Systems, University of Bologna,, 2007.

Philippe Grangier, Michel Gendreau, Fabien Lehuédé, and Louis-Martin Rousseau. An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. *European Journal of Operational Research*, 254(1):80–91, 2016.

Pengfei He and Jing Li. The two-echelon multi-trip vehicle routing problem with dynamic satellites for crop harvesting and transportation. *Applied Soft Computing*, 77:387–398, April 2019.

Vera C. Hemmelmayr, Jean-François Cordeau, and Teodor Gabriel Crainic. An adaptive large neighborhood search heuristic for Two-Echelon Vehicle Routing Problems arising in city logistics. *Computers & Operations Research*, 39(12):3215–3228, December 2012. ISSN 0305-0548. doi: 10.1016/j.cor.2012.04.007. URL `http://www.sciencedirect.com/science/article/pii/S0305054812000871`.

José Holguín-Veras, Johanna Amaya Leal, Ivan Sanchez-Diaz, Michael Browne, and Jeffrey Wojtowicz. State of the art and practice of urban freight management Part II: Financial approaches, logistics, and demand management. *Transportation Research Part A: Policy and Practice*, November 2018. ISSN 0965-8564. doi: 10.1016/j.tra.2018.10.036. URL `http://www.sciencedirect.com/science/article/pii/S0965856418301277`.

Irina Ioachim, Sylvie Gélinas, François Soumis, and Jacques Desrosiers. A dynamic programming algorithm for the shortest path problem with time windows and linear node costs. *Networks*, 31(3):193–204, 1998.

Mads Jepsen, Bjorn Petersen, Simon Spoorendonk, and David Pisinger. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56(2):497–511, 2008.

Mads Jepsen, Simon Spoorendonk, and Stefan Ropke. A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. *Transportation Science*, 47(1): 23–37, 2013.

G. Laporte and Y. Nobert. A branch and bound algorithm for the capacitated vehicle routing problem. *Operations-Research-Spektrum*, 5(2):77–85, Jun 1983.

Hongqi Li, Lu Zhang, Tan Lv, and Xinyu Chang. The two-echelon time-constrained vehicle routing problem in linehaul-delivery systems. *Transportation Research Part B: Methodological*, 94:169–188, December 2016. ISSN 0191-2615. doi: 10.1016/j.trb.2016.09.012. URL `http://www.sciencedirect.com/science/article/pii/S0191261515301211`.

Hongqi Li, Yinying Liu, Kaihang Chen, and Qingfeng Lin. The two-echelon city logistics system with on-street satellites. *Computers & Industrial Engineering*, 139:105577, January 2020a.

Hongqi Li, Haotian Wang, Jun Chen, and Ming Bai. Two-echelon vehicle routing problem with time windows and mobile satellites. *Transportation Research Part B: Methodol*, 138:179–201, 2020b.

Jens Lysgaard. CVRPSEP: a package of separation routines for the capacitated vehicle routing problem, 2018. URL `http://econ.au.dk/research/researcher-websites/jens-lysgaard/cvrpsep/`.

Jens Lysgaard, Adam N. Letchford, and Richard W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2):423–445, Jun 2004.

Guillaume Marques, Ruslan Sadykov, Jean-Christophe Deschamps, and Rémy Dupas. An improved branch-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. *Computers & Operations Research*, 114:104833, 2020.

D. R. McDermott. An Alternative Framework for Urban Goods Distribution: Consolidation. *Transportation Journal*, 15(1):29–39, 1975. ISSN 0041-1612.

Ibrahim Muter, Jean-François Cordeau, and Gilbert Laporte. A branch-and-price algorithm for the multidepot vehicle routing problem with interdepot routes. *Transportation Science*, 48(3): 425–441, 2014.

Pamela C. Nolz, Nabil Absi, Diego Cattaruzza, and Dominique Feillet. Two-echelon distribution with a single capacitated city hub. *EURO Journal on Transportation and Logistics*, page 100015, 2020.

Diego Pecin, Artur Pessoa, Marcus Poggi, and Eduardo Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9(1):61–100, 2017a.

Diego Pecin, Artur Pessoa, Marcus Poggi, Eduardo Uchoa, and Haroldo Santos. Limited memory rank-1 cuts for vehicle routing problems. *Operations Research Letters*, 45(3):206 – 209, 2017b.

Guido Perboli, Roberto Tadei, and Daniele Vigo. The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. *Transportation Science*, 45(3):364–380, 2011.

Artur Pessoa, Eduardo Uchoa, and Marcus Poggi de Aragão. A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem. *Networks*, 54(4):167–177, 2009.

Artur Pessoa, Ruslan Sadykov, Eduardo Uchoa, and François Vanderbeck. Automation and combination of linear-programming based stabilization techniques in column generation. *INFORMS Journal on Computing*, 30(2):339–360, 2018.

Artur Pessoa, Ruslan Sadykov, Eduardo Uchoa, and François Vanderbeck. A generic exact solver for vehicle routing and related problems. *Mathematical Programming B*, accepted, 2020.

Edward Rothberg. An evolutionary algorithm for polishing mixed integer programming solutions. *INFORMS Journal on Computing*, 19(4):534–541, 2007.

Ruslan Sadykov, Eduardo Uchoa, and Artur Pessoa. A bucket graph based labeling algorithm with application to vehicle routing. *Transportation Science*, accepted, 2020.

Fernando Afonso Santos, Geraldo Robson Mateus, and Alexandre Salles da Cunha. A branch-and-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. *Transportation Science*, 49(2):355–368, 2015.

François Vanderbeck, Ruslan Sadykov, and Issam Tahiri. BaPCod — a generic Branch-And-Price Code, 2019. URL `https://realopt.bordeaux.inria.fr/?page_id=2`.

Kangzhou Wang, Yeming Shao, and Weihua Zhou. Matheuristic for a two-echelon capacitated vehicle routing problem with environmental considerations in city logistics service. *Transportation Research Part D: Transport and Environment*, 57:262 – 276, 2017.

Ziqi Wang and Peihan Wen. Optimization of a low-carbon two-echelon heterogeneous-fleet vehicle routing for cold chain logistics under mixed time window. *Sustainability*, 12(5):179–201, 2020.

Zheng-Yang Zeng, Wei-Sheng Xu, Zhi-Yu Xu, and Wei-Hui Shao. A hybrid GRASP+VND heuristic for the two-echelon vehicle routing problem arising in city logistics,. *Mathematical Problems in Engineering*, pages 1–11, 2014.

A Detailed BCP results for literature single-trip instances

In following tables, Rg stands for the root gap, Rt stands for the time spent in the root node, Nodes stands for the number of nodes explored in the branching tree, Fg stands for the final gap, BPB is the best primal bound found, t stands for the total time spent, Δ is the value of the post-processing objective function, and Conso indicates whether there is consolidation in the final solution. Note that if the time in the column "Literature" has prefix ">" for a given instance, authors did not solve this instance to optimality.

Table 8: Results of experiments on instances of set D

	BCP_{best} with primal heuristic						PostProc		Literature
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso	t (s)
Ca1-2-3-15	0.0	0	1	0.0	612.385	0	0.0	false	0
Ca2-2-3-15	0.0	1	1	0.0	548.953	1	0.0	false	0
Ca3-2-3-15	0.0	0	1	0.0	551.985	0	0.0	false	0
Ca4-2-3-15	0.0	1	1	0.0	569.579	1	0.0	false	0
Ca5-2-3-15	0.0	1	1	0.0	555.796	1	0.0	false	0
Cb1-2-3-15	0.0	1	1	0.0	624.178	1	0.0	false	0
Cb2-2-3-15	0.0	1	1	0.0	516.739	1	0.0	false	0
Cb3-2-3-15	0.0	1	1	0.0	601.897	1	0.0	false	0
Cb4-2-3-15	0.0	1	1	0.0	546.314	1	0.0	false	0

	BCP_{best} with primal heuristic						PostProc		Literature
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso	t (s)
Cb5-2-3-15	0.0	1	1	0.0	494.395	1	0.0	false	0
Cc1-2-3-15	0.0	1	1	0.0	586.856	1	0.0	false	1
Cc2-2-3-15	0.0	1	1	0.0	482.985	1	0.0	false	0
Cc3-2-3-15	0.0	1	1	0.0	539.685	1	0.0	false	0
Cc4-2-3-15	0.0	1	1	0.0	562.798	1	0.0	false	0
Cc5-2-3-15	0.0	1	1	0.0	436.467	1	0.0	false	2
Cd1-2-3-15	0.0	1	1	0.0	597.698	1	0.0	false	0
Cd2-2-3-15	0.0	1	1	0.0	483.133	1	0.0	false	0
Cd3-2-3-15	0.0	1	1	0.0	512.412	1	0.0	false	0
Cd4-2-3-15	0.0	1	1	0.0	585.301	1	0.0	false	0
Cd5-2-3-15	0.0	1	1	0.0	536.764	1	0.0	false	0
Ca1-3-5-15	0.0	2	1	0.0	603.456	2	0.0	false	3
Ca2-3-5-15	0.0	2	1	0.0	628.405	2	0.0	false	6
Ca3-3-5-15	0.0	2	1	0.0	527.679	2	0.0	false	1
Ca4-3-5-15	0.0	2	1	0.0	514.399	2	0.0	false	0
Ca5-3-5-15	0.0	3	1	0.0	521.399	3	0.0	false	4
Cb1-3-5-15	0.0	6	1	0.0	532.34	6	0.0	false	0
Cb2-3-5-15	0.0	6	1	0.0	612.606	6	0.0	false	5
Cb3-3-5-15	0.0	6	1	0.0	537.256	6	0.0	false	1
Cb4-3-5-15	0.0	9	1	0.0	438.674	9	0.0	false	0
Cb5-3-5-15	0.0	7	1	0.0	538.57	7	0.0	false	3
Cc1-3-5-15	0.8	9	3	0.0	473.553	12	0.0	false	1
Cc2-3-5-15	0.0	5	1	0.0	494.727	5	0.0	false	1
Cc3-3-5-15	0.0	3	1	0.0	522.837	3	0.0	false	3
Cc4-3-5-15	0.0	5	1	0.0	506.35	5	0.0	false	3
Cc5-3-5-15	0.0	3	1	0.0	545.077	3	0.0	false	4
Cd1-3-5-15	0.0	4	1	0.0	568.629	4	0.0	false	4
Cd2-3-5-15	0.0	4	1	0.0	506.612	4	0.0	false	0
Cd3-3-5-15	0.0	2	1	0.0	522.8	2	0.0	false	3
Cd4-3-5-15	0.3	4	3	0.0	506.825	6	0.0	false	3
Cd5-3-5-15	0.0	3	1	0.0	544.084	3	0.0	false	3
Ca1-6-4-15	0.0	2	1	0.0	551.457	2	0.0	false	0
Ca2-6-4-15	0.0	3	1	0.0	560.919	3	0.0	false	0
Ca3-6-4-15	0.0	3	1	0.0	556.642	3	0.0	false	1
Ca4-6-4-15	0.0	5	1	0.0	465.226	5	0.0	false	0
Ca5-6-4-15	0.0	2	1	0.0	416.632	2	0.0	false	0
Cb1-6-4-15	0.0	4	1	0.0	567.151	4	0.0	false	0
Cb2-6-4-15	0.0	7	1	0.0	631.512	7	0.0	false	0
Cb3-6-4-15	0.0	4	1	0.0	561.536	4	0.0	false	0
Cb4-6-4-15	0.0	4	1	0.0	510.954	4	0.0	false	0
Cb5-6-4-15	0.0	5	1	0.0	460.569	5	0.0	false	0
Cc1-6-4-15	0.0	2	1	0.0	566.013	2	0.0	false	0
Cc2-6-4-15	0.0	3	1	0.0	549.229	3	0.0	false	1
Cc3-6-4-15	0.0	3	1	0.0	540.875	3	0.0	false	0
Cc4-6-4-15	0.0	3	1	0.0	521.621	3	0.0	false	1
Cc5-6-4-15	0.0	2	1	0.0	425.83	2	0.0	false	0
Cd1-6-4-15	0.0	2	1	0.0	551.492	2	0.0	false	0
Cd2-6-4-15	0.0	2	1	0.0	554.83	2	0.0	false	0
Ca1-2-3-30	0.0	1	1	0.0	981.374	1	0.0	false	424
Ca2-2-3-30	0.0	1	1	0.0	968.324	1	0.0	false	3
Ca3-2-3-30	0.0	1	1	0.0	1069.81	1	0.0	false	1
Ca4-2-3-30	0.7	3	3	0.0	948.047	5	0.0	false	5
Ca5-2-3-30	0.2	3	3	0.0	968.231	6	0.0	false	11
Cb1-2-3-30	0.0	6	3	0.0	985.763	8	0.0	false	7
Cb2-2-3-30	0.0	3	1	0.0	937.119	3	0.0	false	11
Cb3-2-3-30	0.0	3	1	0.0	1076.254	3	0.0	false	1
Cb4-2-3-30	0.1	4	3	0.0	981.344	6	0.0	false	2
Cb5-2-3-30	0.1	6	3	0.0	834.854	8	0.0	false	9
Cc1-2-3-30	0.0	3	1	0.0	799.562	3	0.0	false	19
Cc2-2-3-30	0.0	2	1	0.0	937.712	2	0.0	false	13
Cc3-2-3-30	0.4	3	3	0.0	1013.546	7	0.0	false	675
Cc4-2-3-30	0.0	2	1	0.0	943.279	2	0.0	false	157
Cc5-2-3-30	0.3	4	3	0.0	949.387	7	0.0	false	31
Cd1-2-3-30	0.0	1	1	0.0	807.247	1	0.0	false	4
Cd2-2-3-30	0.0	1	1	0.0	945.304	1	0.0	false	5
Cd3-2-3-30	0.3	2	3	0.0	1030.606	4	0.0	false	12
Cd4-2-3-30	0.0	2	3	0.0	958.674	4	0.0	false	24
Cd5-2-3-30	0.4	4	3	0.0	965.894	6	0.0	false	67

	BCP_{best} with primal heuristic						PostProc		Literature
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso	t (s)
Ca1-3-5-30	0.1	9	3	0.0	892.746	13	0.0	false	22
Ca2-3-5-30	0.0	6	1	0.0	960.533	7	0.0	false	30
Ca3-3-5-30	0.2	5	3	0.0	924.509	8	0.0	false	21
Ca4-3-5-30	0.0	4	1	0.0	909.869	6	0.0	false	22
Ca5-3-5-30	0.0	6	1	0.0	892.66	6	0.0	false	18
Cb1-3-5-30	0.5	22	3	0.0	928.046	30	0.0	false	58
Cb2-3-5-30	0.7	26	3	0.0	958.302	38	0.0	false	88
Cb3-3-5-30	0.0	16	1	0.0	900.905	16	0.0	false	10
Cb4-3-5-30	0.3	22	3	0.0	903.874	33	0.0	false	29
Cb5-3-5-30	0.6	38	3	0.0	977.906	75	0.0	false	55
Cc1-3-5-30	1.0	9	3	0.0	943.74	18	0.0	false	75
Cc2-3-5-30	0.0	12	1	0.0	897.767	12	0.0	false	78
Cc3-3-5-30	0.0	7	1	0.0	943.295	7	0.0	false	35
Cc4-3-5-30	0.4	14	3	0.0	890.652	19	0.0	false	391
Cc5-3-5-30	0.0	17	1	0.0	772.331	17	0.0	false	368
Cd1-3-5-30	0.5	7	3	0.0	955.083	11	0.0	false	44
Cd2-3-5-30	0.0	8	1	0.0	944.603	8	0.0	false	46
Cd3-3-5-30	0.0	5	1	0.0	907.529	5	0.0	false	35
Cd4-3-5-30	0.0	8	1	0.0	917.202	8	0.0	false	114
Cd5-3-5-30	0.2	11	3	0.0	917.008	17	0.0	false	101
Ca1-6-4-30	0.0	5	3	0.0	922.215	10	0.0	false	1104
Ca2-6-4-30	1.6	4	3	0.0	**956.349**	9	0.0	false	3
Ca3-6-4-30	0.0	5	1	0.0	869.334	5	0.0	false	6
Ca4-6-4-30	0.0	7	3	0.0	867.416	12	0.0	false	14
Ca5-6-4-30	0.0	3	1	0.0	851.516	3	0.0	false	2
Cb1-6-4-30	0.3	13	3	0.0	968.054	20	0.0	false	6
Cb2-6-4-30	0.0	11	1	0.0	920.495	11	0.0	false	4
Cb3-6-4-30	0.0	10	1	0.0	929.483	10	0.0	false	4
Cb4-6-4-30	0.0	16	1	0.0	877.54	16	0.0	false	15
Cb5-6-4-30	0.0	10	1	0.0	895.742	10	0.0	false	11
Cc1-6-4-30	0.0	7	1	0.0	861.127	7	0.0	false	48
Cc2-6-4-30	0.0	5	1	0.0	912.379	5	0.0	false	12
Cc3-6-4-30	0.9	10	3	0.0	899.833	20	0.0	false	24
Cc4-6-4-30	0.0	7	1	0.0	833.637	7	0.0	false	228
Cc5-6-4-30	0.0	5	1	0.0	813.91	5	0.0	false	572
Cd1-6-4-30	0.1	7	3	0.0	918.387	11	0.0	false	26
Cd2-6-4-30	0.0	4	1	0.0	914.267	4	0.0	false	4
Cd3-6-4-30	0.2	7	3	0.0	880.593	11	0.0	false	10
Cd4-6-4-30	0.2	7	3	0.0	885.413	13	0.0	false	15
Cd5-6-4-30	0.1	6	3	0.0	943.678	10	0.0	false	19
Ca1-2-3-50	0.1	2	3	0.0	1501.802	3	0.0	false	3
Ca2-2-3-50	0.1	7	3	0.0	1586.208	11	0.0	false	>12973
Ca3-2-3-50	0.0	5	3	0.0	1479.559	8	0.0	false	48
Ca4-2-3-50	0.4	6	3	0.0	1485.699	11	0.0	false	4117
Ca5-2-3-50	0.2	8	3	0.0	1397.052	13	0.0	false	235
Cb1-2-3-50	0.0	10	1	0.0	1455.112	10	0.0	false	33
Cb2-2-3-50	0.4	15	3	0.0	1468.47	25	0.0	false	1010
Cb3-2-3-50	0.8	12	3	0.0	1492.247	30	0.0	false	4042
Cb4-2-3-50	0.4	13	3	0.0	1469.009	20	0.0	false	680
Cb5-2-3-50	0.6	11	3	0.0	1415.231	20	0.0	false	1379
Cc1-2-3-50	0.0	7	1	0.0	**1407.683**	7	0.0	false	>12973
Cc2-2-3-50	0.0	6	3	0.0	1468.881	13	0.0	false	>12973
Cc3-2-3-50	0.4	8	3	0.0	**1376.549**	28	0.0	false	>12973
Cc4-2-3-50	0.5	7	3	0.0	**1425.013**	29	0.0	false	>12973
Cc5-2-3-50	0.2	6	3	0.0	1442.652	15	0.0	false	393
Cd1-2-3-50	0.0	6	1	0.0	1458.036	6	0.0	false	37
Cd2-2-3-50	0.1	7	3	0.0	**1568.682**	12	0.0	false	>12973
Cd3-2-3-50	0.4	8	3	0.0	1449.698	13	0.0	false	160
Cd4-2-3-50	0.4	12	3	0.0	1453.238	18	0.0	false	154
Cd5-2-3-50	0.4	7	3	0.0	1462.295	20	0.0	false	37
Ca1-3-5-50	0.0	12	3	0.0	1351.827	16	0.0	false	796
Ca2-3-5-50	0.4	14	3	0.0	1354.119	25	0.0	false	207
Ca3-3-5-50	0.2	23	3	0.0	1271.666	46	0.0	false	587
Ca4-3-5-50	0.4	20	3	0.0	1299.863	34	0.0	false	446
Ca5-3-5-50	0.0	19	1	0.0	1267.494	19	0.0	false	148
Cb1-3-5-50	1.1	67	11	0.0	1355.495	304	0.0	false	759
Cb2-3-5-50	1.2	68	3	0.0	1391.312	149	0.0	false	955
Cb3-3-5-50	0.6	71	3	0.0	1368.969	150	0.0	false	1018

	BCP_{best} with primal heuristic						PostProc		Literature
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso	t (s)
Cb4-3-5-50	0.7	65	3	0.0	1298.335	142	0.0	false	310
Cb5-3-5-50	0.5	67	3	0.0	1270.393	146	0.0	false	555
Cc1-3-5-50	0.0	15	1	0.0	1293.882	15	0.0	false	1811
Cc2-3-5-50	0.1	23	3	0.0	1317.821	46	0.0	false	180
Cc3-3-5-50	0.1	23	3	0.0	1288.899	45	0.0	false	298
Cc4-3-5-50	0.8	38	7	0.0	1348.835	129	0.0	false	>12973
Cc5-3-5-50	0.2	26	3	0.0	1276.467	52	0.0	false	1083
Cd1-3-5-50	0.2	23	3	0.0	1314.586	44	0.0	false	6002
Cd2-3-5-50	0.3	21	3	0.0	1326.045	43	0.0	false	567
Cd3-3-5-50	0.2	24	3	0.0	1351.504	47	0.0	false	10243
Cd4-3-5-50	0.0	15	1	0.0	1315.544	15	0.0	false	535
Cd5-3-5-50	0.4	16	3	0.0	1289.705	36	0.0	false	2485
Ca1-6-4-50	0.5	10	3	0.0	1335.553	21	0.0	false	>12973
Ca2-6-4-50	0.6	15	5	0.0	1376.13	52	0.0	false	6906
Ca3-6-4-50	0.4	12	3	0.0	1337.15	24	0.0	false	1578
Ca4-6-4-50	0.0	9	1	0.0	1242.562	9	0.0	false	36
Ca5-6-4-50	0.0	15	1	0.0	1268.843	15	0.0	false	97
Cb1-6-4-50	0.4	26	3	0.0	1355.919	39	0.0	false	77
Cb2-6-4-50	1.4	39	43	0.0	1456.853	698	0.0	false	3819
Cb3-6-4-50	0.3	31	3	0.0	1345.022	51	0.0	false	90
Cb4-6-4-50	0.1	28	3	0.0	1281.665	39	0.0	false	44
Cb5-6-4-50	0.6	35	3	0.0	1341.249	75	0.0	false	294
Cc1-6-4-50	0.0	14	1	0.0	1262.599	14	0.0	false	1458
Cc2-6-4-50	0.7	17	7	0.0	1304.677	99	0.0	false	8674
Cc3-6-4-50	0.1	13	3	0.0	1315.851	29	0.0	false	7327
Cc4-6-4-50	0.1	12	3	0.0	1206.641	25	0.0	false	>12973
Cc5-6-4-50	0.0	13	1	0.0	1453.393	13	0.0	false	509
Cd1-6-4-50	0.1	18	3	0.0	1300.531	31	0.0	false	71
Cd2-6-4-50	0.2	15	3	0.0	1338.281	33	0.0	false	88
Cd3-6-4-50	0.1	11	3	0.0	1375.061	22	0.0	false	60
Cd4-6-4-50	0.1	16	3	0.0	1236.79	27	0.0	false	410
Cd5-6-4-50	0.3	17	3	0.0	1332.863	29	0.0	false	1477
Ca1-2-3-100	0.4	15	3	0.0	**2777.868**	41	0.0	false	>12973
Ca2-2-3-100	0.3	25	3	0.0	**2863.084**	81	0.0	false	>12973
Ca3-2-3-100	0.1	18	3	0.0	**2677.441**	42	0.0	false	>12973
Ca4-2-3-100	0.4	12	5	0.0	3037.909	107	0.0	false	10388
Ca5-2-3-100	0.2	15	3	0.0	2757.454	33	0.0	false	1787
Cb1-2-3-100	0.5	51	135	0.0	**2490.589**	2500	0.0	false	>12973
Cb2-2-3-100	0.7	40	171	0.0	**2737.557**	2068	0.0	false	>12973
Cb3-2-3-100	0.5	39	15	0.0	**2716.633**	359	0.0	false	>12973
Cb4-2-3-100	1.0	43	3223	0.6	**2859.862**	36000	0.0	true	>12973
Cb5-2-3-100	0.6	42	191	0.0	**2798.853**	2544	0.0	false	>12973
Cc1-2-3-100	0.0	21	3	0.0	**2770.561**	37	0.0	false	>12973
Cc2-2-3-100	0.1	22	3	0.0	**2661.935**	54	0.0	false	>12973
Cc3-2-3-100	0.0	27	3	0.0	**2665.606**	49	0.0	false	>12973
Cc4-2-3-100	0.3	16	29	0.0	**2893.907**	293	0.0	false	>12973
Cc5-2-3-100	0.2	19	3	0.0	**2751.94**	53	0.0	false	2112
Cd1-2-3-100	0.0	18	3	0.0	2666.543	35	0.0	false	597
Cd2-2-3-100	0.0	17	3	0.0	2852.931	38	0.0	false	4831
Cd3-2-3-100	0.4	14	3	0.0	**2633.607**	51	0.0	false	>12973
Cd4-2-3-100	0.7	19	1427	0.0	**2904.735**	18364	0.0	false	>12973
Cd5-2-3-100	0.1	20	3	0.0	2794.657	40	0.0	false	1525
Ca1-3-5-100	0.3	35	7	0.0	2608.81	182	0.0	false	957
Ca2-3-5-100	0.6	38	11	0.0	**2483.288**	260	0.0	false	>12973
Ca3-3-5-100	0.1	40	3	0.0	**2452.225**	80	0.0	false	>12973
Ca4-3-5-100	0.2	62	27	0.0	**2648.213**	709	0.0	false	>12973
Ca5-3-5-100	0.3	47	3	0.0	**2621.712**	125	0.0	false	>12973
Cb1-3-5-100	0.7	207	159	0.0	**2646.858**	7754	0.0	false	>12973
Cb2-3-5-100	0.5	195	3	0.0	**2478.019**	477	0.0	false	>12973
Cb3-3-5-100	0.5	237	393	0.0	**2547.652**	11356	0.0	false	>12973
Cb4-3-5-100	0.9	231	1015	0.6	**2724.22**	36000	0.0	true	>12973
Cb5-3-5-100	0.5	182	39	0.0	**2760.237**	3093	0.0	false	>12973
Cc1-3-5-100	0.4	73	21	0.0	**2563.823**	659	0.0	false	>12973
Cc2-3-5-100	0.4	73	17	0.0	**2435.402**	422	0.0	false	>12973
Cc3-3-5-100	0.2	63	5	0.0	**2538.477**	179	0.0	false	>12973
Cc4-3-5-100	0.1	68	3	0.0	**2571.435**	201	0.0	false	>12973
Cc5-3-5-100	0.1	54	3	0.0	**2638.087**	111	0.0	false	>12973
Cd1-3-5-100	0.1	42	3	0.0	2985.336	68	0.0	false	5989

	BCP_{best} with primal heuristic						PostProc		Literature
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso	t (s)
Cd2-3-5-100	0.3	58	3	0.0	**2440.952**	118	0.0	false	>12973
Cd3-3-5-100	0.1	43	3	0.0	2523.659	79	0.0	false	2826
Cd4-3-5-100	0.3	44	31	0.0	**2537.095**	778	0.0	false	>12973
Cd5-3-5-100	0.1	75	3	0.0	**2666.298**	128	0.0	false	>12973
Ca1-6-4-100	0.4	33	71	0.0	**2537.918**	1233	0.0	false	>12973
Ca2-6-4-100	0.2	35	3	0.0	2380.645	79	0.0	false	515
Ca3-6-4-100	0.5	28	7	0.0	**2696.39**	143	0.0	false	>12973
Ca4-6-4-100	0.3	32	5	0.0	**2375.643**	106	0.0	false	>12973
Ca5-6-4-100	0.2	32	5	0.0	**2575.222**	112	0.0	false	>12973
Cb1-6-4-100	0.7	113	1283	0.0	**2375.52**	25157	0.0	false	>12973
Cb2-6-4-100	0.5	118	7	0.0	**2566.25**	499	0.0	false	>12973
Cb3-6-4-100	0.4	164	15	0.0	**2637.52**	861	0.0	false	>12973
Cb4-6-4-100	0.3	108	11	0.0	**2743.53**	630	0.0	false	>12973
Cb5-6-4-100	0.3	135	5	0.0	**2569.621**	333	0.0	false	>12973
Cc1-6-4-100	0.3	62	19	0.0	**2524.815**	439	0.0	false	>12973
Cc2-6-4-100	0.1	45	3	0.0	**2552.466**	89	0.0	false	>12973
Cc3-6-4-100	0.1	48	3	0.0	**2385.649**	98	0.0	false	>12973
Cc4-6-4-100	0.3	42	5	0.0	2559.582	206	0.0	false	>12973
Cc5-6-4-100	0.1	48	3	0.0	2461.533	87	0.0	false	>12973
Cd1-6-4-100	0.3	31	3	0.0	**2591.013**	91	0.0	false	>12973
Cd2-6-4-100	0.0	27	3	0.0	2535.57	52	0.0	false	7128
Cd3-6-4-100	0.0	35	3	0.0	**2409.158**	66	0.0	false	>12973
Cd4-6-4-100	0.1	34	3	0.0	**2677.969**	66	0.0	false	>12973
Cd5-6-4-100	0.2	35	3	0.0	**2575.222**	81	0.0	false	>12973

We could not find the same optimal solution as in (Dellaert et al., 2019) for instances Cb2-3-5-15, Cd1-3-5-100, Ca2-6-4-30, and Cc5-2-3-100. We have asked for published solutions, but the authors were not able to send them to us.

B Detailed BCP results for literature multi-trip instances

In following tables, Rg stands for the root gap, Rt stands for the time spent in the root node, Nodes stands for the number of nodes explored in the branching tree, Fg stands for the final gap, BPB is the best primal bound found, t stands for the total time spent, Δ is the value of the post-processing objective function, and Conso indicates whether there is consolidation in the final solution. Note that if the time in the column "Literature" has prefix ">" for a given instance, authors did not solve this instance to optimality.

Table 9: Results of experiments on instances of set G

	BCP_{best} with primal heuristic						PostProc		Literature
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso	Best
c101	2.4	615	93	0.0	1970.3	20581	1478.0	false	2022.4 ——
c102	2.3	2732	25	1.6	1895.5	35977	1517	false	1947.6
c103	∞	4249	13	∞	∞	36036	-	-	1880.7
c104	∞	5310	15	∞	∞	36007	-	-	1811.1
c105	0.8	534	3	0.0	1874.3	713	662	false	1934.0
c106	1.3	1175	3	0.0	1904.1	1973	997	false	1945.0
c107	0.4	1159	5	0.0	1846.9	3098	557	false	1888.9
c108	1.6	1617	19	0.0	1826.4	17205	756	false	1875.3
c109	3.5	3551	13	3.0	1844.0	35972	267	false	1863.1

Table 10: Results of experiments on instances of set H

	BCP_{base} with primal heuristic						PostProc		Literature
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso	Best
c201	8.4	2475	47	0.0	1278.0	15935	5763.0	false	1389.3
c202	-	-	-	-	-	-	-	-	-
c203	-	-	-	-	-	-	-	-	-
c204	-	-	-	-	-	-	-	-	-
c205	∞	7618	7	∞	∞	36017	-	-	1312.1
c206	∞	8717	3	∞	∞	37222	-	-	1312.6
c207	∞	13213	3	∞	∞	37938	-	-	1280.4
c208	∞	9599	5	∞	∞	36001	-	-	1278.3
r101	2.6	405	43	0.0	2300.1	8396	0.0	false	2333.5
r102	2.7	1441	25	0.0	2110.8	15508	0.0	false	2136.8
r103	∞	2076	23	∞	∞	36007	-	-	1942.7
r104	∞	3237	19	∞	∞	36013	-	-	1777.2
r105	3.6	1911	57	2.4	2060.7	36005	140.0	false	2096.8
r106	∞	1279	37	∞	∞	36043	-	-	1992.4
r107	∞	2500	19	∞	∞	36079	-	-	1779.2
r108	∞	3765	15	∞	∞	36023	-	-	1654.3
r109	∞	2745	13	∞	∞	36044	-	-	1925.9
r110	∞	3341	9	∞	∞	36469	-	-	1833.6
r111	∞	1793	33	∞	∞	36014	-	-	1770.8
r112	∞	4562	11	∞	∞	36074	-	-	1746.0
rc101	5.6	3102	21	3.6	2578.8	36315	259.0	true	2577.0
rc102	8.6	3136	17	6.4	2446.0	36007	415.0	false	2407.1
rc103	∞	4830	9	∞	∞	36007	-	-	2476.9
rc104	∞	4189	13	∞	∞	36053	-	-	2125.9
rc105	6.1	2780	11	4.3	2492.8	36014	190.0	false	2542.6
rc106	10.4	2735	11	8.6	2433.3	36030	371.0	false	2494.9
rc107	∞	2850	11	∞	∞	36017	-	-	2271.1
rc108	∞	3884	9	∞	∞	36062	-	-	2202.9

C Detailed BCP results for the single-trip instances based on sets G and H

Table 11: Results of experiments on instances of set G - single trip

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
c101	0.3	233	5	0.0	2077.7	511	0.0	false
c102	0.7	414	7	0.0	2063.5	1279	0.0	false
c103	0.9	608	3	0.0	2047.5	2344	0.0	false
c104	0.9	863	23	0.0	2029.3	19264	0.0	false
c105	0.7	297	11	0.0	2073.8	1131	0.0	false
c106	0.7	328	5	0.0	2062.5	920	0.0	false
c107	0.3	366	3	0.0	2065.9	743	0.0	false
c108	0.7	438	7	0.0	2055.3	1393	0.0	false
c109	1.0	381	7	0.0	2042.1	2028	0.0	false

Table 12: Results of experiments on instances of set H - single trip

	BCP_{base} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
c201	4.4	574	11	0.0	1565.1	2217	0.0	false
c202	4.9	1366	23	0.0	1558.3	9666	0.0	false
c203	7.9	2047	19	5.5	1566.2	36003	0.0	false
c204	∞	3748	11	∞	∞	36319	-	-
c205	4.6	810	15	0.0	1550.7	4726	0.0	false

	BCP_{base} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
c206	6.9	951	49	5.7	1570.3	36005	0.0	false
c207	5.0	1020	19	0.0	1540.1	9687	0.0	false
c208	5.8	1168	51	0.0	1535.3	20595	0.0	false
r101	3.7	463	53	0.0	2324.1	4955	0.0	false
r102	4.4	782	57	0.0	2161.1	12389	0.0	false
r103	4.9	2409	39	2.0	1985.2	36002	0.0	false
r104	20.7	3959	11	18.7	2133.2	36041	0.0	true
r105	3.6	1054	107	0.0	2093.2	22859	0.0	false
r106	8.8	2027	17	6.4	2081.7	36150	0.0	true
r107	19.0	2823	11	17.1	2169.7	36004	0.0	true
r108	24.2	4511	9	22.5	2181.4	36058	0.0	false
r109	5.6	1893	37	3.4	1972.5	36003	0.0	true
r110	18.6	2118	15	14.8	2126.7	36137	0.0	false
r111	16.9	2061	15	16.2	2128.1	36023	0.0	false
r112	7.6	3052	11	5.6	1878.7	36176	0.0	false
rc101	5.2	697	93	0.0	2480.0	14839	0.0	false
rc102	5.1	1106	21	3.7	2390.3	36039	0.0	false
rc103	5.3	1631	15	3.8	2306.5	36048	0.0	false
rc104	5.5	2225	17	3.5	2254.6	36037	0.0	false
rc105	5.5	1084	23	3.4	2447.3	36021	0.0	false
rc106	5.6	1015	93	2.9	2356.5	36001	0.0	false
rc107	4.4	1137	51	0.0	2250.0	16233	0.0	false
rc108	4.3	2360	61	1.4	2217.2	36014	0.0	false

Table 13: Results of experiments on instances of set H - single trip

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
c201	0.1	811	3	0.0	1565.1	1378	0.0	false
c202	0.9	1755	7	0.0	1558.3	6779	0.0	false
c203	1.6	2598	15	0.0	1543.5	14753	0.0	false
c204	4.4	3529	17	3.8	1560.3	36004	0.0	false
c205	0.3	1073	3	0.0	1550.7	2005	0.0	false
c206	1.2	1426	25	0.0	1548.6	16020	0.0	false
c207	0.8	1343	5	0.0	1540.1	4480	0.0	false
c208	1.3	1438	13	0.0	1535.3	11117	0.0	false
r101	1.0	549	11	0.0	2324.1	2100	0.0	false
r102	1.5	1027	25	0.0	2161.1	5521	0.0	false
r103	3.2	3713	15	2.9	2008.5	36148	0.0	true
r104	3.6	4098	11	2.7	1888.6	36020	0.0	false
r105	0.8	1192	31	0.0	2093.2	8854	0.0	false
r106	2.9	2454	15	2.5	2031.8	36015	0.0	false
r107	1.8	2568	15	1.3	1911.5	36027	0.0	false
r108	∞	5577	9	∞	∞	36019	-	-
r109	∞	1888	19	∞	∞	36033	-	-
r110	1.9	2995	15	1.5	1892.1	36028	0.0	true
r111	2.6	2906	15	2.1	1924.5	36025	0.0	true
r112	∞	4243	9	∞	∞	36073	-	-
rc101	1.6	825	23	0.0	2480.0	4980	0.0	false
rc102	4.0	1184	13	3.5	2445.3	36178	0.0	false
rc103	1.3	1858	15	0.0	2297.3	10279	0.0	false
rc104	1.7	2649	21	1.2	2256.3	36020	0.0	false
rc105	1.5	1169	45	0.0	2426.4	20887	0.0	false
rc106	4.5	1049	33	4.1	2410.8	36007	0.0	false
rc107	0.5	1515	9	0.0	2248.8	8073	0.0	false
rc108	0.4	2145	3	0.0	2217.2	3918	0.0	false

D Detailed BCP results for the multi-trip instances with 25 customers

Table 14: Results of experiments on instances of set G - multi trip

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
c101	0.0	4	1	0.0	595.9	4	0.0	false
c102	3.3	14	3	0.0	584.6	52	0.0	false
c103	2.7	16	3	0.0	566.3	54	0.0	false
c104	3.2	19	3	0.0	560.0	51	0.0	false
c105	0.0	5	1	0.0	595.9	5	0.0	false
c106	0.0	3	1	0.0	595.9	3	0.0	false
c107	0.0	6	1	0.0	590.6	7	0.0	false
c108	0.0	12	1	0.0	582.0	12	0.0	false
c109	1.3	6	3	0.0	565.0	18	0.0	false

Table 15: Results of experiments on instances of set H - multi trip

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
c201	0.0	3	1	0.0	555.7	3	0.0	false
c202	0.0	64	1	0.0	520.1	64	0.0	false
c203	0.0	96	1	0.0	516.5	96	0.0	false
c204	1.2	243	27	0.0	511.9	4191	0.0	false
c205	0.0	10	1	0.0	553.1	10	0.0	false
c206	0.6	58	3	0.0	553.1	130	0.0	false
c207	1.8	64	3	0.0	549.5	154	0.0	false
c208	0.0	28	1	0.0	520.6	28	0.0	false
r101	0.0	4	1	0.0	1122.2	4	0.0	false
r102	0.0	12	1	0.0	1025.0	12	0.0	false
r103	0.1	48	3	0.0	878.8	102	0.0	false
r104	0.0	38	1	0.0	815.8	38	0.0	false
r105	0.0	14	3	0.0	976.5	33	0.0	false
r106	0.0	41	1	0.0	889.2	41	0.0	false
r107	0.0	50	1	0.0	823.5	50	0.0	false
r108	0.2	101	3	0.0	802.1	219	0.0	false
r109	0.0	11	1	0.0	850.9	12	0.0	false
r110	0.0	19	1	0.0	840.9	19	0.0	false
r111	0.0	18	1	0.0	827.6	18	0.0	false
r112	0.5	95	3	0.0	797.9	200	0.0	false
rc101	0.0	1	1	0.0	1002.7	1	0.0	false
rc102	0.0	3	1	0.0	932.1	5	116.0	false
rc103	0.0	8	1	0.0	880.3	9	94.0	false
rc104	1.4	18	3	0.0	859.8	43	102.0	false
rc105	0.0	3	1	0.0	991.2	4	96.0	false
rc106	0.0	2	1	0.0	909.7	3	64.0	false
rc107	0.0	3	1	0.0	868.0	4	51.0	false
rc108	0.0	7	1	0.0	850.8	9	52.0	false

Table 16: Results of experiments on instances of set I - multi trip

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
r201	0.0	118	1	0.0	842.2	118	0.0	false
r202	0.0	2009	3	0.0	786.5	2951	0.0	false
r203	1.7	3400	5	0.0	741.6	5376	0.0	false
r204	∞	4520	9	∞	∞	36297	-	-
r205	2.0	976	3	0.0	748.3	2017	0.0	false

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
r206	0.0	3160	1	0.0	682.5	3160	0.0	false
r207	0.2	6019	3	0.0	661.5	11763	0.0	false
r208	0.0	9819	1	0.0	612.7	9819	0.0	false
r209	∞	974	19	∞	∞	36094	-	-
r210	3.0	1661	5	0.0	748.7	6148	0.0	false
r211	∞	2520	17	∞	∞	36023	-	-
rc201	0.0	8	1	0.0	711.0	9	0.0	false
rc202	2.0	59	3	0.0	688.7	131	0.0	false
rc203	1.6	298	3	0.0	673.9	549	0.0	false
rc204	1.0	261	5	0.0	640.0	879	0.0	false
rc205	0.0	22	1	0.0	689.4	22	0.0	false
rc206	0.0	66	1	0.0	675.9	66	0.0	false
rc207	0.9	190	3	0.0	642.8	404	0.0	false
rc208	2.0	504	3	0.0	596.3	786	0.0	false

E Detailed BCP results for the multi-trip instances with 50 customers

Table 17: Results of experiments on instances of set G - multi trip

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
c101	1.0	15	3	0.0	1258.7	47	0.0	false
c102	0.0	28	1	0.0	1203.0	28	0.0	false
c103	0.5	70	3	0.0	1172.6	200	0.0	false
c104	2.0	120	5	0.0	1140.8	713	0.0	false
c105	0.2	12	3	0.0	1220.7	29	0.0	false
c106	1.4	22	5	0.0	1256.9	135	0.0	false
c107	0.5	15	3	0.0	1200.2	38	0.0	false
c108	0.6	34	3	0.0	1192.9	91	0.0	false
c109	1.5	32	5	0.0	1157.8	168	0.0	false

Table 18: Results of experiments on instances of set H - multi trip

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
c201	0.4	70	3	0.0	858.1	174	0.0	false
c202	1.0	849	5	0.0	843.9	2940	0.0	false
c203	2.3	1422	9	0.0	835.6	6537	0.0	false
c204	0.7	3820	3	0.0	778.6	6748	0.0	false
c205	1.4	235	5	0.0	854.4	1065	0.0	false
c206	2.6	388	45	0.0	851.0	11307	0.0	false
c207	5.3	812	57	3.5	857.7	36002	0.0	false
c208	0.2	870	3	0.0	810.6	1579	0.0	false
r101	0.0	30	3	0.0	1666.0	67	0.0	false
r102	0.0	54	1	0.0	1484.7	54	0.0	false
r103	0.2	352	3	0.0	1336.1	610	0.0	false
r104	0.5	919	3	0.0	1140.8	1705	13.0	false
r105	0.0	125	3	0.0	1472.4	240	0.0	false
r106	0.1	197	3	0.0	1329.8	377	0.0	false
r107	0.6	649	3	0.0	1242.8	1094	92.0	false
r108	0.3	894	3	0.0	1119.4	1690	3.0	false
r109	0.5	262	3	0.0	1311.4	706	111.0	false
r110	0.2	433	3	0.0	1242.5	790	201.0	false
r111	0.0	495	3	0.0	1217.9	876	0.0	false
r112	1.2	595	5	0.0	1163.2	2373	87.0	false
rc101	0.1	6	3	0.0	1845.8	8	0.0	false

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
rc102	0.3	26	3	0.0	1753.2	34	0.0	false
rc103	1.5	57	3	0.0	1666.8	167	92.0	false
rc104	0.8	77	3	0.0	1510.9	175	94.0	false
rc105	0.4	15	3	0.0	1802.6	29	0.0	false
rc106	0.3	12	3	0.0	1759.7	23	0.0	false
rc107	1.7	71	3	0.0	1662.9	164	79.0	false
rc108	2.1	48	3	0.0	1589.9	111	0.0	false

Table 19: Results of experiments on instances of set I - multi trip

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
r201	0.0	1714	1	0.0	1213.8	1714	0.0	false
r202	∞	7064	5	∞	∞	36336	-	-
r203	∞	22397	1	∞	∞	36131	-	-
r204	-	-	-	-	-	-	-	-
r205	∞	4432	9	∞	∞	36759	-	-
r206	∞	10852	3	∞	∞	37813	-	-
r207	-	-	-	-	-	-	-	-
r208	∞	36375	1	∞	∞	36375	-	-
r209	∞	8579	1	∞	∞	8582	-	-
r210	∞	10202	7	∞	∞	36692	-	-
r211	∞	27573	1	∞	∞	37270	-	-
rc201	0.0	131	1	0.0	1129.4	131	0.0	false
rc202	0.3	1407	3	0.0	1051.0	2092	0.0	false
rc203	3.1	2385	11	0.0	989.1	12036	541.0	false
rc204	-	-	-	-	-	-	-	-
rc205	0.0	319	1	0.0	1063.7	319	0.0	false
rc206	0.9	2246	3	0.0	1042.1	3588	0.0	false
rc207	3.3	1121	37	0.0	983.3	28915	92.0	false
rc208	-	-	-	-	-	-	-	-

F Detailed BCP results for the multi-trip instances with 75 customers

Table 20: Results of experiments on instances of set G - multi trip

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
c101	1.7	137	37	0.0	1786.3	1902	486.0	false
c102	1.2	520	13	0.0	1740.7	3396	1040.0	false
c103	1.6	1078	27	0.0	1657.8	13353	707.0	false
c104	2.3	1413	27	0.5	1583.0	36026	0.0	false
c105	1.4	105	3	0.0	1705.3	234	516.0	false
c106	1.9	194	35	0.0	1748.8	4227	1294.0	false
c107	1.1	266	45	0.0	1709.7	1705	267.0	false
c108	1.9	511	69	0.0	1699.2	16128	195.0	false
c109	∞	800	35	∞	∞	36047	-	-

Table 21: Results of experiments on instances of set H - multi trip

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
c201	0.6	342	3	0.0	1082.5	708	1195.0	false
c202	0.6	5742	7	0.0	1080.5	22260	925.0	false
c203	∞	6066	1	∞	∞	11024	-	-
c204	-	-	-	-	-	-	-	-
c205	0.0	2317	1	0.0	1066.8	2317	6697.0	false
c206	0.4	4072	3	0.0	1068.4	5950	3676.0	false
c207	∞	2550	5	∞	∞	22395	-	-
c208	-	-	-	-	-	-	-	-
r101	3.3	176	13	0.0	2521.4	614	81.0	false
r102	2.2	832	25	0.0	2298.3	5757	0.0	false
r103	3.2	2069	81	0.0	1975.2	30617	0.0	false
r104	7.0	2678	19	4.4	1735.8	36022	122.0	false
r105	2.7	957	23	0.0	2181.0	17272	164.0	false
r106	4.4	1954	27	1.9	2045.3	36002	292.0	false
r107	6.2	1793	33	4.0	1868.0	36168	151.0	false
r108	∞	2774	19	∞	∞	36060	-	-
r109	5.1	1794	15	4.3	2001.2	36021	49.0	false
r110	4.3	2475	17	3.5	1878.2	36008	63.0	false
r111	3.7	2594	31	2.8	1831.2	36030	7.0	false
r112	8.4	1727	21	6.3	1809.4	36009	174.0	false
rc101	1.1	1022	5	0.0	2444.1	4700	295.0	true
rc102	1.4	1505	33	0.0	2306.5	24844	186.0	false
rc103	2.2	1786	11	0.0	2123.3	18294	349.0	true
rc104	∞	2994	21	∞	∞	36014	-	-
rc105	1.2	1879	13	0.0	2331.4	26099	134.0	false
rc106	2.1	1819	13	0.0	2223.0	16167	225.0	false
rc107	3.0	2120	7	1.4	2125.2	36038	554.0	false
rc108	2.6	3047	7	0.8	2037.0	36034	51.0	false

Table 22: Results of experiments on instances of set I - multi trip

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
r201	∞	25338	1	∞	∞	36062	-	-
rc201	∞	26284	1	∞	∞	37595	-	-

We could obtain a lower bound only for 2 instances in Set I.

G Detailed BCP results for instances with modified vehicle capacity

Table 23: Results of experiments on instances of set D

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
Ca1-2-3-15	0.0	1	1	0.0	612.385	1	0.0	false
Ca2-2-3-15	0.0	0	1	0.0	548.953	0	0.0	false
Ca3-2-3-15	0.0	0	1	0.0	551.985	0	0.0	false
Ca4-2-3-15	0.0	0	1	0.0	569.579	0	0.0	false
Ca5-2-3-15	0.0	0	1	0.0	555.796	0	0.0	false
Cb1-2-3-15	0.0	1	1	0.0	624.178	1	0.0	false
Cb2-2-3-15	0.0	1	1	0.0	516.739	1	0.0	false
Cb3-2-3-15	0.0	1	1	0.0	601.897	1	0.0	false
Cb4-2-3-15	0.0	1	1	0.0	546.314	1	0.0	false

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
Cb5-2-3-15	0.0	1	1	0.0	534.725	1	0.0	false
Cc1-2-3-15	0.0	1	1	0.0	586.856	1	0.0	false
Cc2-2-3-15	0.0	1	1	0.0	482.985	1	0.0	false
Cc3-2-3-15	0.0	1	1	0.0	539.685	1	0.0	false
Cc4-2-3-15	0.0	1	1	0.0	562.798	1	0.0	false
Cc5-2-3-15	0.0	1	1	0.0	483.593	1	0.0	false
Cd1-2-3-15	0.0	1	1	0.0	597.698	1	0.0	false
Cd2-2-3-15	0.0	1	1	0.0	483.133	1	0.0	false
Cd3-2-3-15	0.0	1	1	0.0	553.24	1	0.0	false
Cd4-2-3-15	0.0	1	1	0.0	585.301	1	0.0	false
Cd5-2-3-15	0.0	0	1	0.0	536.764	0	0.0	false
Ca1-3-5-15	0.0	2	1	0.0	603.456	2	0.0	false
Ca2-3-5-15	0.0	2	1	0.0	628.405	2	0.0	false
Ca3-3-5-15	0.0	2	1	0.0	527.679	2	0.0	false
Ca4-3-5-15	0.0	2	1	0.0	514.399	2	0.0	false
Ca5-3-5-15	0.0	3	1	0.0	521.399	3	0.0	false
Cb1-3-5-15	0.0	6	1	0.0	596.41	6	0.0	false
Cb2-3-5-15	0.0	6	1	0.0	613.518	6	0.0	false
Cb3-3-5-15	0.0	5	1	0.0	537.256	6	0.0	false
Cb4-3-5-15	0.7	12	3	0.0	500.048	19	0.0	false
Cb5-3-5-15	0.0	6	1	0.0	538.57	6	0.0	false
Cc1-3-5-15	0.0	5	1	0.0	519.769	5	0.0	false
Cc2-3-5-15	0.0	3	1	0.0	549.425	3	0.0	false
Cc3-3-5-15	0.0	4	1	0.0	522.837	4	0.0	false
Cc4-3-5-15	0.0	5	1	0.0	506.35	5	0.0	false
Cc5-3-5-15	0.0	3	1	0.0	545.077	3	0.0	false
Cd1-3-5-15	0.0	4	1	0.0	568.629	4	0.0	false
Cd2-3-5-15	0.0	2	1	0.0	571.299	2	0.0	false
Cd3-3-5-15	0.0	2	1	0.0	522.8	2	0.0	false
Cd4-3-5-15	0.2	4	3	0.0	506.825	7	0.0	false
Cd5-3-5-15	0.0	3	1	0.0	544.084	3	0.0	false
Ca1-6-4-15	0.0	2	1	0.0	551.457	2	0.0	false
Ca2-6-4-15	0.0	3	1	0.0	560.919	3	0.0	false
Ca3-6-4-15	0.0	3	1	0.0	556.642	3	0.0	false
Ca4-6-4-15	0.0	3	1	0.0	481.062	3	0.0	false
Ca5-6-4-15	0.0	2	1	0.0	459.529	2	0.0	false
Cb1-6-4-15	0.0	4	1	0.0	567.151	4	0.0	false
Cb2-6-4-15	0.0	5	1	0.0	631.512	5	0.0	false
Cb3-6-4-15	0.0	5	1	0.0	561.536	5	0.0	false
Cb4-6-4-15	0.0	4	1	0.0	510.954	4	0.0	false
Cb5-6-4-15	0.0	4	1	0.0	473.939	4	0.0	false
Cc1-6-4-15	0.0	3	1	0.0	566.013	3	0.0	false
Cc2-6-4-15	0.0	3	1	0.0	549.229	3	0.0	false
Cc3-6-4-15	0.0	3	1	0.0	540.875	3	0.0	false
Cc4-6-4-15	0.0	3	1	0.0	521.621	3	0.0	false
Cc5-6-4-15	0.0	2	1	0.0	462.787	2	0.0	false
Cd1-6-4-15	0.0	2	1	0.0	551.492	2	0.0	false
Cd2-6-4-15	0.0	2	1	0.0	554.83	3	0.0	false
Ca1-2-3-30	0.1	2	3	0.0	983.398	4	0.0	false
Ca2-2-3-30	0.3	2	3	0.0	973.926	3	0.0	false
Ca3-2-3-30	0.0	1	1	0.0	1073.453	1	0.0	false
Ca4-2-3-30	1.1	3	3	0.0	955.487	7	0.0	false
Ca5-2-3-30	1.0	4	3	0.0	978.477	8	0.0	false
Cb1-2-3-30	0.0	5	3	0.0	985.763	7	0.0	false
Cb2-2-3-30	0.0	5	3	0.0	939.308	7	0.0	false
Cb3-2-3-30	0.0	2	1	0.0	1076.254	2	0.0	false
Cb4-2-3-30	1.1	4	3	0.0	992.625	8	0.0	false
Cb5-2-3-30	1.2	5	3	0.0	925.364	8	0.0	false
Cc1-2-3-30	0.0	3	1	0.0	870.373	3	0.0	false
Cc2-2-3-30	0.0	1	1	0.0	937.712	1	0.0	false
Cc3-2-3-30	0.3	3	3	0.0	1015.082	8	0.0	false
Cc4-2-3-30	0.0	2	1	0.0	949.066	2	0.0	false
Cc5-2-3-30	1.1	6	3	0.0	962.579	18	0.0	false
Cd1-2-3-30	0.0	2	1	0.0	878.924	2	0.0	false
Cd2-2-3-30	0.0	1	1	0.0	945.304	1	0.0	false
Cd3-2-3-30	0.0	2	1	0.0	1031.949	2	0.0	false
Cd4-2-3-30	0.0	1	1	0.0	959.553	1	0.0	false
Cd5-2-3-30	1.4	4	3	0.0	979.735	16	0.0	false

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
Ca1-3-5-30	0.1	9	3	0.0	892.746	13	0.0	false
Ca2-3-5-30	0.3	8	3	0.0	966.054	15	0.0	false
Ca3-3-5-30	0.5	8	3	0.0	936.106	15	0.0	false
Ca4-3-5-30	0.0	4	1	0.0	909.869	6	0.0	false
Ca5-3-5-30	0.6	6	3	0.0	900.618	13	0.0	false
Cb1-3-5-30	0.7	23	3	0.0	935.466	33	0.0	false
Cb2-3-5-30	0.6	26	3	0.0	958.302	37	0.0	false
Cb3-3-5-30	1.3	24	3	0.0	919.361	36	0.0	false
Cb4-3-5-30	0.2	26	3	0.0	903.874	37	0.0	false
Cb5-3-5-30	0.4	36	3	0.0	977.906	75	0.0	false
Cc1-3-5-30	0.8	11	3	0.0	948.826	22	0.0	false
Cc2-3-5-30	0.2	15	3	0.0	904.011	28	0.0	false
Cc3-3-5-30	0.4	13	3	0.0	957.422	29	0.0	false
Cc4-3-5-30	0.3	15	3	0.0	892.43	23	0.0	false
Cc5-3-5-30	0.0	12	1	0.0	847.761	12	0.0	false
Cd1-3-5-30	1.1	9	3	0.0	968.928	17	0.0	false
Cd2-3-5-30	0.5	9	3	0.0	951.243	18	0.0	false
Cd3-3-5-30	0.1	10	3	0.0	919.623	14	0.0	false
Cd4-3-5-30	0.1	9	3	0.0	918.942	15	0.0	false
Cd5-3-5-30	0.7	12	3	0.0	924.844	30	0.0	false
Ca1-6-4-30	0.1	6	3	0.0	929.931	10	0.0	false
Ca2-6-4-30	1.9	7	3	0.0	967.845	13	0.0	false
Ca3-6-4-30	0.0	4	1	0.0	869.334	4	0.0	false
Ca4-6-4-30	0.2	7	3	0.0	872.009	13	0.0	false
Ca5-6-4-30	0.2	5	3	0.0	858.376	10	0.0	false
Cb1-6-4-30	0.3	13	3	0.0	971.993	18	0.0	false
Cb2-6-4-30	0.0	13	1	0.0	920.495	14	0.0	false
Cb3-6-4-30	1.1	13	3	0.0	942.553	30	0.0	false
Cb4-6-4-30	0.0	17	1	0.0	877.54	17	0.0	false
Cb5-6-4-30	0.0	10	1	0.0	895.742	10	0.0	false
Cc1-6-4-30	0.4	9	3	0.0	869.343	17	0.0	false
Cc2-6-4-30	0.3	8	3	0.0	922.96	15	0.0	false
Cc3-6-4-30	1.3	8	3	0.0	905.357	19	0.0	false
Cc4-6-4-30	0.2	8	3	0.0	839.842	17	0.0	false
Cc5-6-4-30	0.6	8	3	0.0	820.771	19	0.0	false
Cd1-6-4-30	0.3	6	3	0.0	925.222	11	0.0	false
Cd2-6-4-30	0.2	6	3	0.0	917.259	10	0.0	false
Cd3-6-4-30	0.2	6	3	0.0	880.593	11	0.0	false
Cd4-6-4-30	0.0	7	3	0.0	894.445	11	0.0	false
Cd5-6-4-30	0.5	6	3	0.0	948.597	15	0.0	false
Ca1-2-3-50	0.3	2	3	0.0	1603.5	4	0.0	true
Ca2-2-3-50	0.1	6	3	0.0	1679.722	9	0.0	false
Ca3-2-3-50	0.1	5	3	0.0	1565.405	8	0.0	false
Ca4-2-3-50	0.1	5	3	0.0	1571.975	8	0.0	true
Ca5-2-3-50	1.4	9	5	0.0	1427.603	45	0.0	true
Cb1-2-3-50	0.3	14	3	0.0	1464.275	20	0.0	true
Cb2-2-3-50	0.4	15	3	0.0	1469.438	24	0.0	false
Cb3-2-3-50	0.6	12	3	0.0	1574.918	22	0.0	false
Cb4-2-3-50	1.1	13	7	0.0	1486.735	53	0.0	true
Cb5-2-3-50	0.6	12	3	0.0	1417.297	29	0.0	true
Cc1-2-3-50	1.1	10	9	0.0	1431.261	93	0.0	true
Cc2-2-3-50	0.7	7	7	0.0	1488.418	50	0.0	true
Cc3-2-3-50	0.6	11	21	0.0	1391.61	155	0.0	true
Cc4-2-3-50	0.8	9	3	0.0	1436.095	29	0.0	true
Cc5-2-3-50	0.4	8	3	0.0	1537.902	18	0.0	false
Cd1-2-3-50	0.5	9	3	0.0	1471.697	19	0.0	true
Cd2-2-3-50	0.3	8	3	0.0	1669.405	13	0.0	true
Cd3-2-3-50	0.0	7	1	0.0	1530.572	8	0.0	false
Cd4-2-3-50	0.1	11	3	0.0	1538.753	16	0.0	false
Cd5-2-3-50	0.2	7	3	0.0	1536.029	11	0.0	true
Ca1-3-5-50	0.4	15	3	0.0	1433.252	24	0.0	false
Ca2-3-5-50	0.2	15	3	0.0	1426.697	29	0.0	false
Ca3-3-5-50	2.8	24	1045	0.0	1315.377	16144	0.0	true
Ca4-3-5-50	2.9	25	89	0.0	1358.037	1312	0.0	true
Ca5-3-5-50	1.4	20	9	0.0	1293.626	122	0.0	true
Cb1-3-5-50	2.5	77	85	0.0	1386.963	3331	0.0	true
Cb2-3-5-50	0.3	49	3	0.0	1453.784	106	0.0	false
Cb3-3-5-50	1.3	66	5	0.0	1458.2	197	0.0	true

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
Cb4-3-5-50	0.6	73	3	0.0	1375.054	147	0.0	false
Cb5-3-5-50	1.9	64	7	0.0	1292.447	218	0.0	true
Cc1-3-5-50	3.5	31	2543	2.5	1361.007	36000	0.0	true
Cc2-3-5-50	0.6	18	3	0.0	1399.217	42	0.0	true
Cc3-3-5-50	0.1	20	3	0.0	1365.423	44	0.0	false
Cc4-3-5-50	1.0	29	3	0.0	1426.291	106	0.0	false
Cc5-3-5-50	0.3	25	3	0.0	1348.234	52	0.0	false
Cd1-3-5-50	3.2	23	2889	2.0	1373.418	36000	0.0	false
Cd2-3-5-50	0.5	16	3	0.0	1404.434	38	0.0	false
Cd3-3-5-50	0.3	23	3	0.0	1428.076	50	0.0	false
Cd4-3-5-50	0.3	17	3	0.0	1396.529	38	0.0	false
Cd5-3-5-50	4.3	19	271	0.0	1356.708	4566	0.0	false
Ca1-6-4-50	0.4	10	3	0.0	1394.474	17	0.0	false
Ca2-6-4-50	0.4	14	3	0.0	1444.019	33	0.0	false
Ca3-6-4-50	1.0	11	5	0.0	1417.399	45	0.0	false
Ca4-6-4-50	0.7	11	3	0.0	1330.621	30	0.0	false
Ca5-6-4-50	2.9	25	15	0.0	1343.962	222	0.0	false
Cb1-6-4-50	0.6	28	3	0.0	1430.514	48	0.0	false
Cb2-6-4-50	0.8	39	3	0.0	1522.565	100	0.0	false
Cb3-6-4-50	3.9	37	173	0.0	1423.962	2757	0.0	false
Cb4-6-4-50	3.1	30	23	0.0	1347.348	324	0.0	false
Cb5-6-4-50	3.6	43	713	0.0	1404.883	12393	0.0	true
Cc1-6-4-50	0.2	14	3	0.0	1334.999	28	0.0	false
Cc2-6-4-50	3.8	18	1741	0.0	1355.826	27631	0.0	true
Cc3-6-4-50	0.9	17	3	0.0	1407.002	47	0.0	false
Cc4-6-4-50	3.6	16	227	0.0	1272.849	3436	0.0	false
Cc5-6-4-50	0.4	15	3	0.0	1468.821	31	0.0	true
Cd1-6-4-50	0.0	18	3	0.0	1365.653	23	0.0	true
Cd2-6-4-50	0.8	14	3	0.0	1416.28	42	0.0	true
Cd3-6-4-50	1.1	10	3	0.0	1463.237	37	0.0	true
Cd4-6-4-50	2.6	22	9	0.0	1300.999	115	0.0	false
Cd5-6-4-50	0.5	16	3	0.0	1407.247	43	0.0	false
Ca1-2-3-100	0.5	17	3	0.0	2856.269	52	0.0	true
Ca2-2-3-100	0.9	27	661	0.0	2975.808	11217	0.0	true
Ca3-2-3-100	0.5	19	79	0.0	2773.629	1468	0.0	true
Ca4-2-3-100	0.5	13	9	0.0	3139.773	196	0.0	true
Ca5-2-3-100	0.4	13	9	0.0	2854.0	154	0.0	true
Cb1-2-3-100	0.5	58	645	0.0	2564.154	9462	0.0	true
Cb2-2-3-100	1.8	39	2317	1.4	2790.905	36000	0.0	true
Cb3-2-3-100	0.4	44	25	0.0	**2731.597**	526	0.0	true
Cb4-2-3-100	0.9	43	301	0.0	2953.158	4927	0.0	true
Cb5-2-3-100	0.7	38	65	0.0	2818.346	995	0.0	true
Cc1-2-3-100	0.4	26	99	0.0	2861.586	1423	0.0	true
Cc2-2-3-100	0.4	23	9	0.0	2761.255	169	0.0	true
Cc3-2-3-100	0.7	24	419	0.0	2774.704	10232	0.0	true
Cc4-2-3-100	0.5	17	95	0.0	3001.084	960	0.0	true
Cc5-2-3-100	0.3	15	49	0.0	2847.753	563	0.0	true
Cd1-2-3-100	0.2	17	3	0.0	2746.927	34	0.0	true
Cd2-2-3-100	0.8	21	585	0.0	2975.524	14320	0.0	true
Cd3-2-3-100	0.4	15	21	0.0	2651.506	342	0.0	true
Cd4-2-3-100	0.5	21	11	0.0	3000.405	184	0.0	true
Cd5-2-3-100	0.3	20	31	0.0	2893.672	255	0.0	true
Ca1-3-5-100	1.2	34	1871	0.7	2726.408	36001	0.0	true
Ca2-3-5-100	0.4	37	7	0.0	2559.522	172	0.0	true
Ca3-3-5-100	1.3	56	105	0.0	2524.226	2743	0.0	true
Ca4-3-5-100	2.2	61	1325	2.0	2803.123	36001	0.0	true
Ca5-3-5-100	1.1	52	1035	0.9	2738.777	36001	0.0	true
Cb1-3-5-100	0.9	192	161	0.0	2747.783	8716	0.0	true
Cb2-3-5-100	4.1	225	293	3.5	2598.631	36001	0.0	true
Cb3-3-5-100	4.8	218	161	4.3	2700.236	36003	0.0	true
Cb4-3-5-100	1.9	275	763	1.6	2841.227	36000	0.0	true
Cb5-3-5-100	1.5	191	695	1.1	2884.362	36001	0.0	true
Cc1-3-5-100	1.6	70	1021	1.4	2702.51	36000	0.0	true
Cc2-3-5-100	0.3	67	15	0.0	2513.607	411	0.0	true
Cc3-3-5-100	2.0	77	897	1.7	2690.776	36001	0.0	true
Cc4-3-5-100	1.3	80	1121	1.0	2697.893	36003	0.0	true
Cc5-3-5-100	0.6	76	213	0.0	2759.097	8721	0.0	true
Cd1-3-5-100	0.3	54	7	0.0	3094.081	161	0.0	true

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
Cd2-3-5-100	0.4	55	57	0.0	2525.283	1311	0.0	true
Cd3-3-5-100	0.6	42	377	0.0	2632.836	9971	0.0	true
Cd4-3-5-100	0.7	47	989	0.0	2632.315	24894	0.0	true
Cd5-3-5-100	1.4	66	821	1.0	2802.845	36000	0.0	true
Ca1-6-4-100	0.4	35	41	0.0	2622.746	792	0.0	true
Ca2-6-4-100	1.7	36	1237	1.4	2500.997	36002	0.0	true
Ca3-6-4-100	0.6	33	11	0.0	2786.025	209	0.0	true
Ca4-6-4-100	0.8	29	1499	0.0	2474.52	35243	0.0	true
Ca5-6-4-100	0.9	37	831	0.0	2678.69	18389	0.0	true
Cb1-6-4-100	0.6	120	361	0.0	2458.361	9227	0.0	true
Cb2-6-4-100	3.4	134	337	2.9	2665.607	36000	0.0	true
Cb3-6-4-100	0.6	144	9	0.0	2723.519	825	0.0	true
Cb4-6-4-100	1.0	106	1417	0.8	2844.658	36000	0.0	true
Cb5-6-4-100	0.7	114	27	0.0	2654.829	838	0.0	true
Cc1-6-4-100	0.3	57	7	0.0	2611.41	270	0.0	true
Cc2-6-4-100	0.6	45	93	0.0	2650.476	2050	0.0	true
Cc3-6-4-100	0.8	64	5	0.0	2490.204	465	0.0	true
Cc4-6-4-100	0.2	53	5	0.0	2645.581	159	0.0	true
Cc5-6-4-100	0.3	45	5	0.0	2545.584	140	0.0	true
Cd1-6-4-100	0.7	32	149	0.0	2688.914	2696	0.0	true
Cd2-6-4-100	0.2	33	5	0.0	2627.338	111	0.0	true
Cd3-6-4-100	1.5	40	1699	1.1	2537.407	36001	0.0	true
Cd4-6-4-100	1.2	35	1531	1.0	2792.626	36000	0.0	true
Cd5-6-4-100	1.3	37	1581	0.9	2689.365	36000	0.0	true

Table 24: Results of experiments on instances of set I - multi trip

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
c101	1.7	127	19	0.0	1794.8	864	0.0	false
c102	1.0	500	9	0.0	1743.3	2234	463.0	false
c103	2.4	687	43	0.7	1672.4	36005	654.0	false
c104	2.7	1331	35	0.0	1592.8	32956	0.0	false
c105	3.2	103	19	0.0	1737.2	1383	1035.0	false
c106	2.1	204	39	0.0	1756.3	4309	237.0	false
c107	1.1	278	29	0.0	1710.7	2392	302.0	false
c108	1.9	508	27	0.0	1702.4	7689	195.0	false
c109	3.3	693	45	2.4	1673.1	36002	127.0	false

Table 25: Results of experiments on instances of set I - multi trip

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
c201	0.2	439	3	0.0	1028.6	1138	6840.0	false
c202	0.1	8703	3	0.0	1026.6	14929	6111.0	false
c203	0.5	22257	1	0.5	1015.5	37110	6112.0	false
c205	0.1	3065	3	0.0	1022.1	5350	2180.0	false
c206	0.8	4714	3	0.0	1018.1	19400	1907.0	false
c207	0.4	9996	5	0.0	1014.4	24035	1469.0	false
r101	1.8	273	5	0.0	2523.1	611	0.0	false
r102	1.9	811	27	0.0	2306.2	4794	81.0	false
r103	2.7	2327	13	0.0	1977.3	10045	114.0	false
r104	∞	3050	19	∞	∞	36017	-	-
r105	2.1	982	21	0.0	2181.0	13216	41.0	false
r106	3.2	1773	55	0.8	2036.9	36018	54.0	false
r107	5.7	1666	19	4.1	1869.9	36039	190.0	false
r108	∞	2956	19	∞	∞	36002	-	-
r109	4.0	1838	13	2.0	1992.8	36014	86.0	false
r110	6.3	2609	17	5.5	1926.4	36029	241.0	false
r111	3.6	3317	23	0.9	1844.1	36011	116.0	false

	BCP_{best} with primal heuristic						PostProc	
Instance	Rg (%)	Rt (s)	Nodes	Fg(%)	BPB	t (s)	Δ	Conso
r112	∞	1810	19	∞	∞	36038	-	-
rc101	1.2	1096	11	0.0	2449.9	11091	406.0	true
rc102	1.5	1635	15	0.0	2309.3	20043	409.0	true
rc103	2.3	2754	19	1.3	2133.8	36042	355.0	true
rc104	4.6	2149	19	3.9	2033.6	36024	170.0	false
rc105	1.5	2560	19	0.7	2343.1	36004	124.0	false
rc106	1.9	1801	7	0.0	2223.0	12302	379.0	true
rc107	2.3	2651	5	1.1	2122.7	36034	554.0	false
rc108	2.5	2844	9	0.0	2037.0	35744	406.0	true