**Submitted to**
**manuscript**



# Non-Elementary Formulations for Single Vehicle Routing Problems with Pickups and Deliveries

Bruno P. Bruck
Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Via Amendola 2, 42122 Reggio Emilia, Italy, bruno.petratobruck@unimore.it

Manuel Iori
Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Via Amendola 2, 42122 Reggio Emilia, Italy, manuel.iori@unimore.it

We study the class of one-to-many-to-one single vehicle routing problems with pickups and deliveries, in which a single capacitated vehicle is used to serve a set of customers requiring a delivery, a pickup, or both. These problems have many real-world applications, including beverage distribution, courier service transportation, and reverse logistics. We first concentrate on a well-studied problem in this class, known as the *single vehicle routing problem with deliveries and selective pickups* (SVRPDSP), in which deliveries are mandatory but pickups are optional and generate a revenue if performed, and customers requiring both a delivery and a pickup (combined demand) can be visited either once or twice.

Most exact algorithms in the literature solve SVRPDSP by looking for Elementary tours on an extended network which is obtained by transforming each combined demand customer into two different customers, one requiring only the delivery and the other one only the pickup. Because this can result in a significant loss in performance, in this work we focus instead on the original problem network and present formulations that can yield non-Elementary tours. Through the use of Benders Decomposition, valid inequalities, and tailored optimization techniques based on branch-and-cut frameworks, we develop exact algorithms that outmatch previous results in the literature and obtain proven optimal solutions for all benchmark instances. We then generalize the algorithms to solve several other vehicle routing problems with pickups and deliveries, including the cases of split deliveries, intermediate dropoffs, mandatory pickups, and multiple vehicles.

*Key words*: pickup and delivery; one-to-many-to-one; selective pickups; valid inequalities; branch-and-cut

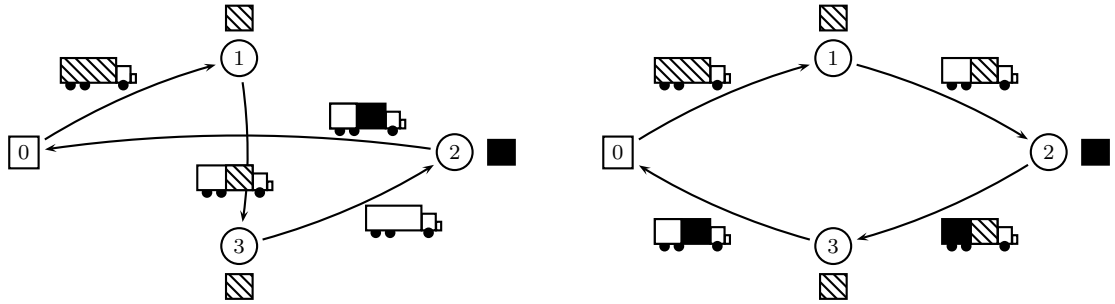
# 1. Introduction

This article addresses a large class of *pickup and delivery* routing problems, where a single vehicle leaves the depot to perform both a series of deliveries of a first commodity and a series of collections of a second commodity which is brought back to the depot. These problems are known as *one-to-many-to-one single vehicle pickup and delivery problems* (1-M-1 SVPDPs), see Berbeglia et al. (2007) and Gribkovskaia and Laporte (2008), and are among the most important problems in the large area of pickup and delivery because they can model many real-world situations. The classical example found in textbooks arises in *beverage distribution*, and is that of a capacitated vehicle that delivers full drink bottles to customers and returns empty bottles to the depot (see, e.g., Golden et al. 2002). Several other interesting applications can be found, among which we name the distribution in the European *electrical and electronic market*, where the WEEE directive imposes that the delivery of new appliances should be accompanied by the pickup of old ones, *courier service transportation*, where the courier delivers packages to customers and brings back to the depot other packages to be shipped later on, and in general the large area of *reverse logistics*, whose aim is to minimize the kilometers traveled with an empty load (see, e.g., Daniel et al. 2009).

To better model situations arising in various real-world contexts, different problem variants have been studied. The relevant literature has focused on two classes of problems:

- *single demand* (SD): problem instances contain only customers that require either a delivery or a pickup, but not both;
- *combined demand* (CD): problem instances may contain any type of customers.

Figure 1 shows two important SD cases. The delivery commodity is depicted with dashed lines and the pickup commodity in black. Vertex 0 represents the depot, vertices 1 and 3 are *delivery customers* (also referred to as *linehaul customers* in the literature), and vertex 2 is a *pickup customer* (also known as *backhaul customer*). Figure 1-(a) depicts the *SVPDP with backhauls*, in which all deliveries are performed before any pickup is made. In Figure 1-(b) the backhaul constraint is relaxed and *mixed deliveries* are allowed, and thus pickup and delivery operations can be performed

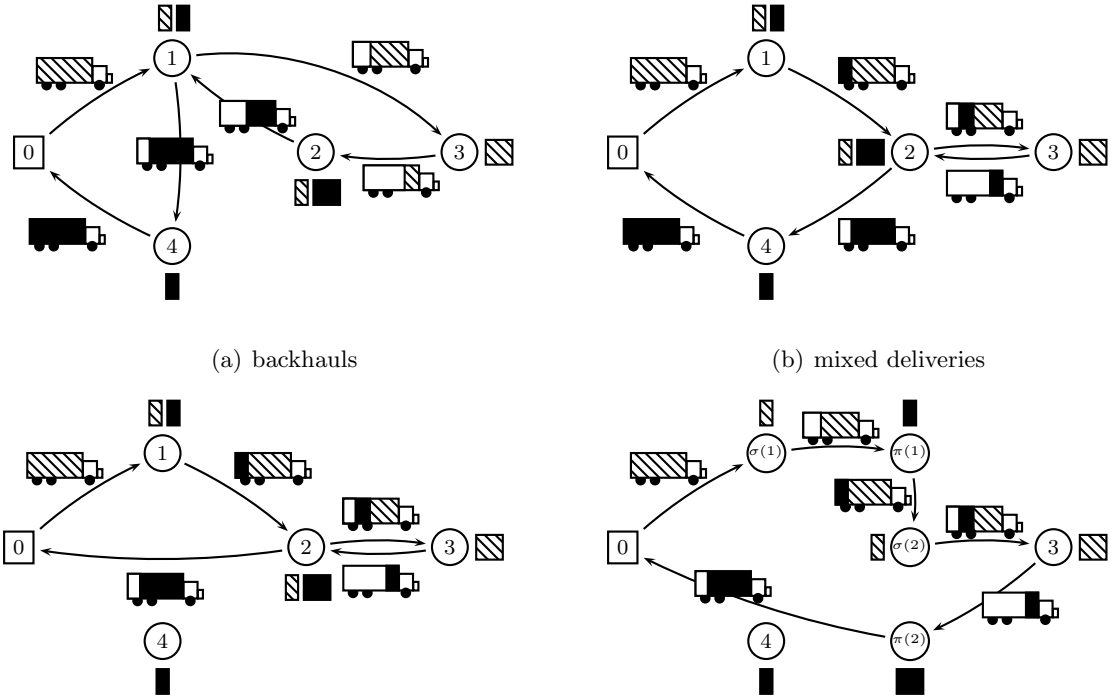



(a) backhauls (b) mixed deliveries

**Figure 1 Two common 1-M-1 SVPDPs single demand variants (vertex 0 is the depot, hatched boxes are deliveries, and solid boxes are pickups).**

in any order. This problem, called *mixed SVPDP*, is less constrained than the former. Its solution could possibly require reshuffling of the cargo, but usually consists of a shorter route.

Regarding the CD case, if only a *single visit* is allowed to any customer, then the problem can be transformed into an SD case by an easy modification of the input (see again Gribkovskaia and Laporte 2008). Variants where *multiple visits* are allowed have been studied because they can lead to significant cost reductions. Figure 2 presents three common variants with multiple visits (disregard for the moment Figure 2-(d)). Customer 3 requires only a delivery and 4 only a pickup, whereas the *combined demand customers* (*combined customers* for short in the following) 1 and 2 are characterized by both a delivery and a pickup. In Figure 2-(a) the vehicle adopts backhaul deliveries, by first performing all deliveries (1, 3, and 2), and then all pickups (2, 1, and 4). This problem is an extension to the CD case of the problem illustrated in Figure 1-(a). Figure 2-(b) depicts an example with mixed deliveries, where the vehicle performs delivery and pickup in 1, delivery in 2, delivery in 3, pickup in 2 (not possible during the first visit because of the capacity constraint), and pickup in 4. The problem is known as *single vehicle routing problem with pickups and deliveries* (and also as *general SVPDP*) and extends the one in Figure 1-(b). In all the cases discussed so far the pickup operations are *mandatory*. Figure 2-(c) depicts instead the case of *selective pickups*, where not only the operations can be performed in mixed order, but also pickups are optional and give a revenue if performed. In the example the pickup in 4 is not performed because not profitable enough.




(a) backhauls (b) mixed deliveries

(c) selective pickups (d) selective pickups – extended network

**Figure 2 Some common 1-M-1 SVPDPs combined demand variants (vertex 0 is the depot, hatched boxes are deliveries, and solid boxes are pickups; in (d) combined customers $i$ are duplicated into $\sigma(i)$ and $\pi(i)$).**

In the survey by Gribkovskaia and Laporte (2008) the problem depicted in Figure 2-(c) is denoted as the *single vehicle routing problem with deliveries and selective pickups* (SVRPDSP). SVRPDSP is probably the most studied problem among the 1-M-1 SVPDPs. The first mention of it that we know of is in Golden and Assad (1986), who suggest that the study of this distribution strategy is of interest because it can lead to a lower transportation cost.

It is easy to see that the SD and CD versions of SVRPDSP are interchangeable. On one hand, the CD case is clearly a generalization of the SD case. On the other hand, combined demands can be transformed into single demands by *duplicating* each combined customer into two customers: a delivery and a pickup. In the following we will refer to the network obtained by this duplication as to the *extended network*. An example of such a network is depicted in Figure 2-(d) and is obtained by duplicating the combined customers of Figure 2-(c).

As far as we know, most of the literature on SVRPDSP made use of this duplication (that will be formally described in Section 2) and then focused on the solution of the SD case. Some exceptions,

discussed in details in Section 2.1, are provided by Gribkovskaia et al. (2008) and Nagy and Salhi (2005). The rationale for using the extended network is that in any optimal solution to an SD instance no customer needs to be visited twice, and thus known results from famous problems, including the *traveling salesman problem* (TSP) (see, e.g., Applegate et al. 2006), can be reused. For simplicity, in the following we call this type of solutions *Elementary*. This duplication comes however at a price, because the number of vertices can double, thus obliging the algorithms to deal with much larger problem networks.

In this paper we develop exact algorithms for SVRPDSP and then show how to adapt them to other 1-M-1 SVPDPs, including problems with mandatory pickups. We focus on the CD case, but instead of making use of the extended network, we work on the original problem network. We thus deal with solutions that can be *non-Elementary*, where typically a few customers are visited twice. Consequently, the mathematical models that we developed contain variables that are not restrained to be binary and do not admit classical results from the TSP literature. To deal with this situation we developed innovative algorithms. Computational results show that the developed techniques concretely advance the state-of-the-art on SVRPDSP and that they are able to solve to optimality all known benchmark instances (in all the single vehicle problem variants that we addressed) in reasonable computational times.

In many real-world scenarios the distribution process is performed by a fleet of vehicles instead of a single one. The models and algorithms that we implemented for the single vehicle case can be used to obtain interesting insights also on this larger class of problems. In particular, in the last part of our paper we show how our best algorithm can be quite easily adapted to consider multiple homogeneous capacitated vehicles, and we present extensive results on a new set of instances.

The remainder of the paper is organized as follows. In Section 2 a formal description of SVRPDSP is given and the related literature is presented. In Section 3 a standard formulation looking for Elementary tours in the extended network and based on the use of two-commodity flow variables is introduced. In Section 4 we present a more novel non-Elementary formulation, and show that it

provides an optimal solution to the SD case and a valid relaxation to the CD case. In Section 5 we improve the non-Elementary formulation by means of Benders decomposition and cutting planes. In Section 6 we embed the proposed techniques into a list of exact branch-and-cut algorithms, that are then computationally tested in Section 7 on several problem variants.

### 1.1. Main Contributions

The main contributions of our work are the following ones:

- we propose a non-Elementary formulation making use of two-commodity flow variables and solving the SVRPDSP variant in which split deliveries and intermediate dropoffs are allowed;
- we prove that the formulation provides the optimal solution for the SD case and a tight lower bound for the CD case;
- we obtain an enhanced formulation by a Benders decomposition that projects out the two-commodity flow variables; we further improve the formulation by proposing new families of valid inequalities and exact polynomial time separation procedures;
- we embed everything into exact branch-and-cut algorithms and obtain a substantial breakthrough in the computational results for SVRPDSP: existing algorithms solved to proven optimality only instances with up to 22 combined demand customers (see Gribkovskaia et al. 2008) or 68 single demand customers (see Gutiérrez-Jarpa et al. 2009), whereas our best algorithm solves to optimality with comparable times all benchmarks with up to 100 combined demand customers. We also tested our best algorithm on new larger instances with up to 200 customers, showing that proof of optimality cannot be systematically achieved but small gaps can still be obtained;
- we provide insights that can be useful for practitioners, showing how the main features of SVRPDSP can lead to relevant cost savings;
- we adapt the above techniques to deal with a wide range of 1-M-1 pickup and delivery problems, including the cases of mandatory pickups, split deliveries, intermediate dropoffs, and distribution by multiple vehicles.

## 2. Problem Description

In the *single vehicle routing problem with deliveries and selective pickups* (SVRPDSP) we are given a complete directed graph $G = \{V, A\}$, where $V = \{0, 1, 2, \ldots, n\}$, vertex 0 is the depot, vertices $1, \ldots, n$ are the customers, and $A = \{(i, j) : i, j \in V, i \neq j\}$. For convenience of notation we partition the customer set into three subsets, namely $P$, $D$, and $PD$. Each delivery customer $i \in D$ requires a delivery of weight $d_i$, each pickup customer $i \in P$ offers a pickup of weight $p_i$ and revenue $r_i$, whereas each combined customer $i \in PD$ is associated with both a delivery of weight $d_i$ and a pickup of weight $p_i$ and revenue $r_i$. Each delivery customer must be visited exactly once to perform the delivery, but visits to pickup customers are optional. Each combined customer can be visited either once or twice: if visited twice, then its delivery and pickup are performed separately; if visited once, then either only its delivery is performed, or its delivery and pickup are performed simultaneously. In the following we refer to $G$ as to the *original network* of SVRPDSP.

A single vehicle of weight capacity $Q$, with $Q \geq \sum_{i \in V} d_i$, is based at the depot. A traveling cost $c_{ij}$ is associated with each arc $(i, j) \in A$. We consider the general case in which the cost matrix may be asymmetric and might not satisfy the triangular inequality (note that many instances in the VRP literature do not satisfy this inequality because of the habit of rounding costs to the nearest integer, see, e.g., Toth and Vigo 2002). SVRPDSP is to find a route that starts and ends at the depot, performs all deliveries and possibly some pickups, and minimizes the total cost given by the traveling cost minus the revenues generated by the collected pickups. The problem is strongly NP-hard because generalizes the asymmetric version of TSP, arising as a special SVRPDSP case when $P = PD = \emptyset$.

### 2.1. Literature Review

The first mention of SVRPDSP that we know of is by Golden and Assad (1986), who suggest it as a natural extension of the standard backhaul problem that allows taking full advantage of backhaul economies. Since then, several algorithms and models have been implemented for solving SVRPDSP. Most of these attempts are based on Elementary solutions.

Süral and Bookbinder (2003) focused on the SD case (i.e., $PD = \emptyset$) and suggested the use of the extended network that we discussed in Section 1 to deal with the CD case. They introduced a classification of the SD problems, and then studied SVRPDSP (that they called *single vehicle routing problem with unrestricted backhauls*), motivated by the fact that it is the most general problem of their classification. They proposed a compact *mixed integer linear programming* (MILP) formulation based on the classical constraints by Miller et al. (1960), and enriched it with families of valid inequalities. Their algorithm failed in solving to optimality some instances with just 10 SD customers.

Gribkovskaia et al. (2008) studied the CD case, and focused on instances in which all customers require a delivery (i.e., $P = \emptyset$). They developed a new MILP formulation based on the original network, but obtained by splitting combined customers into two vertices: the first is associated with the first visit in which delivery or both delivery and pickup are performed, the second represents the second visit in which a pickup may be performed. They made use of classical vehicle flow variables among the newly created vertices, plus two additional sets of binary variables, indicating whether, respectively, pickup and delivery were performed simultaneously, and whether pickup was performed during the second visit. As for the case of the extended network, this strategy too comes at the price of requiring a large number of vehicle flow variables. Their MILP was improved with several valid inequalities, nevertheless, only instances with up to 22 CD customers (i.e., 44 SD customers) could be solved to optimality within a reasonable amount of time. In addition, they developed a tabu search heuristic, also working on the extended network, and computationally evaluated it on random instances, finding that the best solutions were frequently non-Elementary. Gutiérrez-Jarpa et al. (2009) addressed the symmetric version of SVRPDSP. Similarly to Süral and Bookbinder (2003) they solved the single demand case and referred to the extended network to deal with combined demands. They proposed a MILP formulation only containing vehicle flow variables, and imposed subtour elimination and capacity restrictions by means of families of exponentially-many constraints. The resulting formulation was solved by a branch-and-cut algorithm that obtained favorable results, solving all instances with up to 68 SD customers.

Several papers addressed SVRPDSP with heuristic algorithms. Here we mention only that the most effective algorithms are, according to our knowledge, the evolutionary algorithm by Bruck et al. (2012), and the variable neighborhood search by Coelho et al. (2012), both working on the extended network and hybridized by the use of MILP models. These heuristics could provide solutions to instances with up to 100 vertices, but obtained quite large optimality gap (See Section EC.3 of the electronic companion for details). Heuristics working on the original problem network were developed by Gribkovskaia et al. (2008) for SVRPDSP and by Nagy and Salhi (2005) for problems with mandatory pickups, and both made use of customer duplications only when necessary.

The use of selective pickups allows us to model situations where the capacity of the vehicle is not enough to perform all operations. In some real-world applications concerning third-party logistics providers pickups can be simply skipped if they are not profitable enough. This is the case, for example, studied by Privé et al. (2006), in which "Distribution Jacques Dubois" delivers full bottles to customers in the Quebec City area and gets a revenue for bringing empty bottles back to the distribution center. In other applications pickups will eventually have to be made. In this case optimized solutions may be obtained by executing SVRPDSP in a *rolling horizon* fashion, see, e.g., Cordeau et al. (2015), possibly increasing the revenues of those pickups that are closer to a certain deadline or have waited too long.

A complete review on the large literature on pickup and delivery routing problems is out of the scope of this paper. For the 1-M-1 SVPDPs we refer the interested reader to the classification and survey given by Gribkovskaia and Laporte (2008). For general pickup and delivery routing problems, we refer to the recent surveys by Battarra et al. (2014) and Doerner and Salazar-González (2014), the former on the transportation of goods and the latter on the transportation of people.

## 3. An Elementary Formulation on the Extended Network

Let us first formally define the extended network as follows.

Definition 1. The *duplication* of a combined customer $i \in PD$ consists in replacing $i$ by two customers, a delivery customer $\sigma(i)$ and a pickup customer $\pi(i)$, having $d_{\sigma(i)} = d_i$, $p_{\sigma(i)} = 0$, $r_{\sigma(i)} = 0$,

$d_{\pi(i)} = 0,\ p_{\pi(i)} = p_i,\ r_{\pi(i)} = r_i,$ and costs $c_{\pi(i),\sigma(i)} = c_{\sigma(i),\pi(i)} = 0,\ c_{\sigma(i),j} = c_{\pi(i),j} = c_{ij}$ and $c_{j,\sigma(i)} = c_{j,\pi(i)} = c_{ji}$ for all $j \in V$.

DEFINITION 2. Given an SVRPDSP instance on a graph $G = (V, A)$, the *extended network* is the graph $G' = \{V', A'\}$ obtained by applying the duplication of Definition 1 to each combined customer $i \in PD$, so obtaining $D' = D \cup \{\sigma(i) : i \in PD\}$, $P' = P \cup \{\pi(i) : i \in PD\}$, $PD' = \emptyset$, $V' = \{0\} \cup P' \cup D'$, and $A' = \{(i, j) : i, j \in V', i \neq j\}$.

On the basis of Definition 2, we can model SVRPDSP using the following formulation looking for Elementary solutions on the extended network. Let $x_{ij}$ be a binary variable indicating whether arc $(i, j) \in A'$ is used or not, and $y_j$ be an additional binary variable taking value 1 if the pickup of customer $j \in P'$ is performed, 0 otherwise. Let also $f^d_{ij}$ and $f^p_{ij}$ be non-negative variables giving, respectively, the total delivery and pickup quantities transported by the vehicle when traveling along arc $(i, j) \in A'$. The resulting *two-commodity Elementary extended* (TCEE) formulation is then:

$$\text{(TCEE)} \quad \min\ z_{TCEE} = \sum_{(i,j)\in A'} c_{ij}x_{ij} + \sum_{j\in P'} r_j(1 - y_j) \tag{1}$$

subject to

$$\sum_{i\in V'\setminus\{j\}} x_{ij} = 1 \qquad \forall\ j \in D' \cup \{0\}, \tag{2}$$

$$\sum_{i\in V'\setminus\{j\}} x_{ij} = y_j \qquad \forall\ j \in P', \tag{3}$$

$$\sum_{i\in V'\setminus\{j\}} (x_{ij} - x_{ji}) = 0 \qquad \forall\ j \in V', \tag{4}$$

$$f^d_{ij} + f^p_{ij} \leq Q x_{ij} \qquad \forall\ (i, j) \in A', \tag{5}$$

$$\sum_{i\in V'\setminus\{j\}} \left(f^d_{ij} - f^d_{ji}\right) = d_j \qquad \forall\ j \in V' \setminus \{0\}, \tag{6}$$

$$\sum_{i\in V'\setminus\{j\}} \left(f^p_{ji} - f^p_{ij}\right) = p_j y_j \qquad \forall\ j \in P', \tag{7}$$

$$\sum_{i\in V'\setminus\{j\}} \left(f^p_{ji} - f^p_{ij}\right) = 0 \qquad \forall\ j \in D', \tag{8}$$

$$x_{ij} \in \{0, 1\} \qquad \forall\ (i, j) \in A', \tag{9}$$

$$y_j \in \{0, 1\} \qquad \forall\ j \in P', \tag{10}$$

$$f^d_{ij}, f^p_{ij} \geq 0 \qquad \forall\ (i,j) \in A'. \tag{11}$$

Constraints (2) impose that each delivery vertex is visited once. Constraints (3) tie together the $x$ with the $y$ variables, whereas (5) connect the $x$ with the $f^d$ and $f^p$ variables and ensure that the vehicle capacity is not exceeded. Constraints (4) and (6) impose flow conservation for the vehicle and the delivery quantity, respectively. Similarly, constraints (7) and (8) impose flow conservation of the pickup quantity for the vertices in $P'$ and $D'$, respectively.

The objective function (1) minimizes the sum of the costs plus the sum of the lost revenues of those pickups that were not performed. We found it convenient to adopt this function because it always takes non-negative values, thus facilitating the evaluation of the optimality gaps. Minimizing (1) is equivalent to minimizing the original SVRPDSP objective (traveling cost minus revenues generated by the collected pickups), because the two functions just differ by the fixed term $\sum_{j \in P'} r_j$. Moreover (1) is widely used in the related *traveling salesman problem with profits* (TSP-P), also known as the prize-collecting traveling salesman problem, see, e.g., Feillet et al. (2001). Also note that one could impose initial values for the pickup and delivery quantities leaving the depot, by setting $\sum_{i \in V'} f^d_{0i} = \sum_{i \in D'} d_i$ and $\sum_{i \in V'} f^p_{0i} = 0$. Alternatively, formulation TCEE may return an excess of these quantities (we have $\sum_{i \in D'} d_i \leq \sum_{i \in V'} f^d_{0i} \leq Q$, so when $\sum_{i \in D'} d_i < Q$ an excess of delivery quantity may be loaded on the vehicle and transported all along the route; a similar reasoning applies to $f^p_{0i}$), but these can be easily removed by inspection.

## 4. A Relaxation Based on a Non-Elementary Formulation

Let us consider the original SVRPDSP network $G = \{V, A\}$ described in Section 2. Similarly to what was done for formulation TCEE, let $y_j$ be a binary variable taking value 1 if the pickup of customer $i \in P \cup PD$ is performed, 0 otherwise, and let $f^d_{ij}$ and $f^p_{ij}$ be non-negative variables giving, respectively, total delivery and pickup quantities transported along arc $(i,j) \in A$. Let also $x_{ij}$ be an integer variable indicating the *number of times* in which arc $(i,j) \in A$ has been traveled. For feasibility, $x_{ij}$ can take value 2 only when both $i$ and $j$ are in $PD$, but cannot be greater than

1 otherwise. By defining $A(PD) = \{(i,j) \in A : i,j \in PD\}$, we obtain the following *two-commodity non-Elementary* (TCNE) formulation:

$$\text{(TCNE)} \quad \min\ z_{TCNE} = \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{j\in P\cup PD} r_j(1-y_j) \tag{12}$$

subject to

$$\sum_{i\in V\setminus\{j\}} x_{ij} = 1 \qquad \forall\ j \in D\cup\{0\}, \tag{13}$$

$$\sum_{i\in V\setminus\{j\}} x_{ij} = y_j \qquad \forall\ j \in P, \tag{14}$$

$$\sum_{i\in V\setminus\{j\}} x_{ij} \geq 1 \qquad \forall\ j \in PD, \tag{15}$$

$$\sum_{i\in V\setminus\{j\}} x_{ij} \leq y_j + 1 \qquad \forall\ j \in PD, \tag{16}$$

$$\sum_{i\in V\setminus\{j\}} (x_{ij} - x_{ji}) = 0 \qquad \forall\ j \in V, \tag{17}$$

$$f^d_{ij} + f^p_{ij} \leq Qx_{ij} \qquad \forall\ (i,j) \in A, \qquad (t_{ij}) \tag{18}$$

$$\sum_{i\in V\setminus\{j\}} \left(f^d_{ij} - f^d_{ji}\right) = d_j \qquad \forall\ j \in V\setminus\{0\}, \qquad (v_j) \tag{19}$$

$$\sum_{i\in V\setminus\{j\}} \left(f^p_{ji} - f^p_{ij}\right) = p_j y_j \qquad \forall\ j \in P\cup PD, \qquad (w_j) \tag{20}$$

$$\sum_{i\in V\setminus\{j\}} \left(f^p_{ji} - f^p_{ij}\right) = 0 \qquad \forall\ j \in D, \qquad (w_j) \tag{21}$$

$$x_{ij} \in \{0,1\} \qquad \forall\ (i,j) \in A\setminus A(PD), \tag{22}$$

$$x_{ij} \in \{0,1,2\} \qquad \forall\ (i,j) \in A(PD), \tag{23}$$

$$y_j \in \{0,1\} \qquad \forall\ j \in P\cup PD, \tag{24}$$

$$f^d_{ij}, f^p_{ij} \geq 0 \qquad \forall\ (i,j) \in A. \tag{25}$$

The objective function (12) minimizes the sum of the costs and of the lost revenues. Constraints (13) force the delivery vertices and the depot to be visited once, whereas (15) force the combined vertices to be visited at least once. Constraints (16) are necessary to impose one visit in case the pickup is not performed, or up to two visits in case the pickup is performed. Constraints (14) tie together the $x$ with the $y$ variables for the pickup vertices. Constraints (17) impose flow conservation for the vehicle. Note that in (16) and (17) the sum of the variables associated with the

incoming arcs on $j$ may take value 0, 1, or even 2 when $j \in PD$. Constraints (18) and (19) impose the vehicle capacity restriction and the flow conservation for the delivery quantities, respectively. Similarly, constraints (20) and (21) set the flow conservation for the pickup quantities. Next to constraints (18)-(21) we provide the dual variables that are used later in Section 5. Note that in (18) the total quantity traveling along an arc in $A(PD)$ may take value up to $2Q$. An example of a TCNE solution in which an arc is traveled twice is given in Section EC.2.1 of the electronic companion.

Formulation TCNE has some interesting properties.

**Property 1** *For the SD case formulation TCNE is equivalent to formulation TCEE and thus provides the optimal SVRPDSP solution value.*

This can be trivially checked by removing in TCNE variables and constraints associated with $PD$.

**Property 2** *For the CD case formulation TCNE provides a relaxation of SVRPDSP.*

*Proof* Given in Section EC.1.1 of the electronic companion to this paper.

A consequence of Property 2 is that in the general CD case the optimal solution value of formulation TCNE is a valid lower bound on the optimal SVRPDSP objective value, i.e., $z_{TCNE} \leq z_{TCEE}$. To see that there can be cases in which this lower bound is not tight, we discuss two relevant situations in the next two sections.

### 4.1. Solutions with Split Deliveries or Pickups

Using a common definition in the routing literature, we say that a delivery is *split* if it is partially performed during the first visit to the customer and then completed during the second visit. The same definition applies to pickups. A solution with *split deliveries or pickups* is a solution in which one or more deliveries and/or one or more pickups are split.

This is a characteristic that can be observed in optimal TCNE solutions to CD instances. Refer for example to Figure 3. The figure reports for each vertex $j$ the delivery and pickup quantities in square brackets, i.e., $[d_j, p_j]$, and for each arc $(i, j)$ the delivery and pickup flows in round brackets,

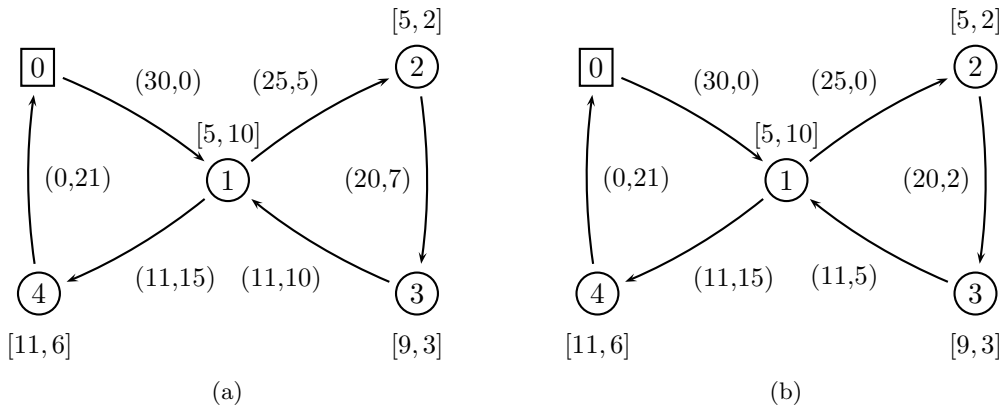



**Figure 3** **Split deliveries or pickups: (a) a feasible TCNE solution with a split pickup in vertex 1; (b) a same-cost solution without split that is feasible for SVRPDSP.**

i.e., $(f_{ij}^d, f_{ij}^p)$. The vehicle capacity is $Q = 30$. Figure 3-(a) shows an example in which the vehicle leaves the depot and performs a first visit to vertex 1 delivering entirely the quantity $d_1 = 5$ and picking up only 5 units of the quantity $p_1 = 10$. The remaining 5 units are picked up in the second visit to the vertex, and the resulting distribution would not invalidate the TCNE constraints.

Split deliveries or pickups are not a major issue, because, as noted by a few authors in related 1-M-1 SVPDPs (see, e.g., Gribkovskaia et al. 2007, 2008), it is never suboptimal to perform the entire delivery first (and thus the entire pickup, if any, after the delivery). Refer for example to Figure 3-(b), that depicts a solution obtained from that of Figure 3-(a) by modifying the loads on the arcs. During the first visit to vertex 1 the vehicle delivers the entire quantity $d_1$ but performs no pickup, and then during the second visit it performs the entire pickup $p_1$. This new solution has the same cost of the previous one, but it is now feasible for SVRPDSP. Formally, we can state the following result.

**Property 3** *A TCNE solution with split deliveries or pickups can be transformed into a solution having the same cost and for which no delivery or pickup is split.*

*Proof* Given in Section EC.1.2 of the electronic companion to this paper.

Given a solution with split deliveries or pickups, the flows $f_{ij}^d$ and $f_{ij}^p$ on the arcs that ensure to have a solution without splits may be found by making use of a simple recursive algorithm, called

REMOVESPLIT, that we describe in detail in Section EC.2.2 of the electronic companion. We just remark here that this algorithm has a complexity that grows exponentially with the number of vertices visited twice, because each such vertex creates two different paths.

### 4.2. Solutions with Intermediate Dropoffs

Using another common definition in the routing literature, we say that a *dropoff* at a vertex $i$ occurs if the vehicle drops part of its load in $i$ during its first visit, and recollects it during its second visit. An example is depicted in Figure 4, where the notation adopted is the same as the one in Figure 3 and the vehicle capacity is again 30. Figure 4-(a) shows an example in which the vehicle leaves the depot with 30 units of load and visits vertex 1, performing the delivery $d_1 = 5$ but also dropping off 2 units of load. This allows performing completely the pickup at vertex 2, without exceeding the vehicle capacity on the following arc $(2,3)$. The two units are then re-loaded on the vehicle during the second visit to 1.

For the case of dropoffs, unfortunately, there is no equivalent of the nice Property 3 that is valid for the case of split deliveries. The reason for this can be intuitively seen from Figure 4-(b): if the load is not dropped off at node 1 and the pickup of vertex 2 is still performed, then the capacity of the vehicle is exceeded on arc $(2,3)$. We thus take care of the dropoffs by devising a few tailored algorithms, described in Section 6. These algorithms eventually need to check whether or not a solution has dropoffs. This is performed by a procedure called CHECKDROPOFF and presented in

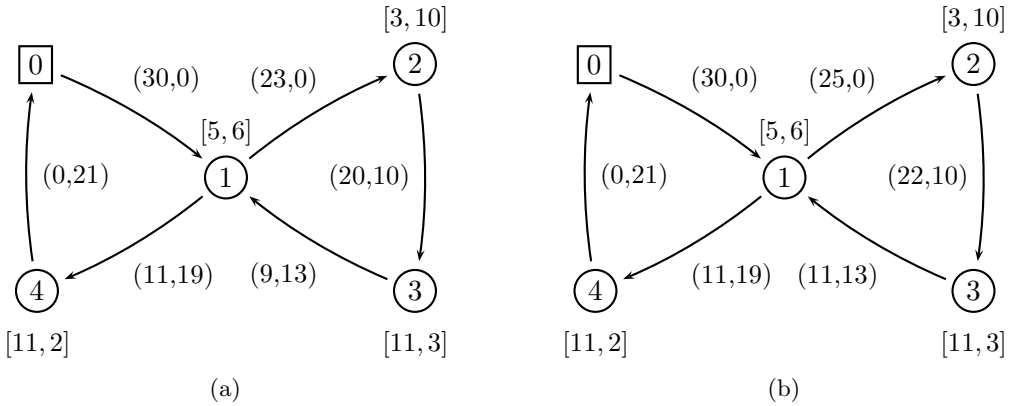



**Figure 4** **Intermediate dropoffs: (a) two units of load are temporarily dropped off at vertex 1; (b) the solution without dropoff violates capacity $Q = 30$ on arc (2,3).**

Section EC.2.3 of the electronic companion. This procedure is based on solving a modified version of formulation TCEE in which we include additional information derived from the Non-Elementary solution at hand.

We conclude this section by noticing the following fact.

**Property 4** *Let SVRPDSP-D be the relaxation of the SVRPDSP in which intermediate dropoffs of the load are allowed at the vertices. An optimal solution to SVRPDSP-D may be obtained by solving formulation TCNE and then executing procedure* REMOVESPLIT *on the resulting solution.*

We believe this statement is of some interest, because, as noted by Gribkovskaia and Laporte (2008), intermediate dropoffs have never been studied in the area of 1-M-1 SVPDPs, so, as far as we know, Section 7 presents the first computational results for this research area. Note that SVRPDSP-D does not allow split deliveries, and that is why REMOVESPLIT is needed in Property 4.

## 5. A Faster Relaxation by Cutting Plane Generation

An inconvenience of formulation TCNE is that the presence of the $f^d$ and $f^p$ continuous variables may slow down the solution process for large size instances. To obtain an equivalent but faster formulation, we make use of the classical decomposition by Benders (1962). We project out the $f^d$ and $f^p$ variables from the model, as well as the constraints involving those variables. The remaining *master problem* (MP) contains the "complicating" variables $x$ and $y$, and requires to minimize (12) subject to (13)–(17) and (22)–(24). Once a feasible solution to MP is found, its values are inserted into the *primal subproblem* (PSP). PSP is a linear problem on the "easy" $f^d$ and $f^p$ variables involving a null objective function and the constraints (18)–(21) and (25).

To solve PSP we make use of duality theory. We associate variables $t_{ij}$ with constraints (18) for all $(i,j) \in A$, $v_j$ with (19) for all $j \in V \setminus \{0\}$, $w_j$ with (20) for all $j \in P \cup PD$ and with (21) for all $j \in D$. We then solve the resulting *dual subproblem* (DSP). The solution of DSP may be either feasible or unbounded. If it is feasible, then it is equal to 0 because of strong duality, so it does not impact the solution cost of MP (in other words, there are no *Benders optimality cuts* because the

costs of the $f^d_{ij}$ and $f^p_{ij}$ variables are 0). If it is unbounded, then let $t^r_{ij}$, $v^r_j$, and $w^r_j$ be the values of an extreme ray $r$ of DSP. We multiply the original constraints by the values of the extreme ray and add the corresponding *Benders feasibility cut* to MP. Let $R$ define the set of the extreme rays of DSP. The resulting *Benders-based non-Elementary* (BBNE) formulation is thus:

$$\text{(BBNE)} \quad \min\ z_{BBNE} = \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{j\in P\cup PD} r_j(1-y_j)$$

subject to (13)–(17), (22)–(24) and

$$\sum_{(i,j)\in A} (Qt^r_{ij})x_{ij} + \sum_{j\in V\setminus\{0\}} d_j v^r_j + \sum_{j\in P\cup PD} (p_j w^r_j)y_j \le 0 \qquad \forall r \in R. \tag{26}$$

Constraints (26) are the above mentioned Benders feasibility cuts, and forbid solutions that are infeasible with respect to the removed $f^d$ and $f^p$ variables. The detailed PSP and DSP models are given in Section EC.2.4 of the electronic companion.

Benders feasibility cuts are known to be weak in practice, and our computational tests confirmed this behavior. We could not find improvements of (26) based, for example, on special constraint structures (as found for other pickup and delivery problems, see, e.g., Hernández-Pérez and Salazar-González 2004) or on the so-called combinatorial Benders cuts (see, e.g., Codato and Fischetti 2006 and Côté et al. 2014). Thus, we looked for additional valid inequalities that could improve the computational behavior of formulation BBNE.

### 5.1. Valid Inequalities

In this section we present a few families of valid inequalities that strengthen the continuous relaxation of formulation BBNE and improve its computational performance. In the following, for any set $S \subseteq V \setminus \{0\}$, let $d(S) = \sum_{j\in S} d_j$, $p(S) = \sum_{j\in S} p_j$, and $\bar{S} = V \setminus S \setminus \{0\}$. Let also $x(\bar{S} : S) = \sum_{i\in\bar{S}} \sum_{j\in S} x_{ij}$, $x(A(S)) = \sum_{i\in S} \sum_{j\in S} x_{ij}$, and $x(\delta^+(S)) = \sum_{i\in S} \sum_{j\in V\setminus S} x_{ij}$.

First, we impose *subtour elimination constraints* (SECs) as follows:

$$x(A(S)) \le |S| - 1 \qquad \forall S \subseteq P, \tag{27}$$

$$x(\delta^+(S)) \ge 1 \qquad \forall S \subseteq D \cup PD. \tag{28}$$

Constraints (27) require that at most $|S|-1$ arcs are selected in a set $S$ containing only pickups, whereas constraints (28) impose that at least one arc should be selected among those leaving a

set $S$ containing at least a delivery but not the depot. It is worth noting that in Elementary formulations it is easy to transform a SEC taking the "$\leq$" form into one with "$\geq$" form because of degree constraints. However, this is not true for non-Elementary formulations and it is easy to find examples where: (28) is not valid for a set $S \subseteq P$ (think of the case in which no pickup is selected in $S$); and (27) is not valid for a set $S \subseteq D \cup PD$ (think of the case in which one or more vertices in $S$ are visited twice).

To obtain SECs involving both sets $P$ and $D$, we adapt an inequality originally proposed for the symmetric version of SVRPDSP by Gutiérrez-Jarpa et al. (2009), namely:

$$x(A(S)) \leq |S_D| + \sum_{j \in S_P} y_j - 1 \qquad \forall S = S_D \cup S_P : S_D \subseteq D, S_D \neq \emptyset, S_P \subseteq P, \tag{29}$$

where $S$ is partitioned into a subset $S_D$ of delivery customers and a subset $S_P$ of pickup customers. We notice that (29) requires $S_D$ to be non-empty, because otherwise the right hand side of the inequality could be negative. We also notice that one could extend the set $S$ by including a subset $S_{PD} \subseteq PD$ of combined customers. However the right hand side of (29) would result in $|S_D + S_{PD}| + \sum_{j \in S_P \cup S_{PD}} y_j - 1$, which is very weak and was consequently not used in our tests.

Wolsey (1998) developed a family of constraints to remove subtours in the related context of TSP-P. The asymmetric version of these constraints directly applies to SVRPDSP as follows:

$$x(A(S)) \leq \sum_{j \in S \setminus \{k\}} y_j \qquad \forall S \subseteq P, k \in S. \tag{30}$$

The number of constraints (30) is very large, and even including them in an iterative way, using a branch-and-cut scheme, results in a poor convergence of our models. As done by a number of authors (see, e.g., Laporte and Martello 1990 for TSP-P, and Gutiérrez-Jarpa et al. 2009 for the symmetric version of SVRPDSP) we found it convenient to aggregate these constraints, obtaining the following result.

**Property 5** *The following inequality is valid for SVRPDSP:*

$$x(\delta^+(S)) \geq \sum_{j \in S} \frac{1}{|S|} y_j \qquad \forall S \subseteq P. \tag{31}$$

*Proof* Given in Section EC.1.3 of the electronic companion to this paper.

A similar valid inequality, that we could not find in the related literature, is the following one.

**Property 6** *The following inequality is valid for SVRPDSP:*

$$x(\delta^+(S)) \geq \sum_{j \in S} \frac{p_j}{Q} y_j \qquad \forall S \subseteq P. \tag{32}$$

*Proof* Given in Section EC.1.4 of the electronic companion to this paper.

In addition to (32), we found other valid inequalities that take into consideration the vehicle capacity. First of all, notice that the vehicle leaves the depot with a residual capacity equal to $Q - d(V)$. Consequently, it cannot visit directly a set $S$ to perform a series of pickups and deliveries that could exceed this residual capacity. This consideration leads to the following result.

**Property 7** *The following inequality (capacity-cut constraint) is valid for SVRPDSP.*

$$Qx(\bar{S} : S) - \sum_{j \in S} p_j y_j \geq d(V) - d(S) - Q \qquad \forall S \subseteq V \setminus \{0\} : p(S) - d(S) > Q - d(V). \tag{33}$$

*Proof* Given in Section EC.1.5 of the electronic companion to this paper.

For the sake of clarity, we notice that there are cases in which a solution may be infeasible for the Benders feasibility cuts (26), but feasible for the capacity-cut constraints (33).

We finally make use of the two following simple inequalities.

$$\sum_{j \in P \cup PD} p_j y_j \leq Q, \tag{34}$$

$$\sum_{i \in V \setminus \{0,j\}} x_{ij} \geq y_j \qquad \forall j \in V \setminus \{0\} : p_j - d_j > Q - d(V). \tag{35}$$

Constraint (34) is a classical knapsack constraint, whereas constraint (35) is an improvement of (33) that can be used when $S$ consists of a single vertex $j$ for which $p_j - d_j$ is greater than the residual capacity of the vehicle when leaving the depot. The resulting *two-index non-Elementary* (TINE) formulation is thus:

$$\text{(TINE)} \quad \min \sum_{(i,j) \in A} c_{ij} x_{ij} + \sum_{j \in P \cup PD} r_j (1 - y_j)$$

$$\text{subject to} \quad (13)\text{–}(17),\ (22)\text{–}(24),\ \text{and}\ (26)\text{–}(35).$$

### 5.2. Separation Procedures

Because most of the families of inequalities that we presented in the previous subsection have exponential size, we found it convenient to add them on the fly in a *branch-and-cut* (B&C) fashion (see, e.g., Padberg and Rinaldi 1991). We thus developed tailored separation procedures, that identify those inequalities that are violated by a given solution, so as to add them to the model.

The separation of (27) and (28) can be done by adapting to the SVRPDSP the classical max-flow algorithm for the TSP (see, e.g., Applegate et al. 2006) as follows. Given a (fractional) solution $(\bar{x}, \bar{y})$, we create a *supporting graph* $\bar{G} = \left(\bar{V}, \bar{A}\right)$, where $\bar{V} = V \setminus \{i \in P : \bar{y}_i = 0\}$ and an arc $(i,j) \in \bar{A}$ has capacity equal to $\bar{x}_{ij}$. We then solve a max-flow problem on $\bar{G}$, from 0 to each $i \in P$ having $\bar{y}_i > 0$ to separate (27), and from 0 to each $i \in D \cup PD$ for (28). The set $\bar{S}$ identified by the min cut and not containing 0 is then checked for detecting a possible violation.

The symmetric version of (29) was separated by an enumerative breadth-first-search procedure in Gutiérrez-Jarpa et al. (2009), but we opted for a standard max-flow based procedure. The separation of (30) may be performed by adapting to the asymmetric case the procedure described by Wolsey (1998). As discussed in the previous subsection, in our implementation instead of (30) we preferred to use the aggregate constraints (31) and (32). To separate these constraints we adopted a quick check: after having determined during the separation of (27) the set $\bar{S}$ leading to the minimum cut, we check whether $\bar{S}$ also violates (31) or (32). If this is true, then we add the corresponding cut (if both (31) and (32) are violated, then we only add the most violated cut).

For the new capacity-cut constraints that we introduced, we can state the following result.

**Property 8** *Constraints* (33) *can be separated in polynomial time.*

*Proof* Given in Section EC.1.6 of the electronic companion to this paper.

Constraint (34) can be improved by using the well known *extended cover inequalities* (ECIs) (see, e.g., Van Roy and Wolsey 1987). In our implementation we looked for ECIs in a heuristic way, using the fast algorithm defined in Section 3.3 of Kaparis and Letchford (2010). There are $O(n)$ constraints in (35), so we directly add all of them to the model.

## 6. Exact Algorithms

Exact algorithms based on formulations where multiple traversals are allowed on arcs have been recently used to solve various routing problems, showing that good results can be obtained in practice. The contributions that we are aware of use these formulations mostly to provide a valid lower bound (see, e.g., Chemla et al. 2013 and Irnich et al. 2015), and just in some cases to attempt to obtain a feasible solution. In this section we extend this idea by showing a few ways in which a non-Elementary formulation can be used iteratively to quickly converge to an optimal solution.

In classical B&C algorithms from the literature the separation procedures are invoked at any node of the enumeration tree, so as to lift as soon as possible the lower bound and fathom the highest amount of nodes. In recent years, however, important advances achieved by commercial MILP solvers made other options computationally effective. These options include the separation of cuts only at integer points using the so-called *lazy callback* (see, e.g., Subramanian et al. 2011 for a related SVPDP), and the old-fashioned cutting plane generation method that solves the MILP to integer optimality before adding cuts (see, e.g., the work on TSP by Pferschy and Staněk 2016). We consequently attempted several policies for our algorithms, first for solving the SD case with formulation TINE, and then with the aim of using TINE as a basis to optimally solve the CD case.

For the solution of the SD case with TINE, we found it convenient to use a B&C (defined TINE B&C in the following) that separates inequalities only at integer points, according to the following order: (27) (possibly obtaining violated cuts also for (31) and (32), as mentioned in Section 5.2) and (28); (33); the ECIs associated with (34); (29) (only for instances having one or more single demand customers); and finally (26). The separation of a family of cuts is invoked only if the previous separations failed in providing violated cuts. If more cuts of the same family are found, all of them are added to the model. We also included a simple heuristic algorithm, based on the one proposed by Coelho et al. (2012), to compute a valid upper bound. This algorithm finds a first feasible solution by solving a TSP on the vertices in $V \setminus P$. It then solves a knapsack problem using the pickup demands in $PD \cup P$ as items, and setting the capacity of the knapsack to $Q$. The

pickups selected in the knapsack solution are inserted in greedy way in the TSP tour, one at a time in the lowest-cost position in which they fit, if any.

Note that separation at integer points may be performed by a graph-search algorithm that looks for a path invalidating a given constraint. We implemented a few simple procedures of this kind to separate the above constraints, but we skip their detailed description because they are slight variations of Algorithm 1 in Section EC.2.2 . We only mention the fact that, as previously noted in Section 4.1, the number of possible paths in a non-Elementary solution increases exponentially with the number of customers visited twice. Thus, if more than six customers are visited twice, we skip the graph-search algorithms and invoke the max-flow based procedures of Section 5.2.

Concerning instead the solution of the CD case, we attempted the following strategies, all based on TINE B&C that we have just described for the SD case.

***Throw Away B&C* (TA)**. TA includes a simple check in TINE B&C: for each incumbent solution found during the search process, TA performs a feasibility test to check whether the solution is feasible for SVRPDSP or requires dropoffs. The test is performed by using procedure CHECK-DROPOFF (see Section EC.2.3 ). Infeasible solutions are simply not used by TA, which continues exploring the enumeration tree until an optimal SVRPDSP solution is found.

***2-Step B&C* (2S)**. During preliminary experiments we found out that TINE B&C is considerably faster than TA, and for several instances the solution it yields is also optimal for SVRPDSP. 2S takes advantage of this observation in a two-step procedure. In the first step it solves TINE B&C, but keeps in memory all the generated cuts along with the best solution that is feasible for SVRPDSP, if any. If the solution found at the end of this step has no dropoffs, then it is optimal for SVRPDSP and 2S terminates. Otherwise, 2S proceeds to the second step where it invokes TA. Additionally 2S improves the convergence of TA by initializing it with the best feasible solution found and all the violated cuts generated during the execution of the first step.

***Minimal Extended Network B&C* (MEN)**. MEN works on the basis that solving the extended network (recall Definition 2) always results in a solution with no dropoffs. However, the

idea behind MEN is to duplicate as few customers as possible, instead of duplicating all of them as in the classical Elementary formulations. The algorithm starts as in the first step of 2S by solving TINE B&C and keeping in memory all generated cuts and the best SVRPDSP feasible solution, if any. In case the optimal solution found has no dropoffs, then MEN terminates with an optimal SVRPDSP solution. Otherwise, the solution is inspected to discover which customers have been visited twice. All such customers are duplicated using the procedure in Definition 1. By doing so, we guarantee that the existing dropoffs are removed from the next optimal solution. The next iteration indeed solves the model by considering the updated graph, all the cuts generated at the previous iterations, and the best SVRPDSP feasible solution found so far, if any. The procedure iterates until the optimal solution found has no dropoffs. The procedure CheckDropoff used to detect the dropoffs can be time consuming when the number of customers visited twice is large. As speed-up technique, we decided to inspect on the fly only solutions that have less than 10 customer visited twice, and insert the other solutions in a list that is kept sorted by non-increasing cost. At the end of MEN, we select the first entry in the list, if any, and detect if it has dropoffs. We iterate until either a solution with no dropoff is found or the list is empty, thus providing the optimal solution. As mentioned in Section 5.1, a solution might be infeasible for the Benders feasibility cuts (26) but feasible for the capacity-cut constraints (33). Given that MEN is able to catch these infeasibilities by simply duplicating customers, we obtained a small computational improvement by removing the Benders cuts from the version of TINE B&C used inside MEN.

## 7. Computational Results

We implemented our algorithms on C++ and ran our tests on a PC with an Intel Core i7-3770 3.40 GHz with 8 Gb of RAM. We used Cplex 12.5.1 to solve the MILPs and implement the B&C algorithms. We selected the default Cplex options, but imposed it to use a single thread to facilitate computational comparison with the literature. Unless stated otherwise, each algorithm was allowed a time limit of one CPU hour for each run. We tested our algorithms on the benchmark instances

from the literature and on newly created large size and multiple vehicle instances. Because this amounts to a fairly large number of tests, we provide here only aggregate results, and refer to Section EC.4 for detailed results.

### 7.1. Results for SVRPDSP

Table 1 presents the results for the SD case, for which two benchmark sets are available. The first one (called SB set in the following) was proposed by Süral and Bookbinder (2003) and contains 63 instances, among which are 24 with 10 customers, 21 with 20 customers, and 18 with 30 customers. The second one (GMO set) was proposed by Gutiérrez-Jarpa et al. (2009) and has 74 instances that we divided into 18 small size, 39 medium size, and 17 large size instances, as shown in Table 1. Formulation TCNE yields an optimal solution for the SD case, as stated in Property 1. The same holds for BBNE and TINE, as they are equivalent to TCNE (see Section 5). We thus tested these three formulations and compared them with the exact algorithms available in the literature. Namely, the mathematical model by Süral and Bookbinder (2003) (SB), that we re-implemented and ran on our computer with one CPU hour of time limit, and the B&C by Gutiérrez-Jarpa et al. (2009) (GMO), that was run on a Dual Core AMD 2.7 GHz, with 21000 seconds of time limit. Formulation TINE was solved by TINE B&C of Section 6. For each algorithm and group of instances we provide in column "opt" the total number of proven optimal solutions, and in column "sec" the average CPU seconds elapsed (considering the time limit value for those instances that were unsolved to proven optimality). The best opt values are in bold. For each group of instances, line "totals" reports the total numbers of instances (#) and opt, and line "averages" the average values for sec.

Table 1 shows that the SB set is very easy. Formulation SB is quite weak as it solves to proven optimality only 59 instances in this set, whereas all other algorithms solve all instances. The fastest algorithms are GMO and TINE, as they need just fractions of a second. On the GMO set, which is more challenging, algorithm GMO again outperforms SB, but in this case it is not able to solve to proven optimality one instance with 90 customers and requires almost the entire time limit to

**Table 1 Computational results on the single demand case (sec gives averages across all instances in the row, considering t.lim. for instances unsolved to proven optimality).**

| | | | literature | | | | new algorithms | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | SB | | GMO* | | TCNE | | BBNE | | TINE | |
| set | size | # | opt | sec | opt | sec | opt | sec | opt | sec | opt | sec |
| SB | $n$=10 | 24 | **24** | 0.5 | **24** | 0.0 | **24** | 0.1 | **24** | 0.0 | **24** | 0.0 |
| | $n$=20 | 21 | 18 | 516.3 | **21** | 0.2 | **21** | 0.5 | **21** | 0.5 | **21** | 0.0 |
| | $n$=30 | 18 | 17 | 621.2 | **18** | 0.6 | **18** | 2.3 | **18** | 3.1 | **18** | 0.1 |
| | totals | 63 | 59 | | **63** | | **63** | | **63** | | **63** | |
| | averages | | | 349.8 | | 0.2 | | 0.9 | | 1.1 | | 0.0 |
| GMO | $25 \leq n \leq 30$ | 18 | **18** | 13.8 | **18** | 2.2 | **18** | 1.5 | **18** | 4.8 | **18** | 0.1 |
| | $38 \leq n \leq 60$ | 39 | 26 | 1691.9 | **39** | 47.0 | **39** | 44.1 | 36 | 466.4 | **39** | 3.6 |
| | $68 \leq n \leq 90$ | 17 | 2 | 3375.9 | 16 | 3510.8 | **17** | 314.6 | 6 | 2511.6 | **17** | 37.3 |
| | totals | 74 | 46 | | 73 | | **74** | | 60 | | **74** | |
| | averages | | | 1670.6 | | 831.9 | | 95.9 | | 824.0 | | 10.5 |

(∗ = run with with Cplex 10 on a Dual Core AMD 2.7 GHz, with 21000 sec time limit)

solve another instance also with 90 customers. On average, formulation TCNE is better than GMO on this set. Formulation BBNE is worse than TCNE, showing that a straight application of the Benders decomposition is not enough to obtain improvements over the two-commodity formulation. TINE B&C is the fastest algorithm, as it can solve all instances in the GMO set in about 10 seconds on average, and never requiring more than 150 seconds. The only unsolved instance in Gutiérrez-Jarpa et al. (2009) was solved to proven optimality by TINE in just 143 seconds.

Table 2 gives the results on the CD case. Here we recall that the solutions of our non-Elementary formulations may have dropoffs, so we use the exact algorithms TA, 2S, and MEN of Section 6. Gribkovskaia et al. (2008) introduced the only benchmark set for the CD case (GLS set), that contains 68 instances having between 15 and 100 customers, all requiring a combined demand. We tested the TA, 2S, and MEN algorithms, and compared them with the SB model, the mathematical model by Gribkovskaia et al. (2008) (GLS) that we re-implemented and ran on our PC for one CPU hour, and formulation TCEE of Section 3. The breakthrough introduced by the new non-Elementary formulations is evident. Among the Elementary formulations, GLS can solve just one instance to proven optimality, SB 5, and TCEE 24. It is worth noting that formulations SB and

**Table 2 Computational results on the combined demand case (sec gives averages across all instances in the row, considering t.lim. for instances unsolved to proven optimality).**

| | | Elementary | | | | | | non-Elementary | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | GLS | | SB | | TCEE | | TA | | 2S | | MEN | |
| GLS set | # | opt | sec | opt | sec | opt | sec | opt | sec | opt | sec | opt | sec |
| $15 \leq n \leq 30$ | 28 | 1 | 3486.5 | 5 | 3056.8 | 23 | 914.9 | **28** | 1.7 | **28** | 0.6 | **28** | 0.2 |
| $32 \leq n \leq 50$ | 24 | 0 | t.lim. | 0 | t.lim. | 1 | 3591.9 | 18 | 916.2 | 18 | 912.0 | **24** | 6.5 |
| $71 \leq n \leq 100$ | 16 | 0 | 2903.7 | 0 | t.lim. | 0 | t.lim. | 8 | 2023.7 | 10 | 1386.1 | *__15__ | 416.6 |
| totals | 68 | 1 | | 5 | | 24 | | 54 | | 56 | | **67** | |
| averages | | | 3389.5 | | 3376.3 | | 2491.5 | | 800.2 | | 648.3 | | 100.4 |

(∗ = remaining instance solved to proven optimality by MEN in 17172 seconds)

GLS (and the one by Gribkovskaia et al. 2007 discussed in Section 7.3) were proposed mainly with the purpose of unambiguously describing the problem, so, even with the inclusion of several valid inequalities, their bad performance is not surprising. Notably, the Elementary formulations solve to optimality just one medium size instance and no large size one. This may be imputed to the increase in the size of the underlying extended network. The algorithms based on non-Elementary formulations have a better performance, with TA being able to solve 54 instances, 2S 56, and MEN 67. MEN is the most efficient algorithm, so we ran it with a larger CPU time limit on the only unsolved instance, and could find a proven optimal solution in 17172 seconds (about 4.7 CPU hours).

We also compared the results of our best exact algorithm with the most efficient metaheuristic algorithms from the literature. From the comparison, presented in Section EC.3 of the electronic companion, we can clearly see that MEN greatly outperforms the metaheuristics in terms of number of optimal solutions found, time consumption, and optimality gaps. This is further evidence of its efficiency in solving practical problems.

### 7.2. Evaluation of the intermediate dropoff relaxation

The main ingredient for the good behavior of the non-Elementary algorithms is the solution of SVRPDSP-D (i.e., the SVRPDSP relaxation allowing dropoffs). In Table 3 we report some more information that allows us to get some insight in the quality of this relaxation. The table reports

aggregate results for the GLS set, by running TINE B&C to solve SVRPDSP-D and MEN to solve SVRPDSP. The table provides the following information for both algorithms: average number of seconds elapsed (sec); average percentage gaps between the optimal solution value and the lower bounds at the root node before and after adding the cuts of Section 5.1 ($\text{gap}_r$ and $\text{gap}_c$, respectively); and average number of nodes explored by the branching tree (nodes). Moreover, for TINE we provide the number of optimal SVRPDSP-D solutions that were also optimal for SVRPDSP (feas), and the average percentage gap between the optimal SVRPDSP-D and SVRPDSP solution values ($\text{gap}_d$). For MEN we also provide the average numbers of customers that were duplicated (dupl) and of iterations that were performed (iter) by the main loop of the algorithm.

The cuts are very effective, as they manage to reduce by about nine times the root node gap on both problems variants. For the small size instances, 26 out of 28 optimal SVRPDSP-D solutions do not use dropoffs, and the remaining 2 can be made feasible for SVRPDSP with just two iterations of MEN. For the medium size instances, SVRPDSP is a bit more challenging than SVRPDSP-D and the root node gap after the cuts increases by about 0.1%, but MEN still needs just two iterations in the worst case to close an instance. For the large size instances the difference between the two problems is larger, as confirmed by the value of $\text{gap}_d$ that raises to 0.05%. Indeed SVRPDSP-D is easier as all instances are solved in less than 6 minutes on average by TINE.

One may argue that, in the worst case, MEN would end up duplicating all customers and solving the extended network in the last iteration, which clearly would result in a bad performance of the

**Table 3 Evaluation of the dropoff relaxation on the combined demand case (sec gives averages across all instances in the row, considering t.lim. for instances unsolved to proven optimality).**

| | | TINE (for SVRPDSP-D) | | | | | | | MEN (for SVRPDSP) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GLS set | # | opt | sec | $\text{gap}_\text{r}$ | $\text{gap}_\text{c}$ | nodes | feas | $\text{gap}_d$ | opt | sec | $\text{gap}_\text{r}$ | $\text{gap}_\text{c}$ | nodes | dupl | iter |
| $15 \leq n \leq 30$ | 28 | 28 | 0.3 | 4.74 | 0.59 | 236 | 26 | 0.00 | 28 | 0.2 | 4.74 | 0.59 | 206 | 0.1 | 1.1 |
| $32 \leq n \leq 50$ | 24 | 24 | 10.4 | 6.27 | 0.88 | 6785 | 24 | 0.00 | 24 | 6.5 | 6.27 | 0.98 | 5209 | 0.4 | 1.4 |
| $71 \leq n \leq 100$ | 16 | 16 | 352.6 | 10.66 | 0.83 | 40085 | 10 | 0.05 | 15 | 416.6 | 10.70 | 0.88 | 32174 | 1.2 | 2.2 |
| totals | 68 | 68 | | | | | 60 | | 67 | | | | | | |
| averages | | | 86.78 | 6.67 | 0.75 | 11924 | | 0.01 | | 100.4 | 6.68 | 0.79 | 9493 | 0.5 | 1.5 |

algorithm. However, in practice we can see that this is not the case. For the large size instances, MEN duplicates about 2 customers on average, and only 4 customers out of 100 in the worst case. These good results confirms a principle already hinted in Subramanian et al. (2011) for a related problem: instead of burdening a formulation in order to enforce conditions that seldom happen in practice, it is better to treat them in a lazier way.

### 7.3. The case of mandatory pickups

We now focus on the special SVRPDSP case arising when all pickups are *mandatory* (see Figure 2-(b)). This is known in the literature as the *single vehicle routing problem with pickups and deliveries* (SVRPPD). To solve SVRPPD we adapted our exact algorithms TA, 2S, and MEN of Section 6 as follows: (i) we set $y_j = 1$ for all $j \in P \cup PD$; (ii) we removed the redundant constraints (29)–(31) and (34)–(35); (iii) we rounded up to the next integer the right hand side of (32), obtaining $x(\delta^+(S)) \geq \left\lceil \sum_{j \in S} p_j/Q \right\rceil$, and the right hand side of (33) after having divided the constraint by $Q$, obtaining $x(\bar{S} : S) \geq \lceil (d(V) + p(S) - d(S))/Q \rceil - 1$. Note that in this case the separation procedure for (33) discussed in Proposition 8 is not exact anymore but only heuristic. We call the resulting algorithms TA-MP, 2S-MP, and MEN-MP. We tested these algorithms on the benchmark set proposed by Gribkovskaia et al. (2007), that contains 34 instances with size ranging from 15 to 100 customers (GHLV set in the following).

Table 4 compares the results of our exact algorithms with those of the formulation by Gribkovskaia et al. (2007)(GHLV), that we implemented and ran for one CPU hour on our PC. The GHLV formulation performs poorly, solving to proven optimality just one small size instance. Among our algorithms 2S-MP is slightly better than TA-MP, as it solves one more instance to proven optimality. The algorithm having the best performance is MEN-MP, because it can solve to proven optimality all benchmark instances in about 22 seconds on average, and 560 in the worst case. Overall, these results attest the efficiency of our algorithms also for this problem variant, showing that it is possible to extend our approaches to other related 1-M-1 problems obtaining useful results.

**Table 4 Computational results on the SVRPPD (i.e., mandatory pickups case) (sec gives averages across all instances in the row, considering t.lim. for instances unsolved to proven optimality).**

| | | literature | | | new algorithms | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | GHLV | | | TA-MP | | | 2S-MP | | | MEN-MP | | |
| GHLV set | # | opt | sec | gap | opt | sec | gap | opt | sec | gap | opt | sec | gap |
| $15 \leq n \leq 30$ | 14 | 1 | 3419.2 | 10.8 | **14** | 0.4 | 0.0 | **14** | 0.5 | 0.0 | **14** | 0.4 | 0.0 |
| $32 \leq n \leq 50$ | 12 | 0 | t.lim. | 16.2 | 11 | 559.7 | 0.1 | 11 | 471.3 | 0.2 | **12** | 3.6 | 0.0 |
| $71 \leq n \leq 100$ | 8 | 0 | t.lim. | 22.0 | 7 | 745.4 | 0.1 | **8** | 198.9 | 0.0 | **8** | 88.2 | 0.0 |
| totals | 42 | 1 | | | 32 | | | 33 | | | **34** | | |
| averages | | | 3525.5 | 19.9 | | 373.1 | 0.1 | | 213.4 | 0.1 | | 22.2 | 0.0 |

### 7.4. Practical insights on the usage of SVRPDSP

SVRPDSP models many practical problems, so it may be interesting for decision makers to understand how important are in practice the main features of this model. To this end we performed two analyses. The first one aims at estimating the savings that can be obtained by allowing mixed deliveries instead of imposing a backhaul distribution. In Table 5 we show the average percentage gap between the optimal solution values of SVRPDSP and of SVRPDSP with backhauls. By allowing interspersed pickups and deliveries it is possible to save on average 11.05% and 12.7% for instances of sets SB and GMO, respectively. The savings increase when the size of the instances increase, reaching almost 15% for the largest instances of the GMO set. It thus appears very convenient in practice to allow mixed deliveries when possible.

Our second analysis aims at evaluating the gap in the total driving distance when allowing selective pickups instead of enforcing mandatory pickups, thus directly comparing SVRPDSP and SVRPPD. This comparison is affected by the values of the revenues assigned to the pickups, and moreover it is possible only when $Q \geq p(V)$. For the SB set revenues are very high, so all pickups are

**Table 5 Average gaps between the optimal solutions of SVRPDSP and SVRPDSP with backhauls.**

| SB set | # | gap | GMO set | # | gap |
|---|---|---|---|---|---|
| $n$=10 | 24 | 9.30 | $25 \leq n \leq 30$ | 18 | 12.28 |
| $n$=20 | 21 | 10.27 | $38 \leq n \leq 60$ | 39 | 11.97 |
| $n$=30 | 18 | 14.29 | $68 \leq n \leq 90$ | 17 | 14.80 |
| averages | | 11.05 | averages | | 12.70 |

**Table 6 Average gap in the total driving distance between SVRPDSP and SVRPPD.**

| GMO set | # | gap |
|---|---|---|
| $25 \leq n \leq 30$ | 15 | 5.96 |
| $38 \leq n \leq 60$ | 30 | 9.74 |
| $68 \leq n \leq 90$ | 14 | 7.49 |
| averages | | 8.25 |

made even in the selective case, and for the GLS set $Q < p(V)$ for all instances. We thus considered the GMO set, that shows a better balance between costs and revenues and for which 59 out of 74 instances have $Q \geq p(V)$. The results are shown in Table 6. On average, more than 8% of the total driving distance can be saved by allowing selective pickups. This confirms in practice the interest in the selective distribution first mentioned by Golden and Assad (1986).

### 7.5. New large size instances

Given that our algorithms were able to provide proven optimal solutions for all benchmark instances in the single vehicle problem variants that we addressed, we decided to better assess the difficulty of the problems by testing our algorithms on new instances. We first attempted instances with asymmetric cost matrices, but did not find relevant differences with respect to the available benchmark instances that have symmetric costs. We then attempted instances with larger numbers of customers, that we created as follows. We selected from the *vehicle routing problem* (VRP) library by Vigo (1999) 4 large size instances having between 120 to 199 customers each, namely: E121-07c, E135-07f, E151-12b, and E200-16b. For the case of selective pickups, we used the procedure by Gribkovskaia et al. (2008) to convert these VRP instances into SVRPDSP ones, obtaining a new set (BI set) containing 16 instances. For the case of mandatory pickups, we used the procedure in Gribkovskaia et al. (2007) to convert the VRP instances into SVRPPD ones, obtaining an additional set (BI-MP set) with 8 instances. In both sets all customers have combined demands, and this increases their complexity.

We addressed these new sets with our best algorithms for the two problem variants, namely MEN and MEN-MP. The results are given in Table 7. For SVRPDSP, MEN is not able to solve

**Table 7 Computational results on large size SVRPDSP and SVRPPD instances (sec gives averages across all instances in the row, considering t.lim. for instances unsolved to proven optimality).**

| Selective pickups (SVRPDSP) | | | | | | Mandatory pickups (SVRPPD) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | MEN | | | | | | MEN-MP | | |
| BI set | # | opt | sec | gap | nodes | BI-MP set | # | opt | sec | gap | nodes |
| $n$=120 | 4 | 1 | 3208.6 | 1.6 | 208570 | $n$=120 | 2 | 2 | 960.3 | 0.0 | 42510 |
| $n$=135 | 4 | 0 | t.lim. | 2.2 | 91743 | $n$=135 | 2 | 2 | 2257.6 | 0.0 | 12980 |
| $n$=151 | 4 | 4 | 669.6 | 0.0 | 35754 | $n$=151 | 2 | 2 | 341.5 | 0.0 | 1922 |
| $n$=200 | 4 | 2 | 2654.0 | 0.7 | 49967 | $n$=200 | 2 | 0 | t.lim. | 1.1 | 46100 |
| totals | 16 | 7 | | | | totals | 8 | 6 | | | |
| averages | | | 2533.0 | 1.1 | 96508 | averages | | | 1789.9 | 0.3 | 25878 |

instances with 120 and 135 customers, although the gap remains quite low, around 2% on average. SVRPPD appears to be easier, as MEN-MP solves all instances with up to 150 customers. The 2 instances with 200 customers are still unsolved, although the gap is around 1%. More research is thus envisaged to find optimal solutions for large size instances.

### 7.6. The case of multiple vehicles

In many real-world scenarios a *fleet* of vehicles instead of a single one is available to serve all customers. We thus decided to extend our study to the case of multiple vehicles, known in the literature as the *multiple vehicle routing problem with deliveries and selective pickups* (MVRPDSP). MVRPDSP is the generalization of SVRPDSP in which a fleet of $k$ homogeneous capacitated vehicles is available to perform pickups and deliveries. Transshipment among vehicles is not allowed, and neither are dropoffs and split deliveries or pickups. To our knowledge, the only paper directly approaching MVRPDSP is the one by Bruck and dos Santos (2012), in which the authors propose a basic three-index mathematical formulation and a hybrid cluster-first heuristic.

To solve MVRPDSP we adapted our best exact approach, namely the MEN algorithm, as follows. Constraints (33) are not valid for the multiple vehicle case, as they are based on the assumption that a vehicle leaves the depot with exactly $d(V)$ units of delivery commodity. We thus replaced them by a set of constraints based on the classical capacity-cut inequalities for the capacitated vehicle routing problem:

$$x(\delta^+(S)) \geq \max\Big( \left\lceil \frac{d(S)}{Q} \right\rceil, \frac{\sum_{i \in S} p_i y_i}{Q} \Big) \qquad \forall S \subseteq V \setminus \{0\} : |S| > 1. \tag{36}$$

As for the single vehicle case, inequalities are separated only at integer points. We thus modified the original separation procedures to deal with the new situation and search for subsets possibly leading to violations. For (36) this results in a heuristic separation. Moreover, when a set $S$ leading to a violation is found, only the most violated constraint between $x(\delta^+(S)) \geq \lceil d(S)/Q \rceil$ and $x(\delta^+(S)) \geq \sum_{i \in S} p_i y_i / Q$ is added to the model. In case no violation is found, then we separate the Benders cuts (26) to ensure feasibility. From now on, let us call MEN-Multi the resulting algorithm.

To analyze the computational behavior of MEN-Multi we modified the SVRPDSP benchmark set GLS by considering a fleet of $k$ vehicles, each having capacity $Q' = \lceil \alpha Q \rceil$, where $Q$ is the vehicle capacity in the original instance. We tested 10 different options for the pair $(\alpha, k)$, namely (0.6, 2), (0.6, 3), (0.6, 4), (0.5, 3), (0.5, 4), (0.5, 5), (0.4, 3), (0.4, 4), (0.4, 5), and (0.4, 6). We obtained in this way a new set of 680 instances, that we call GLS-Multi in the following.

Table 8 shows the results of our computational experiments (detailed results are given in Tables EC.19–EC.27, in Section EC.4.5 of the electronic companion). The algorithm is very efficient and the number of iterations that it requires does not increase significantly relative to the single vehicle case. For $\alpha = 0.6$ it is able to solve all instances to proven optimality. It also solves 193 instances out of 204 for $\alpha = 0.5$, and 264 instances out of 272 for $\alpha = 0.4$. The average percentage gaps remain very low. This analysis shows how our algorithms and ideas can be successfully extended to problem variants with multiple vehicles. An interesting insight in the problem difficulty is that MEN performs very well for those cases in which capacity constraints are rather loose. This can be explained by the fact that, for a given value of $\alpha$, instances get easier for MEN when $k$ increases.

**Table 8 Computational results of MEN-Multi for the benchmark set GLS-Multi (sec gives averages across all instances in the row, considering t.lim. for instances unsolved to proven optimality).**

| | | $\alpha = 0.6$ | | | | $\alpha = 0.5$ | | | | $\alpha = 0.4$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $k$ | # | opt | sec | gap | iter | opt | sec | gap | iter | opt | sec | gap | iter |
| 2 | 68 | 68 | 111.2 | 0.00 | 1.7 | | | | | | | | |
| 3 | 68 | 68 | 56.2 | 0.00 | 1.1 | 62 | 454.5 | 0.07 | 1.8 | 63 | 642.2 | 0.11 | 2.1 |
| 4 | 68 | 68 | 30.3 | 0.00 | 1.1 | 64 | 304.4 | 0.03 | 1.6 | 66 | 494.3 | 0.04 | 1.6 |
| 5 | 68 | | | | | 67 | 315.7 | 0.00 | 1.2 | 67 | 311.5 | 0.00 | 1.3 |
| 6 | 68 | | | | | | | | | 68 | 137.8 | 0.00 | 1.3 |

To better understand this algorithmic behavior, we performed an additional test involving instances with very tight capacities. We thus extended GLS-Multi set by attempting 3 more options for the pair $(\alpha, k)$, namely (0.52, 2), (0.35, 3), and (0.27, 4), for a total of 204 instances. We solved these instances using MEN-Multi. Because of the increased instance difficulty, we initialized our algorithm with the UB obtained by our re-implementation of the heuristic in Bruck and dos Santos (2012), which only requires a few CPU seconds. A time limit of one CPU hour was imposed on each instance. Detailed results are provided in Tables EC.28–EC.30, in Section EC.4.5 of the electronic companion.

The outcome of these tests confirms that instances with tighter capacity constraints are likely to be more difficult to solve with MEN-Multi. For the group of instances with two vehicles, only 38 instances out of 68 could be solved to proven optimality in one hour of computation. For two of these instances, the algorithm iterated 11 times (while for single vehicle instances the maximum experienced number of iterations with MEN was 5). The average gap is 4.69%, but it is usually above 20% for instances having $n = 100$. For the case of three vehicles, the number of proven optimal solutions decreased to 20 out of 68, and the average gap raised to about 11%. For the case of four vehicles, the number of proven optimal solutions further decreased to 14 out of 68, and the average gap further raised to slightly more than 13%.

To solve instances with very tight capacities, other approaches such as branch-and-price might prove more efficient. One could adapt, for example, the algorithm developed for MVRPDSP with time windows by Gutiérrez-Jarpa et al. (2010). The column generation process in this algorithm is based on a sophisticated label-setting, exploiting bounded bi-directional search and decremental search space accelerations (see, e.g., Righini and Salani 2008). Nevertheless, it strongly exploits the existence of time windows and its behavior could possibly be inadequate if these are removed. In such a case, one could invoke enhanced techniques such as the *ng*-route relaxation by Baldacci et al. (2011) or the subset-row cuts by Jepsen et al. (2008). These techniques have been used with success for the VRP (see, e.g., the recent contribution by Pecin et al. 2016), but to the best of our knowledge they have not yet been applied to solve 1-M-1 pickup and delivery problems.

## 8. Conclusions and Future Research Directions

We studied the class of one-to-many-to-one single vehicle pickup and delivery problems, in which a single capacitated vehicle is used to serve a set of customers requiring a delivery, a pickup, or both. We concentrated on the most studied problem in this class, known as the *single vehicle routing problem with deliveries and selective pickups* (SVRPDSP), in which deliveries are mandatory but pickups are optional and generate a revenue if performed.

Most of the approaches in the literature solve SVRPDSP by duplicating each customer requiring both a delivery and a pickup into two separate customers, the first requiring only the delivery and the second only the pickup, and then look for Elementary solutions in the resulting extended network. In this work we decided instead to consider the original problem network, and work with solutions that may be non-Elementary. Although this may seem like a simple modification, by allowing non-Elementary solutions the combinatorial structure of the problem changes drastically, and this offers interesting opportunities for the development of efficient exact algorithms.

For the single demand case, where all customers require either a delivery or a pickup, but not both, we developed a formulation looking for Elementary tours on the extended network and three non-Elementary formulations, two of which were solved by branch-and-cut. Our best algorithm makes use of several families of valid inequalities, and could solve to proven optimality all benchmark instances within a few CPU seconds on a standard PC. For the combined demand case, where customers may require both delivery and pickup, our previous non-Elementary formulations only provide a lower bound to the problem because they allow intermediate dropoffs of commodities. To address this issue we developed three exact algorithms, that extended the previous branch-and-cut by considering different options for removing infeasible solutions. Also in this case our best exact algorithm could solve all benchmark instances, although it required a few CPU hours in the worst case. We then adapted our algorithms to solve the case of mandatory pickups, which turned out to be easier in practice than SVRPDSP, as our best algorithm could solve all benchmark instances in just a few CPU minutes. We also solved the problem where a fleet of homogeneous

vehicles is used, showing that our best exact algorithm is still very efficient, especially when the capacity constraint is loose. These last tests also had the merit of showing how our algorithms can be adapted to solve other problem variants.

As these problems are very challenging, it is not difficult to construct instances in which our algorithms fail to provide proven optimal solutions in limited time. Further research is thus envisaged to solve large size instances of the problem variants that we addressed.

The results presented in this paper indicate that working on the original network of the problem is more challenging than working on the extended network, but worth the effort given the good computational results. Our idea of repeatedly using a non-Elementary formulation to quickly converge to optimality can be applied to a wide range of vehicle routing problems, such as (single or multiple) vehicle routing problems with split deliveries, and vehicle routing problems with load transshipment among vehicles. This research does not have to be limited to one-to-many-to-one pickup and delivery problems, as done in this paper, but could focus also on many-to-many problems (as the ones addressed in, e.g., Chemla et al. 2013 and Salazar-González and Santos-Hernández 2015), and one-to-one pickup and delivery routing problems with split deliveries (see, e.g., Nowak et al. 2008). This appear to be an interesting and innovative research direction.

## Acknowledgments

This research has been partially supported by CAPES/Brazil under grant PVE n° A007/2013. We thank an associate editor and three referees, whose comments helped improved the quality of this paper.

## Brief Author Biographies

**Bruno P. Bruck** is currently a postdoctoral fellow from Université de Montréal at the Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport (CIRRELT). His research interests are in the development of exact and heuristic algorithms for vehicle routing, time slot management, railway optimization, carpooling, and bike sharing problems.

**Manuel Iori** is associate professor of Operations Research at the Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia. His research activities include the study and development of mathematical models and solution algorithms for combinatorial optimization and logistics, with a focus on problems such as bin packing, multidimensional cutting and packing, scheduling, traveling salesman, and vehicle routing.

# Supplemental Material to: Non-Elementary Formulations for Single Vehicle Routing Problems with Pickups and Deliveries

Bruno P. Bruck
Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Via Amendola 2, 42122 Reggio Emilia, Italy, bruno.petratobruck@unimore.it

Manuel Iori
Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Via Amendola 2, 42122 Reggio Emilia, Italy, manuel.iori@unimore.it

The electronic companion is structured as follows. First we provide in Section 9 the proofs of the statements in the paper. Then, we provide an example of a TCNE solution, as well as details on the algorithms adopted to remove split deliveries or pickups, detect dropoffs, and perform the Benders decomposition in Section 10. In Section 11 we present a computational comparison with existing metaheuristic algorithms. Finally, we give detailed computational results for each run of our algorithms in Section 12.

## 9. Proofs of Statements

### 9.1. Property 2

PROPERTY 2. *For the CD case formulation TCNE provides a relaxation of SVRPDSP.*

*Proof* We prove that TCNE searches a larger solution space than TCEE, by first mapping any feasible TCEE solution into a feasible TCNE solution having the same cost, and then providing a counterexample which is feasible for TCNE but not for TCEE. The statement then follows because TCEE optimally solves SVRPDSP.

For the first step, let $(\bar{y}_j,\ \bar{x}_{ij},\ \bar{f}^d_{ij},\ \bar{f}^p_{ij})$ be a feasible TCEE solution, and let $(\tilde{y}_j,\ \tilde{x}_{ij},\ \tilde{f}^d_{ij},\ \tilde{f}^p_{ij})$ denote the TCNE solution that we aim to construct. For the $y$ variables, by recalling the notation given in Definition 2, we simply set $\tilde{y}_j = \bar{y}_j$ for $j \in P$ and $\tilde{y}_j = \bar{y}_{\pi(j)}$ for $j \in PD$. For the $x$ variables, we set

$$\tilde{x}_{ij} = \bar{x}_{ij} \qquad \forall\ (i,j) \in A : i,j \in V \setminus PD, \tag{37}$$

$$\tilde{x}_{ij} = \bar{x}_{\sigma(i),j} + \bar{x}_{\pi(i),j} \qquad \forall\ (i,j) \in A : i \in PD, j \in V \setminus PD, \tag{38}$$

$$\tilde{x}_{ij} = \bar{x}_{i,\sigma(j)} + \bar{x}_{i,\pi(j)} \qquad \forall\ (i,j) \in A : i \in V \setminus PD, j \in PD, \tag{39}$$

$$\tilde{x}_{ij} = \bar{x}_{\sigma(i),\sigma(j)} + \bar{x}_{\sigma(i),\pi(j)} + \bar{x}_{\pi(i),\sigma(j)} + \bar{x}_{\pi(i),\pi(j)} \qquad \forall\ (i,j) \in A : i,j \in PD. \tag{40}$$

The construction of the $\tilde{f}^d_{ij}$ and $\tilde{f}^p_{ij}$ variables is performed in the same manner. By simple algebraic operations we can check that $\sum_{j \in P \cup PD} r_j(1-\tilde{y}_j) = \sum_{j \in P'} r_j(1-\bar{y}_j)$ and $\sum_{(i,j)\in A} c_{ij}\tilde{x}_{ij} = \sum_{(i,j)\in A'} c_{ij}\bar{x}_{ij}$, so the resulting TCNE solution has the same cost of the initial TCEE one. Coming to the feasibility of constraints (13)–(25), let us first focus on the vehicle flow conservation imposed by (17). When $j \in V \setminus PD$, by applying (37)–(39), and recalling that $V = \{0\} \cup P \cup D \cup PD$, $D' = D \cup \{\sigma(i) : i \in PD\}$, $P' = P \cup \{\pi(i) : i \in PD\}$, and $V' = \{0\} \cup P' \cup D'$, we get:

$$\begin{aligned}
\sum_{i \in V\setminus\{j\}} (\tilde{x}_{ij} - \tilde{x}_{ji}) &= \sum_{i \in V\setminus PD\setminus\{j\}} (\tilde{x}_{ij} - \tilde{x}_{ji}) + \sum_{i \in PD} (\tilde{x}_{ij} - \tilde{x}_{ji}) \\
&= \sum_{i \in V\setminus PD\setminus\{j\}} (\bar{x}_{ij} - \bar{x}_{ji}) + \sum_{i \in PD} (\bar{x}_{\sigma(i),j} + \bar{x}_{\pi(i),j} - \bar{x}_{j,\sigma(i)} - \bar{x}_{j,\pi(i)}) \\
&= \sum_{i \in D\cup P\cup\{0\}\setminus\{j\}} (\bar{x}_{ij} - \bar{x}_{ji}) + \sum_{i \in D'\setminus D} (\bar{x}_{ij} - \bar{x}_{ji}) + \sum_{i \in P'\setminus P} (\bar{x}_{ij} - \bar{x}_{ji}) \\
&= \sum_{i \in V'\setminus\{j\}} (\bar{x}_{ij} - \bar{x}_{ji})
\end{aligned}$$

which is equal to 0 because $\bar{x}$ satisfies (4). When $j \in PD$ we get the same results by applying (37), (39), and (40), so (17) is not violated by $\tilde{x}$. Analogous algebraic computations confirm the feasibility of $\tilde{x}$ with respect to the remaining constraints, and this ends the first step of our proof.

For the second step we simply refer to Figure 4-(a) of Section 4.2, which shows a counterexample which is feasible for TCNE but not for TCEE, because of the above mentioned intermediate dropoff of the load at vertex 1. □

### 9.2. Property 3

PROPERTY 3. *A TCNE solution with split deliveries or pickups can be transformed into a solution having the same cost and for which no delivery or pickup is split.*

*Proof* Suppose an optimal solution to TCNE is given in which a combined vertex $i \in PD$ is visited twice and its delivery and/or pickup are split. Let $d^1_i$ and $d^2_i$, respectively $p^1_i$ and $p^2_i$, denote the delivery, respectively pickup, quantities served during the first and second visit to $i$, respectively.

Now build a modified solution in which all the delivery $d_i^1 + d_i^2$ is performed in the first visit, and all the pickup $p_i^1 + p_i^2$ in the second visit. This modification leaves unchanged the load on the vehicle from 0 to the first visit to $i$ and from the second visit to $i$ to 0, and decreases it by $d_i^2 + p_i^1$ from the first to the second visit to $i$. The new solution is still feasible, has no split in $i$, and has cost equal to the original one. The process can be repeated until all deliveries and pickups are not split. □

### 9.3. Property 5

PROPERTY 5. *The following inequality is valid for SVRPDSP:*

$$x(\delta^+(S)) \geq \sum_{j \in S} \frac{1}{|S|} y_j \qquad \forall S \subseteq P. \tag{31}$$

*Proof* We could use an algebraic proof similar to that of Gutiérrez-Jarpa et al. (2009) for the symmetric case, but we opted to proceed in a simpler fashion. First notice that there are two mutually exclusive cases for the vehicle flow that leaves a subset $S$ of pickup vertices, that can be represented by the two following inequalities:

$$x(\delta^+(S)) = 0 \qquad \forall S \subseteq P : \sum\nolimits_{j \in S} y_j = 0, \tag{41}$$

$$x(\delta^+(S)) \geq 1 \qquad \forall S \subseteq P : \sum\nolimits_{j \in S} y_j \geq 1. \tag{42}$$

Notice also that $\sum_{j \in S} \frac{1}{|S|} y_j = 0$ when $\sum_{j \in S} y_j = 0$, and $0 < \sum_{j \in S} \frac{1}{|S|} y_j \leq 1$ when $\sum_{j \in S} y_j \geq 1$, so (31) is valid on both the two mutually exclusive cases and the statement follows. □

### 9.4. Property 6

PROPERTY 6. *The following inequality is valid for SVRPDSP:*

$$x(\delta^+(S)) \geq \sum_{j \in S} \frac{p_j}{Q} y_j \qquad \forall S \subseteq P. \tag{32}$$

*Proof* We proceed again by making use of the two mutually exclusive cases considered in the previous proof of Property 5. The validity of (32) comes from the fact that: (i) $\sum_{j \in S} \frac{p_j}{Q} y_j = 0$ when $\sum_{j \in S} y_j = 0$, so (41) is satisfied; (ii) $\sum_{j \in S} \frac{p_j}{Q} y_j > 0$ and, because of (34), $\sum_{j \in S} \frac{p_j}{Q} y_j \leq 1$, so also (42) is satisfied. Constraint (32) is thus valid for any possible set $S \subseteq P$. □

### 9.5. Property 7

PROPERTY 7. *The following inequality (capacity-cut constraint) is valid for SVRPDSP.*

$$Qx(\bar{S}:S) - \sum_{j\in S} p_j y_j \geq d(V) - d(S) - Q \qquad \forall S \subseteq V \setminus \{0\} : p(S) - d(S) > Q - d(V). \tag{33}$$

*Proof* First recall that according to our notation $d(S) = \sum_{j\in S} d_j$, $p(S) = \sum_{j\in S} p_j$, and $x(\bar{S}:S) = \sum_{i\in\bar{S}} \sum_{j\in S} x_{ij}$. We can rewrite (33) as

$$x(\bar{S}:S) \geq \frac{d(V) - d(S)}{Q} + \frac{\sum_{j\in S} p_j y_j}{Q} - 1 \qquad \forall S \subseteq V \setminus \{0\} : p(S) - d(S) > Q - d(V). \tag{43}$$

Note that $\frac{d(V)-d(S)}{Q} \leq 1$ because $d(V) - d(S) \leq d(V) \leq Q$, and $\frac{\sum_{j\in S} p_j y_j}{Q} \leq 1$ because of (34), so the right hand side of (43) takes a value that is at most one. Consequently, (43) imposes that the vehicle flow that goes from $\bar{S}$ to $S$ is at least one for those sets $S$ and solutions $y$ satisfying $d(V) - d(S) + \sum_{j\in S} p_j y_j > Q$ (and it is instead loose if $d(V) - d(S) + \sum_{j\in S} p_j y_j \leq Q$).

To see that this condition is always satisfied, we consider two mutually exclusive cases:

1. The vehicle leaves the depot and first visits $\bar{S}$. In this case, for connectivity, the vehicle is forced at some point to travel from $\bar{S}$ to $S$, so the vehicle flow from $\bar{S}$ to $S$ is at least one and (43) is satisfied;

2. The vehicle leaves the depot and first visits $S$. In this case notice that the vehicle leaves the depot with a load equal to $d(V)$. Thus, if it visited all the vertices in $S$ performing the pickups described by $y$, its load when finally leaving $S$ would be $d(V) - d(S) + \sum_{j\in S} p_j y_j$, but that would exceed the vehicle capacity $Q$ because of the above assumption on $S$ and $y$. Thus, the vehicle is forced to leave $S$ at some point to perform some deliveries in $\bar{S}$ and increase its residual loading space, and then return to $S$ later on. Consequently, also in this case (43) is satisfied and that proves the statement. □

### 9.6. Property 8

PROPERTY 8. *Constraint* (33) *can be separated in polynomial time.*

*Proof* We are given a possibly fractional solution $(\bar{x}, \bar{y})$ to formulation TINE. Recall that $p(S) = \sum_{j\in S} p_j$, and let us use the following additional notation: $y(S) = \sum_{j\in S} p_j \bar{y}_j$ and $\underline{y}(S) = \sum_{j\in S} p_j(1 - \bar{y}_j)$. We first partition $p(S)$ as

$$p(S) = \sum_{j\in S} p_j \bar{y}_j + \sum_{j\in S} p_j(1 - \bar{y}_j) = y(S) + \underline{y}(S). \tag{44}$$

We also partition the whole pickups as $p(V) = p(S) + p(\bar{S})$. By including this in (44) we get

$$y(S) = p(S) - \underline{y}(S) = p(V) - p(\bar{S}) - \underline{y}(S). \tag{45}$$

Let us now rewrite constraints (33) as

$$x(S:\bar{S}) \geq \frac{d(V)}{Q} - \frac{d(S)}{Q} + \frac{y(S)}{Q} - 1. \tag{46}$$

Then, by using (45), we can rewrite (46) as

$$x(S:\bar{S}) \geq \frac{d(V)}{Q} - \frac{d(S)}{Q} + \frac{p(V) - p(\bar{S}) - \underline{y}(S)}{Q} - 1,$$

so we obtain

$$x(S:\bar{S}) + \frac{d(S)}{Q} + \frac{p(\bar{S})}{Q} + \frac{\underline{y}(S)}{Q} \geq \frac{d(V)}{Q} + \frac{p(V)}{Q} - 1. \tag{47}$$

Consequently, (47) is equivalent to (33). We now present an algorithm to exactly separate (47). We start by the supporting graph already used for the separation of (27) and (28), namely, $\bar{G} = \left(\bar{V}, \bar{A}\right)$, where $\bar{V} = V \setminus \{i \in P : \bar{y}_i = 0\}$ and an arc $(i,j) \in \bar{A}$ has capacity equal to $\bar{x}_{ij}$. Once again we solve a series of max-flow problems, one for each vertex $i \in \bar{V}$, but in this case we first need to modify $\bar{G}$ to obtain a new supporting graph for each $i$. This new graph, that we define $\bar{G}(i) = \left(\bar{V}(i), \bar{A}(i)\right)$, is built as follows:

- copy $\bar{G}$ in $\bar{G}(i)$;
- remove vertex 0 from $\bar{G}(i)$;
- add a new vertex $n+1$ to $\bar{G}(i)$;
- connect $n+1$ to all vertices $j \in \bar{V}(i) \setminus \{n+1\}$ with arcs of capacity equal to $\frac{d_j}{Q} + \frac{p_j(1-\bar{y}_j)}{Q}$;
- connect all vertices $j \in \bar{V}(i) \setminus \{i\}$ to $i$ with arcs of capacity equal to $\bar{x}_{ji} + \frac{p_j}{Q}$.

Now compute the maximum flow from $n+1$ to $i$ in $\bar{G}(i)$, and let $f$ be the resulting flow value. Let also $S$ be the subset identified by the min cut and containing vertex $i$, and $\bar{S}$ the remaining subset containing $n+1$. If $f < \frac{d(V)}{Q} + \frac{p(V)}{Q} - 1$, then the inequality (47) is violated by the current set $S$, and can be added to the model. Otherwise, we proceed with the next vertex $i$, until either a violation is detected or max-flow computations have been performed for all vertices.

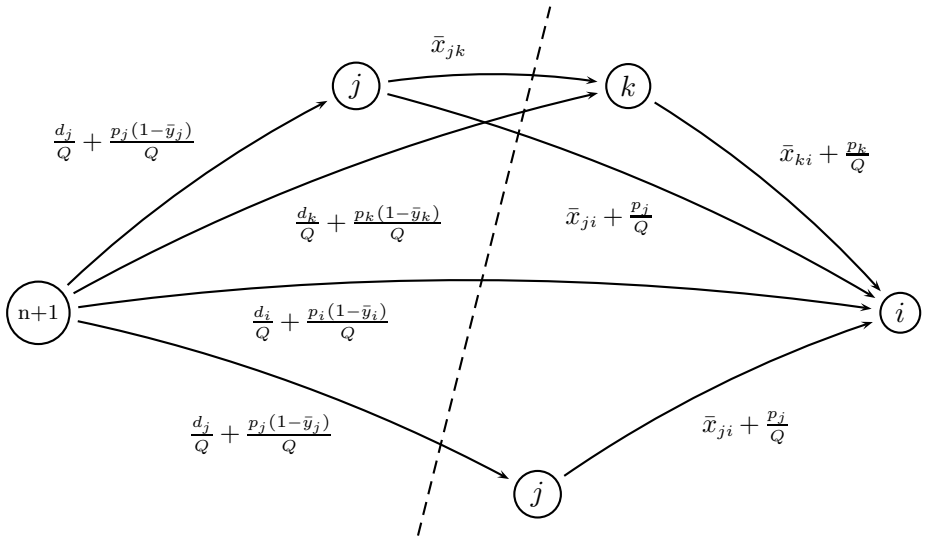



**Figure 5** **Max-flow separation procedure for capacity-cut constraints** (33) **(the max-flow is computed from vertex** $n+1$ **to customer** $i$**).**

To see that this procedure exactly separates (47), let us focus on the example depicted in Figure 5. The value $f$ of the maximum flow is equal to that of the min cut, represented by the dashed line in the figure. Set $S$ is at the right of the min cut, and $\bar{S}$ at the left. For each customer $j \in \bar{V}(i) \setminus \{i, n+1\}$, there are two possible cases:

1. if $j \in S$, then by construction its contribution to $f$ is given by $\frac{d_j + p_j(1-\bar{y}_j)}{Q}$;
2. if instead $j \in \bar{S}$, then its contribution to $f$ is equal to $\frac{p_j}{Q} + \sum_{k \in S} \bar{x}_{jk}$.

Hence, summing up the contributions from all vertices, we obtain

$$f = \sum_{j \in \bar{S}} \frac{p_j}{Q} + \sum_{j \in \bar{S}} \sum_{k \in S} \bar{x}_{jk} + \sum_{j \in S} \left( \frac{d_j + p_j(1-\bar{y}_j)}{Q} \right) = x(\bar{S} : S) + \frac{D(S)}{Q} + \frac{P(\bar{S})}{Q} + \frac{\underline{y}(S)}{Q}.$$

The correctness of the procedure follows from the fact that $f$ is equivalent to the left hand side of (47), so if $f < \frac{d(V)}{Q} + \frac{p(V)}{Q} - 1$, then $S$ induces a violated cut, otherwise no violation exists. For the sake of clarity notice that, in case a violation is found, we are sure that the condition imposed in (33), i.e., $p(S) - d(S) > Q - d(V)$, is satisfied. This can be observed by looking at the rewritten form of the capacity-cut constraints (46). In case $p(S) - d(S) \leq Q - d(V)$, the right hand side of (46) is non-positive and thus the inequality cannot be violated. $\square$

# 10. Supplementary material and details on the implemented algorithms

## 10.1. Example of a TCNE solution with an arc being traversed twice

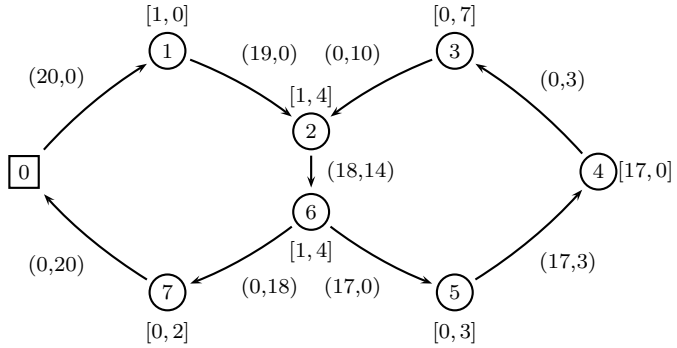
Due to the fact that TCNE works on the original network and multiple visits are allowed to CD customers, it might happen that an arc between two CD customers, say $i$ and $j$, is traversed twice, i.e., $x_{ij} = 2$. In this case the associated total flow passing through the arc is the sum of the flows in the first and second traversals.

To better illustrate this situation, consider the example depicted in Figure 6. Vertex 0 represents the depot and vertices 1 to 7 are the customers. The notation is the same one adopted in the paper. The vehicle capacity is $Q = 20$. In Figure 6-(a) arc $(2, 6)$ is being traversed twice with a total flow $f^d_{26} + f^p_{26} = 32$, which is feasible because it does not exceed $2Q$ (refer to constraints (18)). In Figure 6-(b) the flow is disaggregated in the two traversals, (18,0) during the first traversal and (0,14) during the second one. Notice that, as expected, in neither of them is the vehicle capacity violated. The flow values can be obtained by using procedure removeSplit, described in the next section.

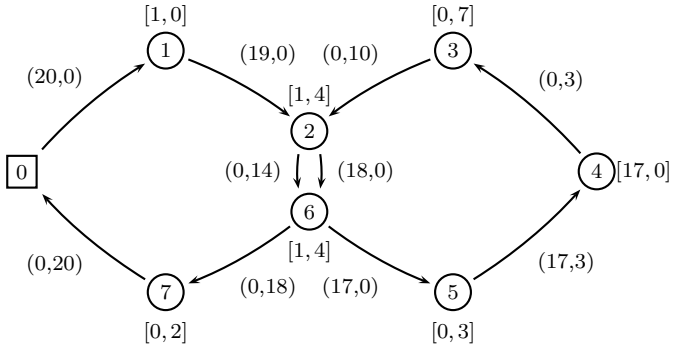
## 10.2. Procedure for removing split deliveries or pickups

We are given a solution to formulation TCNE that possibly contains split deliveries or pickups. Following Property 3, we want to compute the flows of delivery and pickup commodity, $f^d_{ij}$ and $f^p_{ij}$, that do not require any split delivery or pickup. This can be obtained by making use of the recursive procedure removeSplit given in Algorithm 1.

In detail, let $\tilde{x}_{ij}$ and $\tilde{y}_j$ denote the values taken by the corresponding variables in the TCNE solution, and compute the number of times in which a vertex $j$ has been visited as $\tilde{\delta}_j = \sum_{i \in V} \tilde{x}_{ij}$. Initialize an array *visits*$[j]$ to 0 for all vertices $j \in N$, and set to 0 also the value of the current vertex (*current*, in the algorithm), of the pickup load (*pLoad*), and of the delivery load (*dLoad*). Then invoke removeSplit. This procedure attempts to build a path that starts and ends at the depot, and visits each customer $j$ exactly $\tilde{\delta}_j$ times. At each forward step it connects the path to a vertex $i$ that can be reached from *current* (there can be up to two such vertices), and updates the flows by:




(a) Aggregated flows on arc $(2, 6)$




(b) Disaggregated flows on arc $(2, 6)$

**Figure 6 Example of a TCNE solution with an arc being traversed twice.**

- performing the full delivery of $i$ if $i$ is visited for the first time;
- performing the full pickup of $i$ if $\tilde{y}_i = 1$ and $i$ is visited just once, or if $i$ is visited for the second time (in this second case $\tilde{y}_i$ is surely 1).

In the backtrack step the flows are set back to 0. The array *visits* is updated accordingly. When the path returns at the depot, a quick check is performed by means of a simple procedure called allVisited, not reported in explicit, which returns *true* if $visits[i]{=}\tilde{\delta}_i$ for all $i \in N$, and *false* otherwise. The first time that allVisited returns *true*, the value of the flag *isDone* is set to *true* and consequently removeSplit terminates. Note that the complexity of removeSplit grows exponentially with the number of vertices visited twice, because each such vertex may lead to two different recursive calls. This has not been an issue in our tests, but, in case of very large instances,

**Algorithm 1** remove split deliveries or pickups from a TCNE solution.

```
1: function REMOVESPLIT(current, dLoad, pLoad, visits)
2:     if current = 0 and visits[0] ≠ 0 then
3:         if ALLVISITED(visits) then return true
4:         else return false
5:     end if
6:     for each i ∈ N : x̃_{current,i} > 0 do
7:         if visits[i]=0 then
8:             dLoad ← dLoad − d_i
9:             if (ỹ_i = 1 and δ̃_i = 1) then pLoad ← pLoad + p_i
10:        else
11:            pLoad ← pLoad + p_i
12:        end if
13:        f^d_{current,i} ← dLoad;
14:        f^p_{current,i} ← pLoad
15:        visits[i] ← visits[i] + 1
16:        isDone ← REMOVESPLIT(i, dLoad, pLoad, visits)                ▷ forward
17:        if isDone then
18:            return true
19:        else                                                         ▷ backtrack
20:            f^d_{current,i} ← 0;
21:            f^p_{current,i} ← 0
22:            visits[i] ← visits[i] − 1
23:        end if
24:    end for
25:    return false
26: end function
```

one could alternatively remove splits by means of the procedure outlined in the next section, based on a MILP formulation.

### 10.3. Procedure for detecting intermediate dropoffs

The exact algorithms that we implemented for the CD case are based on solutions of TINE formulation, and eventually need to detect if these solutions require dropoffs. This can be achieved by an easy adaptation of Algorithm 1 presented in the previous section, or through a MILP based procedure. The adaptation of Algorithm 1 still has a complexity that grows exponentially with the number of customers visited twice. Here we focus on the description of the MILP based procedure, called CHECKDROPOFF, that is more convenient for large size instances.

In particular, let $(\bar{x}, \bar{y})$ be a solution of TINE formulation (i.e., minimize (12) subject to (13)–(17), (22)–(24), and (26)–(35)), and recall that dropoffs may only happen at customers visited twice. The idea of CheckDropoff is to determine which customers are visited twice in $(\bar{x}, \bar{y})$, create a new graph in which these customers (and only these customers) are duplicated, and then solve on this graph a version of TCEE formulation (i.e., minimize (1) subject to (2)-(11)) that embeds the information from $(\bar{x}, \bar{y})$. The outcome is a feasible solution, if any, with $f^p$ and $f^d$ commodity flows requiring no dropoffs, or a proof of infeasibility of $(\bar{x}, \bar{y})$.

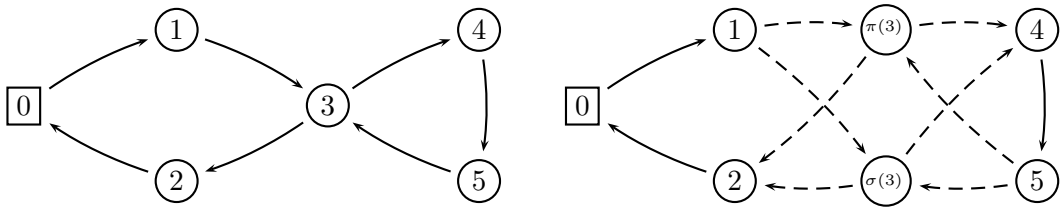
Let $\overline{PD}_2$ be the set of customers visited twice in $(\bar{x}, \bar{y})$ and $\overline{PD}_1 = PD \setminus \overline{PD}_2$ the set of those visited once. We create a new graph $\widetilde{G} = \{\widetilde{V}, \widetilde{A}\}$, by applying the duplication in Definition 1 to all $i \in \overline{PD}_2$, and by setting all customers $i \in \overline{PD}_1$ as pure delivery customers with demand $d_i - p_i \bar{y}_i$. We then embed the information from $(\bar{x}, \bar{y})$ by setting $y_i = \bar{y}_i$ for all $i \in P$, and imposing on each arc $(i, j)$ the following constraints:

- $x_{ij} = \bar{x}_{ij}$ if $i, j \in \widetilde{V} \setminus \overline{PD}_2$;
- $x_{\pi(i),j} + x_{\sigma(i),j} = \bar{x}_{ij}$ if $i \in \overline{PD}_2, j \in \widetilde{V} \setminus \overline{PD}_2$;
- $x_{i,\pi(j)} + x_{i,\sigma(j)} = \bar{x}_{ij}$ if $i \in \widetilde{V} \setminus \overline{PD}_2, j \in \overline{PD}_2$;
- $x_{\pi(i),\pi(j)} + x_{\pi(i),\sigma(j)} + x_{\sigma(i),\pi(j)} + x_{\sigma(i),\sigma(j)} = \bar{x}_{ij}$ if $i, j \in \overline{PD}_2$.

By doing this we require TCEE to follow the tour induced by $\bar{x}_{ij}$, but let it find the best way to visit and serve the customers in $\overline{PD}_2$. If TCEE finds a feasible solution on $\widetilde{G}$, then CheckDropoff easily maps the resulting $f^p$ and $f^d$ flows into the original graph and returns the feasible solution. Otherwise, it returns the proof of infeasibility.

A simple example of the way $\widetilde{G}$ is created is given in Figure 7. Figure 7-(a) represents the original $(\bar{x}, \bar{y})$ solution, where any arc $(i, j)$ in solid lines indicate that $\bar{x}_{ij} = 1$. We then duplicate customer 3 creating $\pi(3)$ and $\sigma(3)$. Figure 7-(b) depicts the resulting graph $\widetilde{G}$, again omitting arcs that are not used in the solution. Notice that dashed lines represent new arcs, whose associated variable values will be fixed by TCEE, whereas the values of $x_{01}$, $x_{45}$, and $x_{20}$ are set to 1.

It is interesting to notice that this procedure can be applied to find the values of flow variables $f^d$ and $f^p$ that do not require split deliveries or pickups, so it is also an alternative to Algorithm 1 outlined in the previous section.




(a) Original non-Elementary solution (b) Supporting graph $\widetilde{G}$

**Figure 7 Creation of the supporting graph used to detect dropoffs (customer 3 is duplicated, creating $\pi(3)$ and $\sigma(3)$, and new arcs are added).**

### 10.4. Details of the Benders Decomposition

In this section we give the details of the primal subproblem (PSP) and the dual subproblem (PSP) used in the Benders decomposition of Section 5. Recall that $V = \{0\} \cup P \cup D \cup PD$. Given a solution $(\bar{x}, \bar{y})$ to the master problem (minimize (12) subject to (13)–(17) and (22)–(24)), PSP can be modeled by

$$\text{(PSP)} \quad \min\ 0 \tag{48}$$

subject to:

$$f^d_{ij} + f^p_{ij} \le Q\bar{x}_{ij} \qquad \forall\ (i,j) \in A, \tag{49}$$

$$\sum_{i \in V \setminus \{j\}} \left(f^d_{ij} - f^d_{ji}\right) = d_j \qquad \forall\ j \in V \setminus \{0\}, \tag{50}$$

$$\sum_{i \in V \setminus \{j\}} \left(f^p_{ji} - f^p_{ij}\right) = p_j \bar{y}_j \qquad \forall\ j \in P \cup PD, \tag{51}$$

$$\sum_{i \in V \setminus \{j\}} \left(f^p_{ji} - f^p_{ij}\right) = 0 \qquad \forall\ j \in D, \tag{52}$$

$$f^d_{ij}, f^p_{ij} \ge 0 \qquad \forall\ (i,j) \in A. \tag{53}$$

The corresponding DSP can be modeled by associating variables $t_{ij}$ with constraints (49) for $(i,j) \in A$, $v_j$ with (50) for $j \in V \setminus \{0\}$, $w_j$ with (51) for $j \in P \cup PD$ and with (52) for $j \in D$, so obtaining:

$$\text{(DSP)} \quad \max \sum_{(i,j) \in A} (Q\bar{x}_{ij}) t_{ij} + \sum_{j \in V \setminus \{0\}} d_j v_j + \sum_{j \in P \cup PD} (p_j \bar{y}_j) w_j \tag{54}$$

subject to:

$$t_{ij} - v_i + v_j \leq 0 \qquad \forall\ i,j \in V \setminus \{0\}, i \neq j, \tag{55}$$

$$t_{ij} - w_j + w_i \leq 0 \qquad \forall\ i,j \in V \setminus \{0\}, i \neq j, \tag{56}$$

$$t_{0i} + v_i \leq 0 \qquad \forall\ i \in V \setminus \{0\}, \tag{57}$$

$$t_{i0} - v_i \leq 0 \qquad \forall\ i \in V \setminus \{0\}, \tag{58}$$

$$t_{0i} - w_i \leq 0 \qquad \forall\ i \in V \setminus \{0\}, \tag{59}$$

$$t_{i0} + w_i \leq 0 \qquad \forall\ i \in V \setminus \{0\}, \tag{60}$$

$$t_{ij} \leq 0 \qquad \forall\ (i,j) \in A, \tag{61}$$

$$v_i, w_i \geqslant 0 \qquad \forall\ i \in V \setminus \{0\}. \tag{62}$$

## 11. Computational Comparison with existing metaheuristic algorithms

In this section we compare the results of our best exact algorithm (MEN) with some efficient metaheuristic algorithms. We selected for the comparison the most effective algorithms in the literature, that are, according to our knowledge, the *general variable neighborhood search* (GVNS) by Coelho et al. (2012), and the *evolutionary algorithm* (EA) by Bruck et al. (2012). Both algorithms were run only on the GLS set and executed several times with different random seeds. GVNS was run 30 times, each with a limited number of iterations on an Intel i7 2.93GHz with 8Gb of RAM, and EA 10 times, each with a time limit of 3 hours on an Intel i7 3.07GHz with 6Gb of RAM.

The results of our comparison are given in Table 9. Columns $\text{gap}_{best}$ and $\text{gap}_{avg}$ give, respectively, the average percentage gaps between the best and average solution values found by the metaheuristic and the optimal solution value. Similarly, column gap gives the average percentage gap between the best upper bound found by MEM and the optimal solution value. The gaps of all algorithms are evaluated by using the objective function that minimizes the sum of the costs plus the sum of the lost revenues of those pickups that were not performed, as in Equation (1).

From the results we can clearly see that MEN greatly outperforms the heuristics in terms of number of optimal solutions found, time consumption, and optimality gaps. This is another strong evidence of its efficiency in practice.

**Table 9 Comparison between MEN and the best metaheuristics from the SVRPDSP literature.**

| | | EA | | | | GVNS | | | | MEN | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GLS set | # | opt | sec | $gap_{best}$ | $gap_{avg}$ | opt | sec | $gap_{best}$ | $gap_{avg}$ | opt | sec | gap |
| $15 \leq n \leq 30$ | 28 | 14 | 518.6 | 0.3 | 1.3 | 20 | 22.3 | 1.0 | 2.9 | 28 | 0.2 | 0.0 |
| $32 \leq n \leq 50$ | 24 | 1 | 5138.2 | 0.8 | 1.2 | 10 | 109.0 | 0.5 | 1.7 | 24 | 6.5 | 0.0 |
| $71 \leq n \leq 100$ | 16 | 0 | 22117.3 | 11.2 | 12.7 | 5 | 1156.4 | 1.4 | 3.3 | 15 | 416.6 | 0.0 |
| totals | 68 | 15 | | | | 35 | | | | 67 | | |
| averages | | | 7231.1 | 3.0 | 3.9 | | 319.7 | 0.9 | 2.5 | | 100.4 | 0.0 |

## 12. Detailed Computational Results

Due to the large number of tests, in Section 7 of the paper we presented only aggregate results. Here we give instead detailed results for each run of our algorithms on each instance. All tests had a time limit of 1 CPU hour on a PC equipped with an Intel Core i7-3770 3.40 GHz with 8 Gb of RAM, with the exception of the GMO branch-and-cut, that was run on a Dual Core AMD 2.7 GHz, with 21000 seconds of time limit. The columns in the tables have the following meanings:

- Column *instance* gives the name of the instance;
- Column $z_{opt}$ reports the optimal solution value, if known;
- For a given algorithm $A$, producing a lower bound $L_A$ and an upper bound $U_A$, column *gap* gives the percentage gap of $A$ computed as $100(U_A - L_A)/U_A$. These values are evaluated for all algorithms by using the objective function that minimizes the sum of the costs plus the sum of the lost revenues of those pickups that were not performed, as in Equation (1). The symbol '−' means that the gap is 0 and the algorithm has found a proven optimal solution;
- Column *sec* reports the number of seconds required by the algorithm to run to completion, 't.lim.' indicates that the algorithm reached the time limit, and 'm.lim.' that it stopped because of memory limit; (for the results on the instances that refer to Table 1, that are very easy, *sec* has two digits, whereas for all other tests it has one digit as in the paper);
- The additional columns in Tables 19, 20, and 21 have the same meaning as those reported in Table 3 in the paper, but refer to single instances instead of aggregate results. Namely: columns $gap_r$ and $gap_c$ give, respectively, the percentage gaps between the optimal solution value and the

lower bounds at the root node before and after adding the cuts of Section 5.1; column feas reports if the solution found for SVRPDSP-D is also feasible for SVRPDSP (feas=1) or not (feas=0); column $\text{gap}_d$ gives the percentage gap between the optimal SVRPDSP-D and SVRPDSP solution values; columns iter and dupl give, respectively, the number of customers that were duplicated and of iterations that were performed by MEN;

- For the large size instances in Tables 25 and 26, column $UB_{best}$ reports the best known upper bound (which is equal to $z_{opt}$ when the gap is 0).

### 12.1. Detailed results for SVRPDSP

**Table 10 Detailed results for Table 1: SB set, $n$=10 ($n = |P| + |D|$).**

| | | | | SB | | GMO | | TCNE | | BBNE | | TINE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|P\|$ | $\|D\|$ | $z_{opt}$ | gap | sec | gap | sec | gap | sec | gap | sec | gap | sec |
| E8_1_10_20 | 2 | 8 | 166 | — | 0.01 | — | 0.02 | — | 0.05 | — | 0.00 | — | 0.00 |
| E8_1_10_30 | 3 | 7 | 166 | — | 0.02 | — | 0.02 | — | 0.04 | — | 0.00 | — | 0.00 |
| E8_1_10_40 | 4 | 6 | 166 | — | 0.01 | — | 0.02 | — | 0.06 | — | 0.01 | — | 0.00 |
| E8_31_40_20 | 2 | 8 | 233 | — | 0.03 | — | 0.03 | — | 0.08 | — | 0.00 | — | 0.00 |
| E8_31_40_30 | 3 | 7 | 233 | — | 0.02 | — | 0.03 | — | 0.09 | — | 0.01 | — | 0.00 |
| E8_31_40_40 | 4 | 6 | 233 | — | 0.03 | — | 0.03 | — | 0.09 | — | 0.01 | — | 0.00 |
| E10_1_10_20 | 2 | 8 | 173 | — | 0.00 | — | 0.02 | — | 0.04 | — | 0.00 | — | 0.00 |
| E10_1_10_30 | 3 | 7 | 173 | — | 0.00 | — | 0.02 | — | 0.06 | — | 0.00 | — | 0.00 |
| E10_1_10_40 | 4 | 6 | 173 | — | 0.01 | — | 0.02 | — | 0.03 | — | 0.01 | — | 0.00 |
| E10_81_90_20 | 2 | 8 | 167 | — | 0.00 | — | 0.02 | — | 0.03 | — | 0.00 | — | 0.00 |
| E10_81_90_30 | 3 | 7 | 167 | — | 0.00 | — | 0.02 | — | 0.07 | — | 0.01 | — | 0.00 |
| E10_81_90_40 | 4 | 6 | 167 | — | 0.00 | — | 0.03 | — | 0.03 | — | 0.00 | — | 0.00 |
| F10_1_10_20 | 2 | 8 | 208 | — | 0.08 | — | 0.04 | — | 0.09 | — | 0.02 | — | 0.00 |
| F10_1_10_30 | 3 | 7 | 258 | — | 0.08 | — | 0.11 | — | 0.15 | — | 0.02 | — | 0.01 |
| F10_1_10_40 | 4 | 6 | 296 | — | 0.07 | — | 0.22 | — | 0.13 | — | 0.03 | — | 0.00 |
| F12_1_10_20 | 2 | 8 | 92 | — | 3.77 | — | 0.03 | — | 0.05 | — | 0.01 | — | 0.00 |
| F12_1_10_30 | 3 | 7 | 92 | — | 3.64 | — | 0.03 | — | 0.07 | — | 0.01 | — | 0.00 |
| F12_1_10_40 | 4 | 6 | 92 | — | 3.20 | — | 0.03 | — | 0.08 | — | 0.01 | — | 0.00 |
| F12_81_90_20 | 2 | 8 | 242 | — | 0.10 | — | 0.04 | — | 0.07 | — | 0.01 | — | 0.00 |
| F12_81_90_30 | 3 | 7 | 242 | — | 0.08 | — | 0.04 | — | 0.09 | — | 0.01 | — | 0.00 |
| F12_81_90_40 | 4 | 6 | 242 | — | 0.10 | — | 0.03 | — | 0.11 | — | 0.01 | — | 0.00 |
| M1_10_20 | 2 | 8 | 3154 | — | 0.03 | — | 0.04 | — | 0.09 | — | 0.01 | — | 0.00 |
| M1_10_30 | 3 | 7 | 3154 | — | 0.03 | — | 0.03 | — | 0.12 | — | 0.01 | — | 0.00 |
| M1_10_40 | 4 | 6 | 3154 | — | 0.04 | — | 0.03 | — | 0.09 | — | 0.01 | — | 0.00 |

**Table 11 Detailed results for Table 1: SB set, $n$=20 ($n = |P| + |D|$).**

| | | | | SB | | GMO | | TCNE | | BBNE | | TINE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|P\|$ | $\|D\|$ | $z_{opt}$ | gap | sec | gap | sec | gap | sec | gap | sec | gap | sec |
| E8_11_30_20 | 4 | 16 | 254 | — | 0.01 | — | 0.03 | — | 0.21 | — | 0.01 | — | 0.01 |
| E8_11_30_30 | 6 | 14 | 254 | — | 0.01 | — | 0.03 | — | 0.26 | — | 0.03 | — | 0.01 |
| E8_11_30_40 | 8 | 12 | 254 | — | 0.01 | — | 0.03 | — | 0.17 | — | 0.01 | — | 0.01 |
| E10_11_30_20 | 4 | 16 | 272 | — | 0.12 | — | 0.04 | — | 0.22 | — | 0.04 | — | 0.02 |
| E10_11_30_30 | 6 | 14 | 272 | — | 0.10 | — | 0.04 | — | 0.38 | — | 0.02 | — | 0.01 |
| E10_11_30_40 | 8 | 12 | 272 | — | 0.14 | — | 0.05 | — | 0.27 | — | 0.04 | — | 0.01 |
| E10_41_60_20 | 4 | 16 | 267 | — | 0.24 | — | 0.48 | — | 0.55 | — | 0.11 | — | 0.02 |
| E10_41_60_30 | 6 | 14 | 267 | — | 0.26 | — | 0.47 | — | 0.43 | — | 0.21 | — | 0.01 |
| E10_41_60_40 | 8 | 12 | 267 | — | 0.22 | — | 0.33 | — | 0.43 | — | 0.13 | — | 0.03 |
| F10_11_30_20 | 4 | 16 | 531 | — | 6.17 | — | 0.14 | — | 0.77 | — | 1.53 | — | 0.03 |
| F10_11_30_30 | 6 | 14 | 531 | — | 10.00 | — | 0.14 | — | 0.78 | — | 0.83 | — | 0.03 |
| F10_11_30_40 | 8 | 12 | 582 | — | 21.61 | — | 0.59 | — | 2.00 | — | 2.31 | — | 0.08 |
| F12_11_30_20 | 4 | 16 | 156 | — | 1.23 | — | 0.22 | — | 0.63 | — | 0.46 | — | 0.04 |
| F12_11_30_30 | 6 | 14 | 156 | — | 0.60 | — | 0.19 | — | 0.82 | — | 0.86 | — | 0.03 |
| F12_11_30_40 | 8 | 12 | 156 | — | 0.78 | — | 0.21 | — | 0.54 | — | 1.01 | — | 0.03 |
| F12_41_60_20 | 4 | 16 | 86 | 13.79 | t.lim. | — | 0.09 | — | 0.32 | — | 0.25 | — | 0.03 |
| F12_41_60_30 | 6 | 14 | 86 | 13.92 | t.lim. | — | 0.09 | — | 0.47 | — | 0.81 | — | 0.02 |
| F12_41_60_40 | 8 | 12 | 86 | 12.80 | t.lim. | — | 0.17 | — | 0.33 | — | 0.65 | — | 0.03 |
| M5_24_20 | 4 | 16 | 4263 | — | 0.28 | — | 0.10 | — | 0.37 | — | 0.09 | — | 0.01 |
| M5_24_30 | 6 | 14 | 4263 | — | 0.22 | — | 0.10 | — | 0.33 | — | 0.21 | — | 0.02 |
| M5_24_40 | 8 | 12 | 4263 | — | 0.15 | — | 0.10 | — | 0.30 | — | 0.11 | — | 0.01 |

**Table 12 Detailed results for Table 1: SB set, $n$=30 ($n = |P| + |D|$).**

| | | | | SB | | GMO | | TCNE | | BBNE | | TINE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|P\|$ | $\|D\|$ | $z_{opt}$ | gap | sec | gap | sec | gap | sec | gap | sec | gap | sec |
| E8_21_50_20 | 6 | 24 | 341 | — | 0.55 | — | 1.92 | — | 1.31 | — | 3.63 | — | 0.14 |
| E8_21_50_30 | 9 | 21 | 341 | — | 1.45 | — | 2.29 | — | 1.48 | — | 2.58 | — | 0.08 |
| E8_21_50_40 | 12 | 18 | 341 | — | 0.71 | — | 1.67 | — | 1.67 | — | 2.60 | — | 0.09 |
| E10_21_50_20 | 6 | 24 | 351 | — | 2.82 | — | 0.49 | — | 2.01 | — | 1.96 | — | 0.09 |
| E10_21_50_30 | 9 | 21 | 351 | — | 1.31 | — | 0.18 | — | 1.53 | — | 1.48 | — | 0.03 |
| E10_21_50_40 | 12 | 18 | 351 | — | 2.06 | — | 0.18 | — | 1.78 | — | 2.61 | — | 0.03 |
| E10_51_80_20 | 6 | 24 | 318 | — | 1.06 | — | 0.35 | — | 1.08 | — | 1.03 | — | 0.04 |
| E10_51_80_30 | 9 | 21 | 318 | — | 2.16 | — | 0.35 | — | 0.75 | — | 1.11 | — | 0.07 |
| E10_51_80_40 | 12 | 18 | 318 | — | 1.25 | — | 0.49 | — | 1.82 | — | 1.04 | — | 0.07 |
| F10_15_44_20 | 6 | 24 | 519 | — | 2839.29 | — | 0.15 | — | 8.03 | — | 2.29 | — | 0.03 |
| F10_15_44_30 | 9 | 21 | 519 | — | 2319.88 | — | 0.14 | — | 6.64 | — | 1.72 | — | 0.04 |
| F10_15_44_40 | 12 | 18 | 519 | — | 2210.41 | — | 0.15 | — | 3.35 | — | 3.68 | — | 0.06 |
| F12_21_50_20 | 6 | 24 | 154 | — | 1.01 | — | 0.12 | — | 0.64 | — | 2.78 | — | 0.08 |
| F12_21_50_30 | 9 | 21 | 154 | — | 0.62 | — | 0.16 | — | 0.83 | — | 0.57 | — | 0.06 |
| F12_21_50_40 | 12 | 18 | 154 | — | 0.67 | — | 0.16 | — | 0.87 | — | 1.20 | — | 0.03 |
| F12_51_80_20 | 6 | 24 | 244 | — | 87.58 | — | 0.19 | — | 2.64 | — | 3.96 | — | 0.07 |
| F12_51_80_30 | 9 | 21 | 244 | — | 109.38 | — | 0.25 | — | 2.29 | — | 3.26 | — | 0.06 |
| F12_51_80_40 | 12 | 18 | 249 | 1.86 | t.lim. | — | 0.77 | — | 3.02 | — | 18.78 | — | 0.12 |

**Table 13 Detailed results for Table 1: GMO set,** $25 \leq n \leq 30$ **(**$n = |P| + |D|$**).**

| | | | | SB | | GMO | | TCNE | | BBNE | | TINE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|P\|$ | $\|D\|$ | $z_{opt}$ | gap | sec | gap | sec | gap | sec | gap | sec | gap | sec |
| A_30_L_400 | 8 | 17 | 11376373 | — | 0.74 | — | 1 | — | 1.03 | — | 1.66 | — | 0.06 |
| A_30_L_500 | 8 | 17 | 11458360 | — | 0.99 | — | 1 | — | 0.83 | — | 1.39 | — | 0.05 |
| A_30_L_1700 | 8 | 17 | 11936711 | — | 0.67 | — | 1 | — | 0.72 | — | 0.49 | — | 0.02 |
| A_50_L_500 | 13 | 12 | 10886313 | — | 2.44 | — | 1 | — | 0.82 | — | 1.89 | — | 0.03 |
| A_50_L_1000 | 13 | 12 | 11558540 | — | 2.20 | — | 1 | — | 0.86 | — | 2.27 | — | 0.05 |
| A_50_L_1700 | 13 | 12 | 12138754 | — | 4.56 | — | 1 | — | 0.92 | — | 5.76 | — | 0.11 |
| A_O_L_200 | 5 | 20 | 11193089 | — | 0.83 | — | 1 | — | 0.91 | — | 0.86 | — | 0.05 |
| A_O_L_400 | 5 | 20 | 11444625 | — | 0.79 | — | 1 | — | 1.00 | — | 1.12 | — | 0.01 |
| A_O_L_1700 | 5 | 20 | 11936711 | — | 0.56 | — | 1 | — | 0.65 | — | 1.13 | — | 0.06 |
| B_30_L_100 | 9 | 21 | 11598071 | — | 7.80 | — | 1 | — | 1.70 | — | 10.18 | — | 0.07 |
| B_30_L_700 | 9 | 21 | 12403680 | — | 3.15 | — | 3 | — | 2.64 | — | 2.70 | — | 0.27 |
| B_30_L_1000 | 9 | 21 | 12437396 | — | 1.51 | — | 3 | — | 1.63 | — | 2.25 | — | 0.24 |
| B_50_L_100 | 15 | 15 | 11514188 | — | 166.50 | — | 1 | — | 1.77 | — | 14.89 | — | 0.18 |
| B_50_L_700 | 15 | 15 | 12384340 | — | 14.49 | — | 4 | — | 2.99 | — | 12.04 | — | 0.37 |
| B_50_L_1000 | 15 | 15 | 12617740 | — | 19.41 | — | 7 | — | 2.83 | — | 19.19 | — | 0.27 |
| B_O_L_100 | 10 | 20 | 11598071 | — | 15.14 | — | 1 | — | 2.62 | — | 4.70 | — | 0.04 |
| B_O_L_700 | 10 | 20 | 12403680 | — | 3.57 | — | 4 | — | 1.94 | — | 2.05 | — | 0.30 |
| B_O_L_1000 | 10 | 20 | 12437396 | — | 3.19 | — | 7 | — | 1.54 | — | 1.88 | — | 0.10 |

**Table 14 Detailed results for Table 1: GMO set, 38 $\leq n \leq$ 60 ($n = |P| + |D|$).**

| | | | | SB | | GMO | | TCNE | | BBNE | | TINE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|P\|$ | $\|D\|$ | $z_{opt}$ | gap | sec | gap | sec | gap | sec | gap | sec | gap | sec |
| C_30_L_100 | 12 | 28 | 12954837 | — | 5.51 | — | 2 | — | 4.34 | — | 16.50 | — | 0.13 |
| C_30_L_200 | 12 | 28 | 13419793 | — | 7.11 | — | 1 | — | 4.26 | — | 13.97 | — | 0.21 |
| C_30_L_500 | 12 | 28 | 14062470 | — | 10.35 | — | 2 | — | 9.24 | — | 7.14 | — | 0.40 |
| C_50_O_L_100 | 20 | 20 | 11658312 | — | 2.30 | — | 1 | — | 3.06 | — | 12.87 | — | 0.18 |
| C_50_O_L_200 | 20 | 20 | 12471864 | — | 2.60 | — | 1 | — | 2.38 | — | 12.15 | — | 0.15 |
| C_50_O_L_500 | 20 | 20 | 13814276 | — | 27.56 | — | 1 | — | 10.48 | — | 19.82 | — | 0.14 |
| D_30_L_400 | 11 | 27 | 12084482 | — | 512.81 | — | 1 | — | 1.91 | — | 9.65 | — | 0.08 |
| D_30_L_700 | 11 | 27 | 12400614 | — | 438.44 | — | 1 | — | 2.42 | — | 7.87 | — | 0.15 |
| D_30_L_1200 | 11 | 27 | 12606406 | — | 705.59 | — | 1 | — | 3.31 | — | 9.55 | — | 0.19 |
| D_50_L_400 | 19 | 19 | 12290137 | 3.92 | t.lim. | — | 6 | — | 6.20 | — | 38.65 | — | 0.41 |
| D_50_L_500 | 19 | 19 | 12480072 | 4.14 | t.lim. | — | 12 | — | 3.58 | — | 74.13 | — | 0.35 |
| D_50_L_700 | 19 | 19 | 12706204 | 3.72 | t.lim. | — | 15 | — | 10.09 | — | 99.21 | — | 0.39 |
| D_O_L_500 | 8 | 30 | 12239682 | — | 610.17 | — | 1 | — | 2.58 | — | 8.90 | — | 0.15 |
| D_O_L_1000 | 8 | 30 | 12579714 | — | 810.94 | — | 1 | — | 2.84 | — | 7.45 | — | 0.15 |
| D_O_L_1500 | 8 | 30 | 12606406 | — | 413.95 | — | 1 | — | 1.70 | — | 3.73 | — | 0.17 |
| E_30_L_300 | 14 | 31 | 14165760 | — | 785.91 | — | 5 | — | 8.99 | — | 56.41 | — | 0.91 |
| E_30_L_500 | 14 | 31 | 14978047 | — | 816.08 | — | 4 | — | 25.46 | — | 32.45 | — | 0.65 |
| E_30_L_1200 | 14 | 31 | 15607486 | — | 129.83 | — | 5 | — | 27.67 | — | 32.31 | — | 1.94 |
| E_50_L_300 | 23 | 22 | 13219046 | — | 670.92 | — | 2 | — | 21.59 | — | 46.67 | — | 0.81 |
| E_50_L_600 | 23 | 22 | 14912502 | — | 1114.21 | — | 8 | — | 36.32 | — | 138.90 | — | 1.76 |
| E_50_L_1300 | 23 | 22 | 15882369 | — | 2754.89 | — | 226 | — | 63.16 | — | 486.07 | — | 18.00 |
| E_O_L_100 | 15 | 30 | 13262096 | — | 753.10 | — | 5 | — | 17.67 | — | 94.90 | — | 1.09 |
| E_O_L_400 | 15 | 30 | 14604360 | — | 479.45 | — | 7 | — | 9.75 | — | 44.36 | — | 0.70 |
| E_O_L_900 | 15 | 30 | 15512086 | — | 149.00 | — | 9 | — | 30.55 | — | 29.95 | — | 1.00 |
| F_30_L_200 | 18 | 42 | 15426659 | 2.14 | t.lim. | — | 11 | — | 90.45 | — | 1198.49 | — | 2.65 |
| F_30_L_400 | 18 | 42 | 16253023 | 3.23 | t.lim. | — | 119 | — | 96.62 | — | 408.15 | — | 2.99 |
| F_30_L_1700 | 18 | 42 | 17528624 | 4.73 | t.lim. | — | 43 | — | 92.70 | — | 657.11 | — | 2.89 |
| F_50_O_L_200 | 30 | 30 | 15003977 | 3.27 | t.lim. | — | 84 | — | 113.95 | 7.64 | t.lim. | — | 10.48 |
| F_50_O_L_400 | 30 | 30 | 16171570 | 5.15 | t.lim. | — | 521 | — | 139.80 | 5.38 | t.lim. | — | 32.66 |
| F_50_O_L_1000 | 30 | 30 | 17317966 | 2.60 | t.lim. | — | 202 | — | 142.90 | 7.65 | t.lim. | — | 11.70 |
| G_30_L_300 | 17 | 40 | 15758461 | 4.81 | t.lim. | — | 2 | — | 62.44 | — | 425.65 | — | 1.42 |
| G_30_L_500 | 17 | 40 | 15978604 | — | 2492.74 | — | 2 | — | 60.74 | — | 208.40 | — | 0.72 |
| G_30_L_2000 | 17 | 40 | 16523165 | — | 822.75 | — | 36 | — | 60.90 | — | 127.38 | — | 1.63 |
| G_50_L_300 | 29 | 28 | 14800857 | — | 1468.31 | — | 19 | — | 77.05 | — | 1036.83 | — | 2.19 |
| G_50_L_500 | 29 | 28 | 15740523 | 2.05 | t.lim. | — | 84 | — | 133.66 | — | 788.14 | — | 6.84 |
| G_50_L_1000 | 29 | 28 | 16545438 | 1.76 | t.lim. | — | 234 | — | 135.07 | — | 583.93 | — | 29.05 |
| G_O_L_300 | 12 | 45 | 16047591 | 5.14 | t.lim. | — | 24 | — | 59.22 | — | 291.54 | — | 1.35 |
| G_O_L_500 | 12 | 45 | 16394865 | — | 1577.59 | — | 79 | — | 79.45 | — | 247.52 | — | 1.26 |
| G_O_L_1000 | 12 | 45 | 16523165 | — | 1621.00 | — | 56 | — | 65.14 | — | 113.80 | — | 2.38 |

**Table 15 Detailed results for Table 1: GMO set, $68 \leq n \leq 90$ ($n = |P| + |D|$).**

| | | | | SB | | GMO | | TCNE | | BBNE | | TINE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|P\|$ | $\|D\|$ | $z_{opt}$ | gap | sec | gap | sec | gap | sec | gap | sec | gap | sec |
| H_30_L_300 | 20 | 48 | 17147339 | 0.49 | t.lim. | — | 5 | — | 168.63 | — | 515.32 | — | 1.32 |
| H_30_L_500 | 20 | 48 | 17327696 | — | 1752.99 | — | 7 | — | 133.67 | — | 247.99 | — | 1.61 |
| H_30_L_1000 | 20 | 48 | 17480958 | — | 1636.56 | — | 10 | — | 52.46 | — | 352.19 | — | 0.83 |
| H_50_L_200 | 34 | 34 | 16362687 | 5.89 | t.lim. | — | 29 | — | 215.50 | 5.69 | t.lim. | — | 7.73 |
| H_50_L_300 | 34 | 34 | 16843003 | 3.77 | t.lim. | — | 18 | — | 175.23 | 0.20 | t.lim. | — | 2.17 |
| H_50_L_400 | 34 | 34 | 17140925 | 2.07 | t.lim. | — | 72 | — | 193.27 | 3.60 | t.lim. | — | 5.67 |
| H_50_L_600 | 34 | 34 | 17489325 | 2.21 | t.lim. | — | 662 | — | 322.38 | 3.60 | t.lim. | — | 74.05 |
| H_50_L_700 | 34 | 34 | 17663525 | 2.65 | t.lim. | — | 13855 | — | 352.17 | 4.42 | t.lim. | — | 55.04 |
| H_O_L_300 | 23 | 45 | 17062720 | 2.01 | t.lim. | — | 1 | — | 122.93 | — | 1122.47 | — | 0.86 |
| H_O_L_500 | 23 | 45 | 17327696 | 0.37 | t.lim. | — | 4 | — | 113.91 | — | 580.77 | — | 1.23 |
| H_O_L_1000 | 23 | 45 | 17480958 | 0.76 | t.lim. | — | 13 | — | 142.20 | — | 278.36 | — | 0.93 |
| I_30_L_100 | 27 | 63 | 18132091 | 2.66 | t.lim. | — | 169 | — | 588.17 | 6.67 | t.lim. | — | 9.45 |
| I_30_L_200 | 27 | 63 | 18721168 | 2.98 | t.lim. | — | 261 | — | 319.47 | 5.97 | t.lim. | — | 30.89 |
| I_30_L_300 | 27 | 63 | 19159066 | 2.99 | t.lim. | — | 3150 | — | 342.42 | 7.14 | t.lim. | — | 29.05 |
| I_50_O_L_100 | 45 | 45 | 17112971 | 6.44 | t.lim. | — | 1546 | — | 718.07 | 12.13 | t.lim. | — | 149.22 |
| I_50_O_L_200 | 45 | 45 | 18218462 | 5.38 | t.lim. | — | 18880 | — | 584.34 | 11.48 | t.lim. | — | 120.90 |
| I_50_O_L_300 | 45 | 45 | 18989572 | 4.94 | t.lim. | — | 21001 | — | 803.12 | 11.53 | t.lim. | — | 143.10 |

**Table 16 Detailed results for Table 2: GLS set, $15 \leq n \leq 30$ ($n = |PD|$).**

| | | | GLS | | SB | | TCEE | | TA | | 2S | | MEN | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | $z_{opt}$ | gap | sec | gap | sec | gap | sec | gap | sec | gap | sec | gap | sec |
| 016_B_p_two | 15 | 221.00 | — | 423.4 | — | 377.4 | — | 4.9 | — | 0.1 | — | 0.1 | — | 0.0 |
| 016_B_half | 15 | 243.07 | 7.8 | t.lim. | — | 575.1 | — | 12.1 | — | 0.0 | — | 0.1 | — | 0.0 |
| 016_B_one | 15 | 262.20 | 16.1 | t.lim. | — | 916.0 | — | 13.4 | — | 0.2 | — | 0.3 | — | 0.0 |
| 016_B_two | 15 | 289.73 | 23.6 | t.lim. | — | 832.9 | — | 14.5 | — | 0.1 | — | 0.3 | — | 0.0 |
| 021_B_p_two | 20 | 265.04 | 0.5 | t.lim. | 1.8 | t.lim. | — | 18.0 | — | 0.1 | — | 0.0 | — | 0.0 |
| 021_B_half | 20 | 292.18 | 9.3 | t.lim. | — | 87.8 | — | 19.3 | — | 0.0 | — | 0.2 | — | 0.0 |
| 021_B_one | 20 | 316.89 | 17.5 | t.lim. | 2.6 | t.lim. | — | 39.3 | — | 0.0 | — | 0.1 | — | 0.0 |
| 021_B_two | 20 | 348.13 | 24.6 | t.lim. | 2.4 | t.lim. | — | 24.0 | — | 0.1 | — | 0.1 | — | 0.0 |
| 022_B_p_two | 21 | 287.91 | 8.7 | t.lim. | 6.6 | t.lim. | — | 470.2 | — | 1.6 | — | 0.5 | — | 0.2 |
| 022_B_half | 21 | 319.76 | 22.0 | t.lim. | 9.5 | t.lim. | — | 314.7 | — | 0.6 | — | 0.9 | — | 0.1 |
| 022_B_one | 21 | 343.74 | 26.8 | t.lim. | 4.4 | t.lim. | — | 580.2 | — | 0.9 | — | 0.7 | — | 0.2 |
| 022_B_two | 21 | 380.16 | 33.2 | t.lim. | 4.1 | t.lim. | — | 253.7 | — | 0.4 | — | 0.2 | — | 0.2 |
| 023_B_p_two | 22 | 527.71 | 20.1 | t.lim. | 11.2 | t.lim. | — | 564.6 | — | 0.8 | — | 0.7 | — | 0.5 |
| 023_B_half | 22 | 565.80 | 20.5 | t.lim. | 7.5 | t.lim. | — | 80.7 | — | 0.1 | — | 0.4 | — | 0.1 |
| 023_B_one | 22 | 595.20 | 32.0 | t.lim. | 7.0 | t.lim. | — | 63.0 | — | 0.1 | — | 0.3 | — | 0.1 |
| 023_B_two | 22 | 653.99 | 55.3 | t.lim. | 5.0 | t.lim. | — | 37.2 | — | 0.2 | — | 0.3 | — | 0.1 |
| 026_B_p_two | 25 | 319.54 | 10.7 | t.lim. | 10.2 | t.lim. | — | 117.7 | — | 0.6 | — | 0.6 | — | 0.1 |
| 026_B_half | 25 | 343.58 | 16.6 | t.lim. | 10.5 | t.lim. | — | 534.7 | — | 0.6 | — | 0.2 | — | 0.2 |
| 026_B_one | 25 | 374.79 | 30.1 | t.lim. | 6.1 | t.lim. | — | 288.8 | — | 4.0 | — | 2.0 | — | 0.3 |
| 026_B_two | 25 | 409.67 | 33.7 | t.lim. | 5.9 | t.lim. | — | 871.7 | — | 0.2 | — | 0.6 | — | 0.3 |
| 030_B_p_two | 29 | 402.70 | 25.6 | t.lim. | 22.7 | t.lim. | 5.1 | t.lim. | — | 1.0 | — | 0.7 | — | 0.3 |
| 030_B_half | 29 | 422.47 | 31.9 | t.lim. | 19.8 | t.lim. | 3.8 | t.lim. | — | 0.8 | — | 0.5 | — | 0.3 |
| 030_B_one | 29 | 450.10 | 39.1 | t.lim. | 18.9 | t.lim. | 3.1 | t.lim. | — | 3.2 | — | 0.7 | — | 0.3 |
| 030_B_two | 29 | 505.34 | 48.3 | t.lim. | 16.2 | t.lim. | 3.1 | t.lim. | — | 10.8 | — | 0.3 | — | 0.2 |
| 031_B_p_two | 30 | 326.60 | 6.3 | t.lim. | 4.1 | t.lim. | — | 288.3 | — | 2.8 | — | 1.8 | — | 0.5 |
| 031_B_half | 30 | 361.23 | 20.7 | t.lim. | 2.7 | t.lim. | — | 1256.8 | — | 1.3 | — | 1.3 | — | 0.4 |
| 031_B_one | 30 | 393.78 | 26.9 | t.lim. | 3.0 | t.lim. | — | 1750.9 | — | 1.0 | — | 1.4 | — | 0.8 |
| 031_B_two | 30 | 454.10 | 46.6 | t.lim. | 2.4 | t.lim. | 0.4 | t.lim. | — | 15.1 | — | 2.5 | — | 0.5 |

**Table 17 Detailed results for Table 2: GLS set, $32 \leq n \leq 50$ ($n = |PD|$).**

| | | | GLS | | SB | | TCEE | | TA | | 2S | | MEN | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | $z_{opt}$ | gap | sec | gap | sec | gap | sec | gap | sec | gap | sec | gap | sec |
| 033_B_p_two | 32 | 459.03 | 24.0 | t.lim. | 19.1 | t.lim. | 3.4 | t.lim. | — | 2.4 | — | 3.3 | — | 1.6 |
| 033_B_half | 32 | 490.40 | 27.4 | t.lim. | 16.7 | t.lim. | 2.6 | t.lim. | — | 1.1 | — | 1.5 | — | 1.6 |
| 033_B_one | 32 | 516.49 | 30.4 | t.lim. | 17.0 | t.lim. | 2.4 | t.lim. | — | 2.9 | — | 0.7 | — | 0.3 |
| 033_B_two | 32 | 556.56 | 35.5 | t.lim. | 15.6 | t.lim. | 0.7 | t.lim. | — | 5.0 | — | 0.6 | — | 0.6 |
| 036_B_p_two | 35 | 360.39 | 10.3 | t.lim. | 7.8 | t.lim. | 0.2 | t.lim. | — | 2.3 | — | 4.0 | — | 1.8 |
| 036_B_half | 35 | 397.93 | 19.9 | t.lim. | 5.3 | t.lim. | 0.5 | t.lim. | — | 1.0 | — | 0.5 | — | 0.8 |
| 036_B_one | 35 | 428.00 | 36.9 | t.lim. | 3.9 | t.lim. | 0.9 | t.lim. | — | 1.6 | — | 1.2 | — | 0.4 |
| 036_B_two | 35 | 488.13 | 45.8 | t.lim. | 3.2 | t.lim. | — | 3406.7 | — | 1.2 | — | 1.6 | — | 0.6 |
| 041_B_p_two | 40 | 374.70 | 10.0 | t.lim. | 5.7 | t.lim. | 0.2 | t.lim. | — | 315.4 | — | 258.4 | — | 8.4 |
| 041_B_half | 40 | 426.53 | 32.7 | t.lim. | 3.2 | t.lim. | 0.4 | t.lim. | — | 1.0 | — | 2.3 | — | 0.5 |
| 041_B_one | 40 | 463.90 | 34.0 | t.lim. | 3.9 | t.lim. | 1.6 | t.lim. | — | 7.0 | — | 5.6 | — | 4.0 |
| 041_B_two | 40 | 525.12 | 41.5 | t.lim. | 3.7 | t.lim. | 1.6 | t.lim. | — | 16.3 | — | 4.9 | — | 2.8 |
| 045_B_p_two | 44 | 671.79 | 69.4 | t.lim. | 30.7 | t.lim. | 10.0 | t.lim. | — | 0.8 | — | 0.7 | — | 0.4 |
| 045_B_half | 44 | 693.18 | 73.9 | t.lim. | 28.8 | t.lim. | 6.2 | t.lim. | — | 0.3 | — | 0.7 | — | 0.1 |
| 045_B_one | 44 | 720.09 | 44.4 | t.lim. | 27.6 | t.lim. | 3.9 | t.lim. | — | 0.4 | — | 0.5 | — | 0.2 |
| 045_B_two | 44 | 773.92 | 77.5 | t.lim. | 25.3 | t.lim. | 4.8 | t.lim. | — | 0.3 | — | 0.5 | — | 0.1 |
| 048_B_p_two | 47 | 36703.71 | 28.4 | t.lim. | 16.1 | t.lim. | 5.7 | t.lim. | 0.1 | t.lim. | 0.1 | t.lim. | — | 3.2 |
| 048_B_half | 47 | 39983.45 | 33.1 | t.lim. | 16.3 | t.lim. | 4.1 | t.lim. | <0.1 | t.lim. | <0.1 | t.lim. | — | 2.7 |
| 048_B_one | 47 | 47319.14 | 43.2 | t.lim. | 13.8 | t.lim. | 4.1 | t.lim. | — | 25.5 | — | 1.1 | — | 1.3 |
| 048_B_two | 47 | 60457.70 | 56.7 | t.lim. | 10.1 | t.lim. | 2.2 | t.lim. | — | 3.4 | — | 1.2 | — | 0.8 |
| 051_B_p_two | 50 | 447.85 | 28.1 | t.lim. | 10.8 | t.lim. | 2.1 | t.lim. | 0.3 | t.lim. | 0.3 | t.lim. | — | 36.7 |
| 051_B_half | 50 | 498.17 | 25.5 | t.lim. | 10.1 | t.lim. | 3.3 | t.lim. | 0.7 | t.lim. | 0.7 | t.lim. | — | 32.5 |
| 051_B_one | 50 | 530.96 | 29.4 | t.lim. | 8.3 | t.lim. | 3.9 | t.lim. | 0.7 | t.lim. | 0.8 | t.lim. | — | 26.7 |
| 051_B_two | 50 | 595.80 | 60.8 | t.lim. | 7.7 | t.lim. | 4.6 | t.lim. | 0.7 | t.lim. | 0.8 | t.lim. | — | 28.4 |

**Table 18 Detailed results for Table 2: GLS set, $71 \leq n \leq 100$ ($n = |PD|$).**

| | | | GLS | | SB | | TCEE | | TA | | 2S | | MEN | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | $z_{opt}$ | gap | sec | gap | sec | gap | sec | gap | sec | gap | sec | gap | sec |
| 072_B_p_two | 71 | 212.49 | 64.5 | t.lim. | 28.4 | t.lim. | 26.7 | t.lim. | — | 356.2 | — | 12.3 | — | 3.6 |
| 072_B_half | 71 | 217.78 | 43.7 | t.lim. | 27.8 | t.lim. | 18.4 | t.lim. | — | 567.9 | — | 6.4 | — | 2.9 |
| 072_B_one | 71 | 226.61 | 45.8 | t.lim. | 19.0 | t.lim. | 25.8 | t.lim. | — | 499.3 | — | 9.6 | — | 2.5 |
| 072_B_two | 71 | 244.25 | 49.8 | t.lim. | 105.5 | t.lim. | 22.7 | t.lim. | — | 8.5 | — | 6.9 | — | 5.1 |
| 076_B_p_two | 75 | 560.56 | 48.0 | t.lim. | 9.5 | t.lim. | 4.1 | t.lim. | 1.1 | t.lim. | 1.0 | t.lim. | — | 354.6 |
| 076_B_half | 75 | 666.86 | 68.1 | t.lim. | 2.1 | t.lim. | 4.5 | t.lim. | — | 10.4 | — | 48.0 | — | 14.0 |
| 076_B_one | 75 | 733.30 | 71.2 | t.lim. | 11.2 | t.lim. | 5.3 | t.lim. | — | 60.9 | — | 27.3 | — | 20.6 |
| 076_B_two | 75 | 866.22 | 50.3 | t.lim. | 31.3 | t.lim. | 5.8 | t.lim. | — | 4.5 | — | 10.5 | — | 2.7 |
| 101_B_p_two | 100 | 658.66 | 37.2 | m.lim. | 8.7 | t.lim. | 4.8 | t.lim. | 0.5 | t.lim. | 0.5 | t.lim. | — | 514.3 |
| 101_B_half_a | 100 | 780.00 | 59.8 | m.lim. | 0.6 | t.lim. | 4.5 | t.lim. | — | 2071.5 | — | 220.2 | — | 40.3 |
| 101_B_one_a | 100 | 847.91 | 48.8 | m.lim. | 8.5 | t.lim. | 5.7 | t.lim. | 0.5 | t.lim. | — | 115.8 | — | 63.5 |
| 101_B_two_a | 100 | 983.71 | 56.5 | m.lim. | 63.6 | t.lim. | 5.2 | t.lim. | 0.3 | t.lim. | — | 120.6 | — | 31.6 |
| 101_B_p_two_b | 100 | 536.38 | 52.1 | t.lim. | 23.2 | t.lim. | 26.2 | t.lim. | 1.0 | t.lim. | 1.0 | t.lim. | 0.2 | t.lim. |
| 101_B_half_b | 100 | 589.26 | 53.7 | t.lim. | 22.9 | t.lim. | 27.2 | t.lim. | 0.8 | t.lim. | 0.8 | t.lim. | — | 547.0 |
| 101_B_one_b | 100 | 628.74 | 57.3 | t.lim. | 15.9 | t.lim. | 25.2 | t.lim. | 0.6 | t.lim. | 0.9 | t.lim. | — | 739.9 |
| 101_B_two_b | 100 | 707.68 | 67.0 | t.lim. | 25.3 | t.lim. | 24.9 | t.lim. | 0.7 | t.lim. | 0.6 | t.lim. | — | 722.7 |

## 12.2. Detailed results for the evaluation of the intermediate dropoff relaxation

**Table 19 Detailed results for Table 3: GLS set,** $15 \leq n \leq 30$ **(**$n = |PD|$**).**

| | | TINE (for SVRPDSP-D) | | | | | | | | MEN (for SVRPDSP) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | $z_{opt}$ | gap | sec | $gap_r$ | $gap_c$ | nodes | feas | $gap_d$ | $z_{opt}$ | gap | sec | $gap_r$ | $gap_c$ | nodes | dupl | iter |
| 016_B_p_two | 15 | 221.00 | — | 0.0 | 1.0 | — | 0 | 1 | — | 221.00 | — | 0.0 | 1.0 | — | 0 | 0 | 1 |
| 016_B_half | 15 | 243.07 | — | 0.0 | 0.5 | — | 0 | 1 | — | 243.07 | — | 0.0 | 0.5 | — | 0 | 0 | 1 |
| 016_B_one | 15 | 262.20 | — | 0.0 | 2.2 | — | 0 | 1 | — | 262.20 | — | 0.0 | 2.2 | — | 0 | 0 | 1 |
| 016_B_two | 15 | 289.73 | — | 0.0 | 2.0 | — | 0 | 1 | — | 289.73 | — | 0.0 | 2.0 | — | 0 | 0 | 1 |
| 021_B_p_two | 20 | 265.04 | — | 0.1 | 2.0 | 0.1 | 8 | 1 | — | 265.04 | — | 0.0 | 2.0 | 0.1 | 8 | 0 | 1 |
| 021_B_half | 20 | 292.18 | — | 0.0 | — | — | 32 | 1 | — | 292.18 | — | 0.0 | — | — | 31 | 0 | 1 |
| 021_B_one | 20 | 316.89 | — | 0.0 | 2.6 | — | 0 | 1 | — | 316.89 | — | 0.0 | 2.6 | — | 0 | 0 | 1 |
| 021_B_two | 20 | 348.13 | — | 0.0 | 2.4 | — | 0 | 1 | — | 348.13 | — | 0.0 | 2.4 | — | 0 | 0 | 1 |
| 022_B_p_two | 21 | 287.91 | — | 0.4 | 4.7 | 3.7 | 646 | 1 | — | 287.91 | — | 0.2 | 4.7 | 3.7 | 482 | 0 | 1 |
| 022_B_half | 21 | 319.76 | — | 0.2 | 2.6 | 0.0 | 147 | 1 | — | 319.76 | — | 0.1 | 2.6 | <0.1 | 158 | 0 | 1 |
| 022_B_one | 21 | 343.74 | — | 0.3 | 4.1 | 3.0 | 496 | 1 | — | 343.74 | — | 0.2 | 4.1 | 3.0 | 430 | 0 | 1 |
| 022_B_two | 21 | 380.16 | — | 0.2 | 3.7 | 2.7 | 169 | 1 | — | 380.16 | — | 0.2 | 3.7 | 2.7 | 223 | 0 | 1 |
| 023_B_p_two | 22 | 527.71 | — | 0.5 | 3.9 | 1.5 | 571 | 1 | — | 527.71 | — | 0.5 | 3.9 | 1.5 | 863 | 0 | 1 |
| 023_B_half | 22 | 565.80 | — | 0.1 | 4.5 | — | 0 | 1 | — | 565.80 | — | 0.1 | 4.5 | — | 0 | 0 | 1 |
| 023_B_one | 22 | 595.20 | — | 0.1 | 4.3 | — | 7 | 1 | — | 595.20 | — | 0.1 | 4.3 | — | 6 | 0 | 1 |
| 023_B_two | 22 | 653.99 | — | 0.1 | 3.9 | — | 7 | 1 | — | 653.99 | — | 0.1 | 3.9 | — | 6 | 0 | 1 |
| 026_B_p_two | 25 | 319.54 | — | 0.2 | 5.6 | 0.5 | 74 | 1 | — | 319.54 | — | 0.1 | 5.6 | 0.5 | 35 | 0 | 1 |
| 026_B_half | 25 | 343.58 | — | 0.3 | 3.7 | 0.0 | 311 | 1 | — | 343.58 | — | 0.2 | 3.7 | <0.1 | 244 | 0 | 1 |
| 026_B_one | 25 | 374.79 | — | 0.2 | 5.3 | 0.9 | 67 | 0 | — | 374.79 | — | 0.3 | 5.3 | 0.9 | 262 | 1 | 2 |
| 026_B_two | 25 | 409.67 | — | 0.1 | 4.8 | 0.0 | 7 | 0 | — | 409.67 | — | 0.3 | 4.8 | <0.1 | 56 | 1 | 2 |
| 030_B_p_two | 29 | 402.70 | — | 0.7 | 16.7 | 0.1 | 318 | 1 | — | 402.70 | — | 0.3 | 16.7 | 0.1 | 340 | 0 | 1 |
| 030_B_half | 29 | 422.47 | — | 0.4 | 16.7 | 0.3 | 153 | 1 | — | 422.47 | — | 0.3 | 16.7 | 0.3 | 384 | 0 | 1 |
| 030_B_one | 29 | 450.10 | — | 0.3 | 15.7 | 0.2 | 61 | 1 | — | 450.10 | — | 0.3 | 15.7 | 0.2 | 302 | 0 | 1 |
| 030_B_two | 29 | 505.34 | — | 0.3 | 14.0 | 0.2 | 91 | 1 | — | 505.34 | — | 0.2 | 14.0 | 0.2 | 127 | 0 | 1 |
| 031_B_p_two | 30 | 326.60 | — | 0.5 | 2.4 | 1.6 | 331 | 1 | — | 326.60 | — | 0.5 | 2.4 | 1.6 | 140 | 1 | 2 |
| 031_B_half | 30 | 361.23 | — | 0.6 | 0.8 | 0.2 | 593 | 1 | — | 361.23 | — | 0.4 | 0.8 | 0.2 | 471 | 0 | 1 |
| 031_B_one | 30 | 393.78 | — | 1.0 | 1.3 | 0.8 | 635 | 1 | — | 393.78 | — | 0.8 | 1.3 | 0.8 | 690 | 0 | 1 |
| 031_B_two | 30 | 454.10 | — | 1.8 | 1.2 | 0.7 | 1881 | 1 | — | 454.10 | — | 0.5 | 1.2 | 0.7 | 506 | 0 | 1 |

**Table 20 Detailed results for Table 3: GLS set, $32 \leq n \leq 50$ ($n = |PD|$).**

| | | TINE (for SVRPDSP-D) | | | | | | | | MEN (for SVRPDSP) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | $z_{opt}$ | gap | sec | $\text{gap}_r$ | $\text{gap}_c$ | nodes | feas | $\text{gap}_d$ | $z_{opt}$ | gap | sec | $\text{gap}_r$ | $\text{gap}_c$ | nodes | dupl | iter |
| 033_B_p_two | 32 | 459.03 | — | 1.8 | 10.9 | 1.2 | 2626 | 1 | — | 459.03 | — | 1.6 | 10.9 | 1.2 | 2767 | 0 | 1 |
| 033_B_half | 32 | 490.40 | — | 1.2 | 10.3 | — | 940 | 1 | — | 490.40 | — | 1.6 | 10.3 | — | 2141 | 0 | 1 |
| 033_B_one | 32 | 516.49 | — | 1.0 | 10.9 | 0.1 | 893 | 1 | — | 516.49 | — | 0.3 | 10.9 | 2.4 | 165 | 0 | 1 |
| 033_B_two | 32 | 556.56 | — | 0.9 | 10.1 | 0.1 | 662 | 1 | — | 556.56 | — | 0.6 | 10.1 | 0.1 | 525 | 0 | 1 |
| 036_B_p_two | 35 | 360.39 | — | 2.4 | 3.1 | 2.3 | 1766 | 1 | — | 360.39 | — | 1.8 | 3.1 | 2.3 | 1209 | 1 | 2 |
| 036_B_half | 35 | 397.93 | — | 2.6 | 0.7 | 0.1 | 559 | 1 | — | 397.93 | — | 0.8 | 0.7 | 0.1 | 692 | 0 | 1 |
| 036_B_one | 35 | 428.00 | — | 2.1 | 0.7 | 0.1 | 221 | 1 | — | 428.00 | — | 0.4 | 0.7 | — | 183 | 0 | 1 |
| 036_B_two | 35 | 488.13 | — | 0.7 | 0.6 | — | 527 | 1 | — | 488.13 | — | 0.6 | 0.6 | — | 433 | 0 | 1 |
| 041_B_p_two | 40 | 374.70 | — | 3.9 | 2.8 | 2.5 | 2045 | 1 | — | 374.70 | — | 8.4 | 2.8 | 2.5 | 11034 | 1 | 2 |
| 041_B_half | 40 | 426.53 | — | 1.3 | 0.6 | 0.1 | 163 | 1 | — | 426.53 | — | 0.5 | 0.6 | 0.1 | 153 | 0 | 1 |
| 041_B_one | 40 | 463.90 | — | 5.9 | 2.0 | 1.5 | 5844 | 1 | — | 463.90 | — | 4.0 | 2.0 | 1.5 | 4836 | 0 | 1 |
| 041_B_two | 40 | 525.12 | — | 3.9 | 1.7 | 1.3 | 2888 | 1 | — | 525.12 | — | 2.8 | 1.7 | 1.3 | 2789 | 0 | 1 |
| 045_B_p_two | 44 | 671.79 | — | 0.8 | 17.6 | 0.3 | 74 | 1 | — | 671.79 | — | 0.4 | 17.6 | 0.3 | 74 | 0 | 1 |
| 045_B_half | 44 | 693.18 | — | 0.1 | 16.8 | — | 0 | 1 | — | 693.18 | — | 0.1 | 16.8 | — | 0 | 0 | 1 |
| 045_B_one | 44 | 720.09 | — | 0.2 | 16.2 | — | 3 | 1 | — | 720.09 | — | 0.2 | 16.2 | — | 3 | 0 | 1 |
| 045_B_two | 44 | 773.92 | — | 0.1 | 15.1 | — | 1 | 1 | — | 773.92 | — | 0.1 | 15.1 | — | 1 | 0 | 1 |
| 048_B_p_two | 47 | 36703.71 | — | 5.0 | 6.2 | 1.9 | 1319 | 1 | — | 36703.71 | — | 3.2 | 6.2 | 1.9 | 676 | 2 | 3 |
| 048_B_half | 47 | 39983.45 | — | 6.9 | 5.7 | 1.9 | 2251 | 1 | — | 39983.45 | — | 2.7 | 5.7 | 1.9 | 1035 | 1 | 2 |
| 048_B_one | 47 | 47319.14 | — | 2.6 | 3.7 | 0.2 | 423 | 1 | — | 47319.14 | — | 1.3 | 3.7 | 0.2 | 515 | 0 | 1 |
| 048_B_two | 47 | 60457.70 | — | 5.1 | 2.9 | 0.1 | 1036 | 1 | — | 60457.70 | — | 0.8 | 2.9 | 0.1 | 99 | 0 | 1 |
| 051_B_p_two | 50 | 447.85 | — | 31.2 | 3.0 | 1.8 | 23890 | 1 | — | 447.85 | — | 36.7 | 3.0 | 1.8 | 29186 | 1 | 2 |
| 051_B_half | 50 | 498.17 | — | 62.1 | 3.2 | 1.9 | 46946 | 1 | — | 498.17 | — | 32.5 | 3.2 | 2.1 | 22605 | 1 | 2 |
| 051_B_one | 50 | 530.96 | — | 40.6 | 3.1 | 1.9 | 25665 | 1 | — | 530.96 | — | 26.7 | 3.1 | 1.9 | 22717 | 1 | 2 |
| 051_B_two | 50 | 595.80 | — | 68.6 | 2.7 | 1.7 | 42107 | 1 | — | 595.80 | — | 28.4 | 2.7 | 1.7 | 21177 | 1 | 2 |

**Table 21 Detailed results for Table 3: GLS set, $71 \leq n \leq 100$ ($n = |PD|$). The value $z_{opt}$=536.38 for instance 101_B_p_two_b was obtained by running MEN for 17172 seconds and 2410673 nodes.**

| | | TINE (for SVRPDSP-D) | | | | | | | | MEN (for SVRPDSP) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | $z_{opt}$ | gap | sec | $\text{gap}_r$ | $\text{gap}_c$ | nodes | feas | $\text{gap}_d$ | $z_{opt}$ | gap | sec | $\text{gap}_r$ | $\text{gap}_c$ | nodes | dupl | iter |
| 072_B_p_two | 71 | 212.49 | — | 10.5 | 17.0 | 0.6 | 486 | 1 | — | 212.49 | — | 3.6 | 17.0 | 0.6 | 416 | 0 | 1 |
| 072_B_half | 71 | 217.78 | — | 28.5 | 16.6 | 0.6 | 594 | 1 | — | 217.78 | — | 2.9 | 16.6 | 0.6 | 214 | 0 | 1 |
| 072_B_one | 71 | 226.61 | — | 6.6 | 15.9 | 0.5 | 68 | 1 | — | 226.61 | — | 2.5 | 15.9 | 0.5 | 174 | 0 | 1 |
| 072_B_two | 71 | 244.25 | — | 9.2 | 14.8 | 0.5 | 278 | 1 | — | 244.25 | — | 5.1 | 14.8 | 0.5 | 348 | 0 | 1 |
| 076_B_p_two | 75 | 559.16 | 0.25 | 355.4 | 1.6 | 1.4 | 96245 | 0 | 0.2 | 560.56 | — | 354.6 | 1.9 | 1.6 | 64000 | 3 | 4 |
| 076_B_half | 75 | 666.86 | — | 7.4 | 0.5 | 0.1 | 1099 | 1 | — | 666.86 | — | 14.0 | 0.5 | 0.1 | 2868 | 0 | 1 |
| 076_B_one | 75 | 733.30 | — | 39.1 | 0.5 | 0.1 | 9001 | 1 | — | 733.30 | — | 20.6 | 0.5 | 0.1 | 5068 | 0 | 1 |
| 076_B_two | 75 | 866.22 | — | 3.2 | 0.4 | 0.1 | 270 | 1 | — | 866.22 | — | 2.7 | 0.4 | 0.1 | 310 | 0 | 1 |
| 101_B_p_two | 100 | 657.52 | 0.17 | 122.1 | 2.3 | 1.3 | 8796 | 0 | 0.2 | 658.66 | — | 514.3 | 2.5 | 1.5 | 54226 | 4 | 5 |
| 101_B_half_a | 100 | 780.00 | — | 82.7 | 1.6 | 0.9 | 14539 | 1 | — | 780.00 | — | 40.3 | 1.6 | 0.9 | 7997 | 0 | 1 |
| 101_B_one_a | 100 | 847.91 | — | 160.7 | 1.5 | 0.8 | 13845 | 1 | — | 847.91 | — | 63.5 | 1.5 | 0.8 | 12495 | 0 | 1 |
| 101_B_two_a | 100 | 983.71 | — | 145.6 | 1.3 | 0.7 | 9615 | 1 | — | 983.71 | — | 31.6 | 1.3 | 0.7 | 4718 | 0 | 1 |
| 101_B_p_two_b | 100 | 534.72 | 0.34 | t.lim. | 27.4 | 1.7 | 335348 | 0 | 0.3 | 536.38 | 0.2 | t.lim. | 27.6 | 2.0 | 184356 | 3 | 4 |
| 101_B_half_b | 100 | 589.23 | 0.01 | 318.3 | 25.1 | 1.5 | 49924 | 0 | 0.0 | 589.26 | — | 547.0 | 25.1 | 1.5 | 48811 | 3 | 4 |
| 101_B_one_b | 100 | 628.71 | 0.00 | 302.1 | 23.3 | 1.4 | 42555 | 0 | 0.0 | 628.74 | — | 739.9 | 23.3 | 1.4 | 59168 | 3 | 4 |
| 101_B_two_b | 100 | 707.65 | 0.00 | 450.8 | 20.7 | 1.2 | 58697 | 0 | 0.0 | 707.68 | — | 722.7 | 20.7 | 1.2 | 69607 | 3 | 4 |

## 12.3. Detailed results for the case of mandatory pickups

**Table 22 Detailed results for Table 4: GHLV set, $15 \leq n \leq 30$ ($n = |PD|$).**

| | | | GHLV | | TA-MP | | 2S-MP | | MEN-MP | |
|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | $z_{opt}$ | sec | gap | sec | gap | sec | gap | sec | gap |
| 016_B02_CA | 15 | 220.99 | t.lim. | 2.7 | 0.0 | — | 0.0 | — | 0.1 | — |
| 016_B02_CB | 15 | 220.74 | 1068.5 | — | 0.1 | — | 0.1 | — | 0.0 | — |
| 021_B02_CA | 20 | 267.23 | t.lim. | 4.2 | 0.7 | — | 0.6 | — | 0.6 | — |
| 021_B02_CB | 20 | 261.81 | t.lim. | 5.2 | 0.5 | — | 0.7 | — | 0.4 | — |
| 022_B02_CA | 21 | 278.43 | t.lim. | 13.1 | 0.3 | — | 0.1 | — | 0.1 | — |
| 022_B02_CB | 21 | 278.43 | t.lim. | 14.5 | 0.1 | — | 0.5 | — | 0.5 | — |
| 023_B02_CA | 22 | 482.78 | t.lim. | 9.9 | 0.2 | — | 0.5 | — | 0.1 | — |
| 023_B02_CB | 22 | 482.71 | t.lim. | 16.9 | 0.1 | — | 0.4 | — | 0.1 | — |
| 026_B02_CA | 25 | 306.93 | t.lim. | 18.0 | 0.2 | — | 0.1 | — | 0.2 | — |
| 026_B02_CB | 25 | 314.16 | t.lim. | 15.2 | 0.5 | — | 0.1 | — | 0.6 | — |
| 030_B02_CA | 29 | 388.43 | t.lim. | 25.2 | 0.5 | — | 0.5 | — | 0.7 | — |
| 030_B02_CB | 29 | 386.9 | t.lim. | 18.0 | 0.5 | — | 0.4 | — | 0.4 | — |
| 031_B02_CA | 30 | 316.67 | t.lim. | 3.1 | 0.4 | — | 0.4 | — | 0.2 | — |
| 031_B02_CB | 30 | 322.32 | t.lim. | 5.4 | 1.2 | — | 2.7 | — | 1.6 | — |

**Table 23 Detailed results for Table 4: GHLV set, $32 \leq n \leq 50$ ($n = |PD|$).**

| | | | GHLV | | TA-MP | | 2S-MP | | MEN-MP | |
|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | $z_{opt}$ | sec | gap | sec | gap | sec | gap | sec | gap |
| 033_B02_CA | 32 | 447.05 | t.lim. | 20.3 | 15.9 | — | 0.5 | — | 0.8 | — |
| 033_B02_CB | 32 | 463.68 | t.lim. | 17.5 | t.lim. | 1.1 | t.lim. | 1.2 | 1.4 | — |
| 036_B02_CA | 35 | 353.26 | t.lim. | 6.8 | 1.6 | — | 0.9 | — | 1.4 | — |
| 036_B02_CB | 35 | 359.54 | t.lim. | 7.6 | 10.4 | — | 1.1 | — | 1.4 | — |
| 041_B02_CA | 40 | 365.6 | t.lim. | 5.0 | 1.1 | — | 0.6 | — | 0.4 | — |
| 041_B02_CB | 40 | 367.97 | t.lim. | 5.9 | 5.3 | — | 2.4 | — | 2.2 | — |
| 045_B02_CA | 44 | 619.17 | t.lim. | 34.3 | 0.5 | — | 0.4 | — | 0.3 | — |
| 045_B02_CB | 44 | 619.18 | t.lim. | 34.7 | 0.4 | — | 0.4 | — | 0.4 | — |
| 048_B02_CA | 47 | 33523.67 | t.lim. | 20.0 | 0.7 | — | 0.5 | — | 0.8 | — |
| 048_B02_CB | 47 | 34056.27 | t.lim. | 19.1 | 1.7 | — | 9.6 | — | 1.3 | — |
| 051_B02_CA | 50 | 428.86 | t.lim. | 10.9 | 12.0 | — | 3.2 | — | 2.9 | — |
| 051_B02_CB | 50 | 435.51 | t.lim. | 12.6 | 3067.0 | — | 2036.7 | — | 29.7 | — |

**Table 24 Detailed results for Table 4: GHLV set, $71 \leq n \leq 100$ ($n = |PD|$).**

| | | | GHLV | | TA-MP | | 2S-MP | | MEN-MP | |
|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | $z_{opt}$ | sec | gap | sec | gap | sec | gap | sec | gap |
| 072_B02_CA | 71 | 198.92 | t.lim. | 33.6 | 81.8 | — | 6.8 | — | 6.7 | — |
| 072_B02_CB | 71 | 194.68 | t.lim. | 32.1 | 3.2 | — | 2.3 | — | 2.3 | — |
| 076_B02_CA | 75 | 545.59 | t.lim. | 10.0 | 5.8 | — | 4.6 | — | 4.4 | — |
| 076_B02_CB | 75 | 548.27 | t.lim. | 10.6 | 859.5 | — | 37.9 | — | 36.2 | — |
| 101_B02_CA_a | 100 | 640.13 | t.lim. | 9.2 | 7.7 | — | 8.3 | — | 8.1 | — |
| 101_B02_CB_a | 100 | 646.39 | t.lim. | 10.1 | t.lim. | 0.5 | 1441.0 | — | 560.9 | — |
| 101_B02_CA_b | 100 | 503.96 | t.lim. | 35.3 | 60.3 | — | 61.0 | — | 58.0 | — |
| 101_B02_CB_b | 100 | 503.96 | t.lim. | 34.9 | 1344.7 | — | 29.6 | — | 29.0 | — |

## 12.4. Detailed results for the new large size instances

**Table 25 Detailed results for Table 7: selective pickups, BI set ($n = |PD|$).**

| | | | MEN | | | |
|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | $UB_{best}$ | UB | sec | gap | nodes |
| 121_B_p_two | 120 | 606.07 | 607.60 | t.lim. | 3.3 | 170600 |
| 121_B_half | 120 | 739.39 | 739.39 | t.lim. | 0.3 | 271802 |
| 121_B_one | 120 | 851.97 | 866.29 | t.lim. | 0.9 | 113068 |
| 121_B_two | 120 | 1060.42 | 1060.42 | 2034.4 | — | 278810 |
| 135_B_p_two | 134 | 755.24 | 785.75 | t.lim. | 1.5 | 119824 |
| 135_B_half | 134 | 765.09 | 765.09 | t.lim. | 1.0 | 61780 |
| 135_B_one | 134 | 795.69 | 795.69 | t.lim. | 1.3 | 135104 |
| 135_B_two | 134 | 845.32 | 845.32 | t.lim. | 1.2 | 50265 |
| 151_B_p_two | 150 | 755.88 | 755.88 | 1335.9 | — | 81717 |
| 151_B_half | 150 | 863.09 | 863.09 | 747.7 | — | 33432 |
| 151_B_one | 150 | 946.75 | 946.75 | 214.0 | — | 9810 |
| 151_B_two | 150 | 1114.10 | 1114.10 | 380.7 | — | 18055 |
| 200_B_p_two | 199 | 860.83 | 860.83 | t.lim. | 2.8 | 70647 |
| 200_B_half | 199 | 1021.23 | 1021.23 | 3076.3 | — | 19019 |
| 200_B_one | 199 | 1152.87 | 1152.87 | 339.6 | — | 11771 |
| 200_B_two | 199 | 1416.95 | 1416.95 | t.lim. | 0.1 | 98431 |

**Table 26 Detailed results for Table 7: mandatory pickups, BI-MP set ($n = |PD|$).**

| | | | MEN-MP | | | |
|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | $UB_{best}$ | UB | sec | gap | nodes |
| 121_B02_CA | 120 | 544.16 | 544.16 | 103.6 | — | 2952 |
| 121_B02_CB | 120 | 549.16 | 549.16 | 1817.0 | — | 82068 |
| 135_B02_CA | 134 | 720.56 | 720.56 | 3026.7 | — | 14968 |
| 135_B02_CB | 134 | 721.85 | 721.85 | 1488.5 | — | 10992 |
| 151_B02_CA | 150 | 710.16 | 710.16 | 366.9 | — | 3502 |
| 151_B02_CB | 150 | 714.66 | 714.66 | 316.2 | — | 343 |
| 200_B02_CA | 199 | 790.5 | 790.5 | t.lim. | 1.6 | 36200 |
| 200_B02_CB | 199 | 782.43 | 782.43 | t.lim. | 0.5 | 56000 |

## 12.5. Detailed results for the case of multiple vehicles

**Table 27 Detailed results for Table 8: GLS-Multi set, $\alpha = 0.6$, $15 \leq n \leq 30$ ($n = |PD|$)).**

| | | $k=2$ | | | | $k=3$ | | | | $k=4$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | UB | sec | gap | iter | UB | sec | gap | iter | UB | sec | gap | iter |
| 016_B_p_two | 15 | 238.66 | 0.0 | — | 1 | 251.64 | 0.0 | — | 1 | 269.22 | 0.1 | — | 1 |
| 016_B_half | 15 | 238.66 | 0.0 | — | 1 | 251.64 | 0.0 | — | 1 | 269.22 | 0.1 | — | 1 |
| 016_B_one | 15 | 238.66 | 0.0 | — | 1 | 251.64 | 0.0 | — | 1 | 269.22 | 0.1 | — | 1 |
| 016_B_two | 15 | 238.66 | 0.0 | — | 1 | 251.64 | 0.0 | — | 1 | 269.22 | 0.1 | — | 1 |
| 021_B_p_two | 20 | 278.40 | 0.4 | — | 1 | 287.04 | 0.0 | — | 1 | 302.51 | 0.0 | — | 1 |
| 021_B_half | 20 | 289.29 | 7.7 | — | 3 | 287.04 | 0.0 | — | 1 | 302.51 | 0.0 | — | 1 |
| 021_B_one | 20 | 289.29 | 3.8 | — | 3 | 287.04 | 0.0 | — | 1 | 302.51 | 0.0 | — | 1 |
| 021_B_two | 20 | 289.29 | 3.4 | — | 3 | 287.04 | 0.0 | — | 1 | 302.51 | 0.0 | — | 1 |
| 022_B_p_two | 21 | 290.43 | 0.5 | — | 1 | 301.72 | 0.1 | — | 1 | 320.79 | 0.1 | — | 1 |
| 022_B_half | 21 | 297.82 | 0.2 | — | 1 | 308.21 | 0.2 | — | 1 | 320.79 | 0.1 | — | 1 |
| 022_B_one | 21 | 297.82 | 0.3 | — | 1 | 308.21 | 0.2 | — | 1 | 320.79 | 0.1 | — | 1 |
| 022_B_two | 21 | 297.82 | 0.3 | — | 1 | 308.21 | 0.2 | — | 1 | 320.79 | 0.1 | — | 1 |
| 023_B_p_two | 22 | 547.05 | 18.6 | — | 4 | 550.50 | 5.7 | — | 3 | 583.16 | 0.8 | — | 3 |
| 023_B_half | 22 | 567.59 | 19.6 | — | 4 | 568.14 | 9.7 | — | 3 | 583.16 | 0.8 | — | 3 |
| 023_B_one | 22 | 596.99 | 16.2 | — | 4 | 569.20 | 4.7 | — | 3 | 583.16 | 0.8 | — | 3 |
| 023_B_two | 22 | 655.79 | 14.2 | — | 4 | 569.20 | 3.1 | — | 3 | 583.16 | 0.8 | — | 3 |
| 026_B_p_two | 25 | 334.00 | 0.8 | — | 1 | 342.75 | 0.2 | — | 1 | 361.86 | 0.3 | — | 1 |
| 026_B_half | 25 | 337.22 | 1.9 | — | 2 | 342.75 | 0.2 | — | 1 | 361.86 | 0.3 | — | 1 |
| 026_B_one | 25 | 337.22 | 1.9 | — | 2 | 342.75 | 0.2 | — | 1 | 361.86 | 0.3 | — | 1 |
| 026_B_two | 25 | 337.22 | 2.2 | — | 2 | 342.75 | 0.4 | — | 1 | 361.86 | 0.3 | — | 1 |
| 030_B_p_two | 29 | 390.12 | 0.2 | — | 1 | 408.55 | 0.5 | — | 1 | 430.11 | 0.3 | — | 1 |
| 030_B_half | 29 | 390.12 | 0.2 | — | 1 | 408.55 | 0.3 | — | 1 | 430.11 | 0.3 | — | 1 |
| 030_B_one | 29 | 390.12 | 0.2 | — | 1 | 408.55 | 0.5 | — | 1 | 430.11 | 0.3 | — | 1 |
| 030_B_two | 29 | 390.12 | 0.2 | — | 1 | 408.55 | 0.7 | — | 1 | 430.11 | 0.3 | — | 1 |
| 031_B_p_two | 30 | 331.88 | 0.9 | — | 1 | 342.25 | 0.8 | — | 1 | 353.98 | 0.5 | — | 1 |
| 031_B_half | 30 | 331.88 | 0.9 | — | 1 | 342.25 | 0.8 | — | 1 | 353.98 | 0.5 | — | 1 |
| 031_B_one | 30 | 331.88 | 0.9 | — | 1 | 342.25 | 0.8 | — | 1 | 353.98 | 0.5 | — | 1 |
| 031_B_two | 30 | 331.88 | 0.9 | — | 1 | 342.25 | 0.8 | — | 1 | 353.98 | 0.5 | — | 1 |

**Table 28 Detailed results for Table 8: GLS-Multi set, $\alpha = 0.6$, $32 \leq n \leq 50$ ($n = |PD|$)).**

| | | $k=2$ | | | | $k=3$ | | | | $k=4$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | UB | sec | gap | iter | UB | sec | gap | iter | UB | sec | gap | iter |
| 033_B_p_two | 32 | 558.94 | 13.9 | — | 3 | 620.50 | 1.1 | — | 1 | 688.65 | 0.5 | — | 1 |
| 033_B_half | 32 | 558.94 | 9.1 | — | 3 | 620.50 | 1.5 | — | 1 | 688.65 | 0.5 | — | 1 |
| 033_B_one | 32 | 558.94 | 12.1 | — | 3 | 620.50 | 1.8 | — | 1 | 688.65 | 0.5 | — | 1 |
| 033_B_two | 32 | 558.94 | 12.6 | — | 3 | 620.50 | 1.1 | — | 1 | 688.65 | 0.5 | — | 1 |
| 036_B_p_two | 35 | 360.54 | 0.3 | — | 1 | 370.54 | 0.9 | — | 1 | 380.91 | 0.4 | — | 1 |
| 036_B_half | 35 | 360.54 | 0.3 | — | 1 | 370.54 | 0.9 | — | 1 | 380.91 | 0.5 | — | 1 |
| 036_B_one | 35 | 360.54 | 0.4 | — | 1 | 370.54 | 0.9 | — | 1 | 380.91 | 0.4 | — | 1 |
| 036_B_two | 35 | 360.54 | 0.4 | — | 1 | 370.54 | 0.9 | — | 1 | 380.91 | 0.5 | — | 1 |
| 041_B_p_two | 40 | 376.64 | 4.9 | — | 1 | 386.86 | 5.3 | — | 1 | 396.48 | 4.4 | — | 1 |
| 041_B_half | 40 | 377.71 | 2.7 | — | 1 | 386.86 | 6.4 | — | 1 | 397.48 | 4.0 | — | 1 |
| 041_B_one | 40 | 377.71 | 2.9 | — | 1 | 386.86 | 6.6 | — | 1 | 397.48 | 3.0 | — | 1 |
| 041_B_two | 40 | 377.71 | 2.9 | — | 1 | 386.86 | 3.1 | — | 1 | 397.48 | 3.1 | — | 1 |
| 045_B_p_two | 44 | 640.47 | 692.0 | — | 5 | 636.30 | 0.6 | — | 1 | 641.80 | 2.1 | — | 1 |
| 045_B_half | 44 | 640.47 | 587.4 | — | 5 | 636.30 | 0.6 | — | 1 | 641.80 | 2.1 | — | 1 |
| 045_B_one | 44 | 640.47 | 617.3 | — | 5 | 636.30 | 0.6 | — | 1 | 641.80 | 2.1 | — | 1 |
| 045_B_two | 44 | 640.47 | 700.6 | — | 6 | 636.30 | 0.6 | — | 1 | 641.80 | 2.1 | — | 1 |
| 048_B_p_two | 47 | 34693.66 | 3.1 | — | 1 | 35410.25 | 4.4 | — | 1 | 36254.40 | 1.5 | — | 1 |
| 048_B_half | 47 | 34693.66 | 2.2 | — | 1 | 35410.25 | 4.5 | — | 1 | 36254.40 | 1.5 | — | 1 |
| 048_B_one | 47 | 34693.66 | 2.2 | — | 1 | 35410.25 | 5.9 | — | 1 | 36254.40 | 1.5 | — | 1 |
| 048_B_two | 47 | 34693.66 | 2.2 | — | 1 | 35410.25 | 6.5 | — | 1 | 36254.40 | 1.5 | — | 1 |
| 051_B_p_two | 50 | 443.99 | 11.9 | — | 1 | 447.19 | 19.0 | — | 1 | 457.89 | 12.5 | — | 1 |
| 051_B_half | 50 | 444.31 | 9.3 | — | 1 | 448.42 | 15.4 | — | 1 | 458.02 | 8.3 | — | 1 |
| 051_B_one | 50 | 444.31 | 19.6 | — | 1 | 448.42 | 24.0 | — | 1 | 458.02 | 8.8 | — | 1 |
| 051_B_two | 50 | 444.31 | 10.8 | — | 1 | 448.42 | 20.0 | — | 1 | 458.02 | 8.9 | — | 1 |

**Table 29 Detailed results for Table 8: GLS-Multi set, $\alpha = 0.6$, $71 \leq n \leq 100$ ($n = |PD|$)).**

| | | $k=2$ | | | | $k=3$ | | | | $k=4$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | UB | sec | gap | iter | UB | sec | gap | iter | UB | sec | gap | iter |
| 072_B_p_two | 71 | 204.06 | 4.4 | — | 1 | 215.44 | 2.1 | — | 1 | 227.39 | 5.0 | — | 1 |
| 072_B_half | 71 | 204.06 | 4.5 | — | 1 | 215.44 | 2.2 | — | 1 | 227.39 | 5.0 | — | 1 |
| 072_B_one | 71 | 204.06 | 4.4 | — | 1 | 215.44 | 2.1 | — | 1 | 227.39 | 5.0 | — | 1 |
| 072_B_two | 71 | 204.06 | 4.3 | — | 1 | 215.44 | 2.1 | — | 1 | 227.39 | 5.0 | — | 1 |
| 076_B_p_two | 75 | 556.59 | 44.2 | — | 1 | 562.40 | 150.9 | — | 1 | 569.15 | 71.6 | — | 1 |
| 076_B_half | 75 | 556.59 | 193.4 | — | 1 | 562.40 | 108.9 | — | 1 | 570.05 | 21.3 | — | 1 |
| 076_B_one | 75 | 556.59 | 42.8 | — | 1 | 562.40 | 114.8 | — | 1 | 570.05 | 42.3 | — | 1 |
| 076_B_two | 75 | 556.59 | 134.6 | — | 1 | 562.40 | 126.1 | — | 1 | 570.05 | 56.9 | — | 1 |
| 101_B_p_two_a | 100 | 651.35 | 193.7 | — | 1 | 656.10 | 37.0 | — | 1 | 665.44 | 75.8 | — | 1 |
| 101_B_half_a | 100 | 654.53 | 756.9 | — | 1 | 660.66 | 515.8 | — | 1 | 668.23 | 98.0 | — | 1 |
| 101_B_one_a | 100 | 654.53 | 848.0 | — | 1 | 660.66 | 776.7 | — | 1 | 668.23 | 156.6 | — | 1 |
| 101_B_two_a | 100 | 654.53 | 860.6 | — | 1 | 660.66 | 454.9 | — | 1 | 668.23 | 292.0 | — | 1 |
| 101_B_p_two_b | 100 | 526.01 | 385.5 | — | 1 | 544.21 | 298.7 | — | 1 | 564.58 | 353.6 | — | 1 |
| 101_B_half_b | 100 | 526.01 | 344.3 | — | 1 | 544.21 | 201.0 | — | 1 | 564.58 | 236.6 | — | 1 |
| 101_B_one_b | 100 | 526.01 | 313.6 | — | 1 | 544.21 | 532.3 | — | 1 | 564.58 | 229.5 | — | 1 |
| 101_B_two_b | 100 | 526.01 | 610.0 | — | 1 | 544.21 | 335.6 | — | 1 | 564.58 | 326.5 | — | 1 |

**Table 30 Detailed results for Table 8: GLS-Multi set, $\alpha = 0.5$, $15 \leq n \leq 30$ ($n = |PD|$)).**

| | | $k=3$ | | | | $k=4$ | | | | $k=5$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | UB | sec | gap | iter | UB | sec | gap | iter | UB | sec | gap | iter |
| 016_B_p_two | 15 | 253.76 | 0.7 | — | 1 | 269.54 | 0.1 | — | 1 | 290.87 | 0.0 | — | 1 |
| 016_B_half | 15 | 253.76 | 0.1 | — | 1 | 269.54 | 0.1 | — | 1 | 290.87 | 0.0 | — | 1 |
| 016_B_one | 15 | 253.76 | 0.1 | — | 1 | 269.54 | 0.1 | — | 1 | 290.87 | 0.0 | — | 1 |
| 016_B_two | 15 | 253.76 | 0.1 | — | 1 | 269.54 | 0.1 | — | 1 | 290.87 | 0.0 | — | 1 |
| 021_B_p_two | 20 | 297.13 | 4.0 | — | 2 | 304.21 | 0.0 | — | 1 | 326.57 | 0.0 | — | 1 |
| 021_B_half | 20 | 303.18 | 1.9 | — | 1 | 304.21 | 0.0 | — | 1 | 326.57 | 0.0 | — | 1 |
| 021_B_one | 20 | 303.18 | 1.5 | — | 1 | 304.21 | 0.0 | — | 1 | 326.57 | 0.0 | — | 1 |
| 021_B_two | 20 | 303.18 | 1.2 | — | 1 | 304.21 | 0.0 | — | 1 | 326.57 | 0.0 | — | 1 |
| 022_B_p_two | 21 | 312.00 | 0.8 | — | 1 | 324.76 | 0.8 | — | 1 | 341.15 | 0.2 | — | 1 |
| 022_B_half | 21 | 318.18 | 1.3 | — | 1 | 329.62 | 1.0 | — | 1 | 341.15 | 0.1 | — | 1 |
| 022_B_one | 21 | 318.18 | 0.8 | — | 1 | 329.62 | 0.7 | — | 1 | 341.15 | 0.1 | — | 1 |
| 022_B_two | 21 | 318.18 | 0.9 | — | 1 | 329.62 | 0.8 | — | 1 | 341.15 | 0.1 | — | 1 |
| 023_B_p_two | 22 | 550.50 | 5.6 | — | 3 | 583.16 | 0.8 | — | 3 | 624.98 | 0.2 | — | 1 |
| 023_B_half | 22 | 568.14 | 9.6 | — | 3 | 583.16 | 0.8 | — | 3 | 624.98 | 0.2 | — | 1 |
| 023_B_one | 22 | 569.20 | 4.6 | — | 3 | 583.16 | 0.8 | — | 3 | 624.98 | 0.2 | — | 1 |
| 023_B_two | 22 | 569.20 | 3.1 | — | 3 | 583.16 | 0.8 | — | 3 | 624.98 | 0.2 | — | 1 |
| 026_B_p_two | 25 | 359.67 | 30.9 | — | 4 | 367.80 | 1.4 | — | 2 | 386.91 | 0.1 | — | 1 |
| 026_B_half | 25 | 359.67 | 6.5 | — | 2 | 367.80 | 2.0 | — | 2 | 386.91 | 0.1 | — | 1 |
| 026_B_one | 25 | 359.67 | 4.9 | — | 2 | 367.80 | 1.3 | — | 2 | 386.91 | 0.1 | — | 1 |
| 026_B_two | 25 | 359.67 | 3.9 | — | 2 | 367.80 | 1.7 | — | 2 | 386.91 | 0.1 | — | 1 |
| 030_B_p_two | 29 | 489.59 | 7.0 | — | 1 | 499.22 | 0.3 | — | 1 | 517.65 | 0.1 | — | 1 |
| 030_B_half | 29 | 493.68 | 6.2 | — | 1 | 499.22 | 0.3 | — | 1 | 517.65 | 0.1 | — | 1 |
| 030_B_one | 29 | 493.68 | 3.6 | — | 1 | 499.22 | 0.3 | — | 1 | 517.65 | 0.1 | — | 1 |
| 030_B_two | 29 | 493.68 | 1.9 | — | 1 | 499.22 | 0.3 | — | 1 | 517.65 | 0.1 | — | 1 |
| 031_B_p_two | 30 | 347.69 | 15.0 | — | 1 | 354.03 | 1.5 | — | 1 | 366.89 | 1.3 | — | 1 |
| 031_B_half | 30 | 350.64 | 8.0 | — | 1 | 357.66 | 5.3 | — | 1 | 368.38 | 1.0 | — | 1 |
| 031_B_one | 30 | 350.64 | 12.6 | — | 1 | 358.86 | 5.4 | — | 1 | 368.38 | 1.4 | — | 1 |
| 031_B_two | 30 | 350.64 | 4.4 | — | 1 | 358.86 | 3.9 | — | 1 | 368.38 | 1.0 | — | 1 |

**Table 31 Detailed results for Table 8: GLS-Multi set, $\alpha = 0.5$, $32 \leq n \leq 50$ ($n = |PD|$)).**

| | | $k=3$ | | | | $k=4$ | | | | $k=5$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | UB | sec | gap | iter | UB | sec | gap | iter | UB | sec | gap | iter |
| 033_B_p_two | 32 | 638.53 | 342.6 | — | 4 | 700.64 | 21.0 | — | 1 | 804.60 | 7.9 | — | 1 |
| 033_B_half | 32 | 668.55 | t.lim. | 1.67 | 6 | 716.36 | 130.1 | — | 3 | 807.83 | 71.3 | — | 4 |
| 033_B_one | 32 | 675.58 | 1095.0 | — | 7 | 736.40 | 271.4 | — | 3 | 807.83 | 70.2 | — | 4 |
| 033_B_two | 32 | 675.58 | 68.4 | — | 5 | 739.68 | 114.9 | — | 4 | 807.83 | 61.4 | — | 4 |
| 036_B_p_two | 35 | 379.92 | 69.4 | — | 1 | 383.65 | 3.6 | — | 1 | 393.85 | 2.0 | — | 1 |
| 036_B_half | 35 | 382.62 | 31.4 | — | 1 | 388.26 | 18.0 | — | 1 | 397.25 | 2.4 | — | 1 |
| 036_B_one | 35 | 382.62 | 9.8 | — | 1 | 389.96 | 13.2 | — | 1 | 397.25 | 2.1 | — | 1 |
| 036_B_two | 35 | 382.62 | 10.5 | — | 1 | 389.96 | 11.4 | — | 1 | 397.25 | 2.1 | — | 1 |
| 041_B_p_two | 40 | 395.57 | 216.1 | — | 1 | 402.43 | 72.1 | — | 1 | 413.29 | 14.9 | — | 1 |
| 041_B_half | 40 | 397.59 | 75.4 | — | 1 | 407.90 | 219.9 | — | 1 | 415.10 | 18.4 | — | 1 |
| 041_B_one | 40 | 397.59 | 59.8 | — | 1 | 407.90 | 87.0 | — | 1 | 415.10 | 22.3 | — | 1 |
| 041_B_two | 40 | 397.59 | 39.9 | — | 1 | 407.90 | 85.6 | — | 1 | 415.10 | 18.2 | — | 1 |
| 045_B_p_two | 44 | 643.48 | 10.4 | — | 2 | 643.48 | 25.9 | — | 2 | 648.06 | 0.5 | — | 1 |
| 045_B_half | 44 | 643.48 | 12.7 | — | 2 | 643.48 | 8.8 | — | 2 | 648.06 | 0.5 | — | 1 |
| 045_B_one | 44 | 643.48 | 37.6 | — | 3 | 643.48 | 15.8 | — | 2 | 648.06 | 0.5 | — | 1 |
| 045_B_two | 44 | 643.48 | 26.8 | — | 2 | 643.48 | 13.4 | — | 2 | 648.06 | 0.5 | — | 1 |
| 048_B_p_two | 47 | 36552.49 | 240.2 | — | 5 | 37290.99 | 123.1 | — | 5 | 38036.39 | 15.0 | — | 1 |
| 048_B_half | 47 | 36552.49 | 126.2 | — | 5 | 37290.99 | 98.6 | — | 5 | 38036.39 | 7.1 | — | 1 |
| 048_B_one | 47 | 36552.49 | 173.3 | — | 5 | 37290.99 | 121.9 | — | 5 | 38036.39 | 6.5 | — | 1 |
| 048_B_two | 47 | 36552.49 | 185.0 | — | 5 | 37290.99 | 131.1 | — | 5 | 38036.39 | 6.4 | — | 1 |
| 051_B_p_two | 50 | 455.40 | 33.6 | — | 1 | 459.48 | 10.2 | — | 1 | 474.05 | 44.2 | — | 1 |
| 051_B_half | 50 | 455.40 | 32.3 | — | 1 | 459.48 | 5.4 | — | 1 | 474.05 | 32.0 | — | 1 |
| 051_B_one | 50 | 455.40 | 51.5 | — | 1 | 459.48 | 8.4 | — | 1 | 474.05 | 26.2 | — | 1 |
| 051_B_two | 50 | 455.40 | 16.4 | — | 1 | 459.48 | 13.2 | — | 1 | 474.05 | 16.0 | — | 1 |

**Table 32 Detailed results for Table 8: GLS-Multi set, $\alpha = 0.5$, $71 \leq n \leq 100$ ($n = |PD|$)).**

| | | $k=3$ | | | | $k=4$ | | | | $k=5$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | UB | sec | gap | iter | UB | sec | gap | iter | UB | sec | gap | iter |
| 072_B_p_two | 71 | 220.37 | 35.0 | — | 1 | 231.75 | 92.8 | — | 1 | 243.70 | 18.7 | — | 1 |
| 072_B_half | 71 | 220.37 | 23.9 | — | 1 | 231.75 | 32.5 | — | 1 | 243.70 | 14.1 | — | 1 |
| 072_B_one | 71 | 220.37 | 46.3 | — | 1 | 231.75 | 90.1 | — | 1 | 243.70 | 11.3 | — | 1 |
| 072_B_two | 71 | 220.37 | 21.5 | — | 1 | 231.75 | 77.4 | — | 1 | 243.70 | 16.2 | — | 1 |
| 076_B_p_two | 75 | 571.03 | 3397.4 | — | 1 | 576.01 | 1314.7 | — | 1 | 584.32 | 2451.0 | — | 1 |
| 076_B_half | 75 | 571.03 | t.lim. | 0.56 | 1 | 576.84 | 1455.6 | — | 1 | 585.44 | 2601.7 | — | 1 |
| 076_B_one | 75 | 571.03 | 522.8 | — | 1 | 576.84 | 410.2 | — | 1 | 585.44 | 590.7 | — | 1 |
| 076_B_two | 75 | 571.03 | 477.3 | — | 1 | 576.84 | 418.7 | — | 1 | 585.44 | 660.3 | — | 1 |
| 101_B_p_two_a | 100 | 661.69 | 507.1 | — | 1 | 670.12 | 97.7 | — | 1 | 679.56 | 428.2 | — | 1 |
| 101_B_half_a | 100 | 661.69 | 308.8 | — | 1 | 670.12 | 187.4 | — | 1 | 679.56 | 302.0 | — | 1 |
| 101_B_one_a | 100 | 661.69 | 399.4 | — | 1 | 670.12 | 274.3 | — | 1 | 679.56 | 441.1 | — | 1 |
| 101_B_two_a | 100 | 661.69 | 452.1 | — | 1 | 670.12 | 187.1 | — | 1 | 679.56 | 163.3 | — | 1 |
| 101_B_p_two_b | 100 | 553.34 | t.lim. | 0.73 | 1 | 571.84 | t.lim. | 0.81 | 1 | 592.21 | t.lim. | 0.18 | 2 |
| 101_B_half_b | 100 | 553.23 | t.lim. | 0.91 | 1 | 571.92 | t.lim. | 0.48 | 1 | 592.21 | 3435.7 | — | 2 |
| 101_B_one_b | 100 | 553.26 | t.lim. | 0.36 | 2 | 573.21 | t.lim. | 0.74 | 1 | 592.21 | 3206.9 | — | 2 |
| 101_B_two_b | 100 | 553.23 | t.lim. | 0.41 | 2 | 572.29 | t.lim. | 0.26 | 2 | 592.21 | 3065.0 | — | 2 |

**Table 33 Detailed results for Table 8: GLS-Multi set, $\alpha = 0.4$, $15 \leq n \leq 30$ ($n = |PD|$)).**

| | | $k=3$ | | | | $k=4$ | | | | $k=5$ | | | | $k=6$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | UB | sec | gap | iter | UB | sec | gap | iter | UB | sec | gap | iter | UB | sec | gap | iter |
| 016_B_p_two | 15 | 263.16 | 1.7 | — | 2 | 272.95 | 0.1 | — | 1 | 290.87 | 0.1 | — | 1 | 313.41 | 0.0 | — | 1 |
| 016_B_half | 15 | 268.54 | 2.9 | — | 3 | 272.95 | 0.1 | — | 1 | 290.87 | 0.1 | — | 1 | 313.41 | 0.0 | — | 1 |
| 016_B_one | 15 | 268.54 | 2.4 | — | 3 | 272.95 | 0.1 | — | 1 | 290.87 | 0.1 | — | 1 | 313.41 | 0.0 | — | 1 |
| 016_B_two | 15 | 268.54 | 3.0 | — | 3 | 272.95 | 0.1 | — | 1 | 290.87 | 0.1 | — | 1 | 313.41 | 0.0 | — | 1 |
| 021_B_p_two | 20 | 305.77 | 0.9 | — | 1 | 316.49 | 0.9 | — | 2 | 329.00 | 0.4 | — | 1 | 353.06 | 0.1 | — | 1 |
| 021_B_half | 20 | 315.95 | 190.1 | — | 6 | 320.22 | 0.3 | — | 1 | 329.00 | 0.3 | — | 1 | 353.06 | 0.1 | — | 1 |
| 021_B_one | 20 | 315.95 | 90.5 | — | 6 | 320.22 | 0.3 | — | 1 | 329.00 | 0.6 | — | 1 | 353.06 | 0.1 | — | 1 |
| 021_B_two | 20 | 315.95 | 53.1 | — | 6 | 320.22 | 0.3 | — | 1 | 329.00 | 0.4 | — | 1 | 353.06 | 0.1 | — | 1 |
| 022_B_p_two | 21 | 340.71 | 2.5 | — | 2 | 345.64 | 1.9 | — | 2 | 357.10 | 0.6 | — | 2 | 369.62 | 0.0 | — | 1 |
| 022_B_half | 21 | 340.71 | 4.3 | — | 2 | 346.63 | 0.9 | — | 1 | 358.09 | 0.3 | — | 1 | 369.62 | 0.0 | — | 1 |
| 022_B_one | 21 | 340.71 | 3.7 | — | 1 | 346.63 | 0.6 | — | 1 | 358.09 | 0.3 | — | 1 | 369.62 | 0.0 | — | 1 |
| 022_B_two | 21 | 340.71 | 2.0 | — | 1 | 346.63 | 0.5 | — | 1 | 358.09 | 0.2 | — | 1 | 369.62 | 0.0 | — | 1 |
| 023_B_p_two | 22 | 550.50 | 5.8 | — | 3 | 583.16 | 0.8 | — | 3 | 624.98 | 0.2 | — | 1 | 668.46 | 0.1 | — | 1 |
| 023_B_half | 22 | 568.14 | 9.7 | — | 3 | 583.16 | 0.8 | — | 3 | 624.98 | 0.2 | — | 1 | 668.46 | 0.1 | — | 1 |
| 023_B_one | 22 | 569.20 | 4.7 | — | 3 | 583.16 | 0.8 | — | 3 | 624.98 | 0.2 | — | 1 | 668.46 | 0.1 | — | 1 |
| 023_B_two | 22 | 569.20 | 3.1 | — | 3 | 583.16 | 0.8 | — | 3 | 624.98 | 0.2 | — | 1 | 668.46 | 0.1 | — | 1 |
| 026_B_p_two | 25 | 362.31 | 1.1 | — | 1 | 375.24 | 0.2 | — | 1 | 394.35 | 0.9 | — | 1 | 421.02 | 0.5 | — | 1 |
| 026_B_half | 25 | 365.49 | 1.3 | — | 1 | 375.24 | 0.2 | — | 1 | 394.35 | 1.5 | — | 1 | 421.02 | 0.8 | — | 1 |
| 026_B_one | 25 | 365.49 | 0.9 | — | 1 | 375.24 | 0.3 | — | 1 | 394.35 | 0.9 | — | 1 | 421.02 | 0.7 | — | 1 |
| 026_B_two | 25 | 365.49 | 1.0 | — | 1 | 375.24 | 0.3 | — | 1 | 394.35 | 1.3 | — | 1 | 421.02 | 0.7 | — | 1 |
| 030_B_p_two | 29 | 493.68 | 2.2 | — | 1 | 505.00 | 0.3 | — | 1 | 523.43 | 0.3 | — | 1 | 544.99 | 0.3 | — | 1 |
| 030_B_half | 29 | 493.68 | 4.2 | — | 1 | 505.00 | 0.3 | — | 1 | 523.43 | 0.3 | — | 1 | 544.99 | 0.3 | — | 1 |
| 030_B_one | 29 | 493.68 | 1.8 | — | 1 | 505.00 | 0.3 | — | 1 | 523.43 | 0.3 | — | 1 | 544.99 | 0.3 | — | 1 |
| 030_B_two | 29 | 493.68 | 1.5 | — | 1 | 505.00 | 0.3 | — | 1 | 523.43 | 0.3 | — | 1 | 544.99 | 0.3 | — | 1 |
| 031_B_p_two | 30 | 356.27 | 14.0 | — | 2 | 366.64 | 4.8 | — | 2 | 379.10 | 6.0 | — | 1 | 391.96 | 2.7 | — | 1 |
| 031_B_half | 30 | 356.27 | 12.8 | — | 2 | 366.64 | 6.2 | — | 2 | 379.50 | 4.3 | — | 1 | 392.35 | 2.3 | — | 1 |
| 031_B_one | 30 | 356.27 | 14.7 | — | 2 | 366.64 | 9.1 | — | 2 | 379.50 | 4.5 | — | 1 | 392.35 | 1.8 | — | 1 |
| 031_B_two | 30 | 356.27 | 9.9 | — | 2 | 366.64 | 5.9 | — | 2 | 379.50 | 7.3 | — | 1 | 392.35 | 1.9 | — | 1 |

**Table 34 Detailed results for Table 8: GLS-Multi set, $\alpha = 0.4$, $32 \leq n \leq 50$ ($n = |PD|$)).**

| | | $k=3$ | | | | $k=4$ | | | | $k=5$ | | | | $k=6$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | UB | sec | gap | iter | UB | sec | gap | iter | UB | sec | gap | iter | UB | sec | gap | iter |
| 033_B_p_two | 32 | 683.70 | 17.6 | — | 1 | 751.18 | 8.9 | — | 1 | 822.72 | 6.6 | — | 1 | 928.22 | 7.0 | — | 1 |
| 033_B_half | 32 | 684.81 | 28.3 | — | 2 | 751.18 | 4.7 | — | 1 | 822.72 | 6.7 | — | 1 | 928.22 | 5.0 | — | 1 |
| 033_B_one | 32 | 684.81 | 81.0 | — | 2 | 751.18 | 5.5 | — | 1 | 822.72 | 7.7 | — | 1 | 928.22 | 6.3 | — | 1 |
| 033_B_two | 32 | 684.81 | 28.3 | — | 2 | 751.18 | 8.6 | — | 1 | 822.72 | 3.4 | — | 1 | 928.22 | 3.4 | — | 1 |
| 036_B_p_two | 35 | 387.94 | 58.5 | — | 1 | 396.00 | 18.7 | — | 1 | 408.17 | 6.4 | — | 1 | 421.03 | 3.2 | — | 1 |
| 036_B_half | 35 | 390.55 | 203.0 | — | 2 | 396.00 | 13.8 | — | 1 | 408.17 | 8.3 | — | 1 | 421.03 | 3.8 | — | 1 |
| 036_B_one | 35 | 390.55 | 123.1 | — | 2 | 396.00 | 19.5 | — | 1 | 408.17 | 5.3 | — | 1 | 421.03 | 4.4 | — | 1 |
| 036_B_two | 35 | 390.55 | 103.9 | — | 2 | 396.00 | 16.9 | — | 1 | 408.17 | 5.1 | — | 1 | 421.03 | 4.6 | — | 1 |
| 041_B_p_two | 40 | 401.88 | 138.0 | — | 1 | 411.73 | 49.9 | — | 1 | 423.34 | 50.0 | — | 1 | 435.57 | 38.6 | — | 1 |
| 041_B_half | 40 | 404.09 | 222.5 | — | 1 | 411.73 | 187.6 | — | 1 | 423.34 | 44.8 | — | 1 | 435.57 | 47.5 | — | 1 |
| 041_B_one | 40 | 404.09 | 158.9 | — | 1 | 411.73 | 78.2 | — | 1 | 423.34 | 42.3 | — | 1 | 435.57 | 37.7 | — | 1 |
| 041_B_two | 40 | 404.09 | 146.9 | — | 1 | 411.73 | 68.0 | — | 1 | 423.34 | 55.8 | — | 1 | 435.57 | 23.2 | — | 1 |
| 045_B_p_two | 44 | 665.54 | 156.3 | — | 2 | 662.40 | 17.6 | — | 1 | 666.98 | 24.4 | — | 1 | 672.48 | 3.2 | — | 1 |
| 045_B_half | 44 | 665.54 | 108.0 | — | 2 | 662.40 | 60.4 | — | 1 | 666.98 | 24.0 | — | 1 | 672.48 | 3.7 | — | 1 |
| 045_B_one | 44 | 665.54 | 112.7 | — | 2 | 662.40 | 6.6 | — | 1 | 666.98 | 6.0 | — | 1 | 672.48 | 2.3 | — | 1 |
| 045_B_two | 44 | 665.54 | 138.8 | — | 2 | 662.40 | 10.0 | — | 1 | 666.98 | 8.0 | — | 1 | 672.48 | 2.4 | — | 1 |
| 048_B_p_two | 47 | 37806.13 | t.lim. | 2.40 | 6 | 38436.23 | t.lim. | 2.02 | 7 | 38940.60 | 1170.7 | — | 6 | 40533.68 | 907.7 | — | 6 |
| 048_B_half | 47 | 37907.98 | t.lim. | 2.66 | 7 | 38076.73 | 2976.8 | — | 7 | 38940.60 | 598.2 | — | 6 | 40533.68 | 653.6 | — | 6 |
| 048_B_one | 47 | 37331.33 | t.lim. | 0.02 | 7 | 38076.73 | 3444.3 | — | 7 | 38940.60 | 790.8 | — | 6 | 40533.68 | 773.8 | — | 6 |
| 048_B_two | 47 | 37331.33 | t.lim. | 1.38 | 7 | 38076.73 | 2680.0 | — | 7 | 38940.60 | 695.4 | — | 6 | 40533.68 | 737.3 | — | 6 |
| 051_B_p_two | 50 | 462.15 | 101.1 | — | 1 | 466.23 | 42.3 | — | 1 | 480.04 | 62.0 | — | 1 | 494.01 | 54.4 | — | 1 |
| 051_B_half | 50 | 463.26 | 116.4 | — | 1 | 467.34 | 87.2 | — | 1 | 480.04 | 97.2 | — | 1 | 494.01 | 60.3 | — | 1 |
| 051_B_one | 50 | 463.26 | 103.9 | — | 1 | 467.34 | 62.6 | — | 1 | 480.04 | 97.1 | — | 1 | 494.01 | 41.8 | — | 1 |
| 051_B_two | 50 | 463.26 | 106.7 | — | 1 | 467.34 | 41.4 | — | 1 | 480.04 | 73.7 | — | 1 | 494.01 | 30.4 | — | 1 |

**Table 35 Detailed results for Table 8: GLS-Multi set, $\alpha = 0.4$, $71 \leq n \leq 100$ ($n = |PD|$)).**

| | | $k=3$ | | | | $k=4$ | | | | $k=5$ | | | | $k=6$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | UB | sec | gap | iter | UB | sec | gap | iter | UB | sec | gap | iter | UB | sec | gap | iter |
| 072_B_p_two | 71 | 225.85 | 2695.2 | — | 3 | 235.47 | 181.6 | — | 1 | 246.87 | 63.7 | — | 1 | 259.16 | 59.5 | — | 1 |
| 072_B_half | 71 | 225.85 | 915.2 | — | 3 | 235.47 | 49.3 | — | 1 | 246.87 | 65.3 | — | 1 | 259.16 | 50.1 | — | 1 |
| 072_B_one | 71 | 225.85 | 544.9 | — | 3 | 235.47 | 73.1 | — | 1 | 246.87 | 81.0 | — | 1 | 259.16 | 36.4 | — | 1 |
| 072_B_two | 71 | 225.85 | 785.8 | — | 3 | 235.47 | 94.9 | — | 1 | 246.87 | 63.7 | — | 1 | 259.16 | 40.4 | — | 1 |
| 076_B_p_two | 75 | 575.37 | 2404.3 | — | 1 | 581.18 | 1443.1 | — | 1 | 590.74 | t.lim. | 0.20 | 1 | 600.54 | 534.0 | — | 1 |
| 076_B_half | 75 | 575.88 | 1964.4 | — | 1 | 581.33 | 1358.8 | — | 1 | 590.74 | 2147.2 | — | 1 | 600.54 | 654.1 | — | 1 |
| 076_B_one | 75 | 575.88 | 1841.4 | — | 1 | 581.33 | 1526.4 | — | 1 | 590.74 | 2312.7 | — | 1 | 600.54 | 865.3 | — | 1 |
| 076_B_two | 75 | 575.88 | 1533.2 | — | 1 | 581.33 | 1050.3 | — | 1 | 590.74 | 1296.5 | — | 1 | 600.54 | 733.5 | — | 1 |
| 101_B_p_two_a | 100 | 667.50 | 740.4 | — | 1 | 674.58 | 469.7 | — | 1 | 682.35 | 199.2 | — | 1 | 693.94 | 142.3 | — | 1 |
| 101_B_half_a | 100 | 667.58 | 581.4 | — | 1 | 674.67 | 934.8 | — | 1 | 682.44 | 254.9 | — | 1 | 693.94 | 218.8 | — | 1 |
| 101_B_one_a | 100 | 667.58 | 767.2 | — | 1 | 674.67 | 485.2 | — | 1 | 682.44 | 248.3 | — | 1 | 693.94 | 76.8 | — | 1 |
| 101_B_two_a | 100 | 667.58 | 423.7 | — | 1 | 674.67 | 1067.5 | — | 1 | 682.44 | 549.2 | — | 1 | 693.94 | 100.0 | — | 1 |
| 101_B_p_two_b | 100 | 555.73 | 2971.1 | — | 1 | 573.93 | 2288.5 | — | 1 | 594.30 | 1495.5 | — | 1 | 614.98 | 676.4 | — | 1 |
| 101_B_half_b | 100 | 560.93 | t.lim. | 0.89 | 1 | 575.14 | 2621.5 | — | 1 | 594.61 | 1795.5 | — | 1 | 614.98 | 513.0 | — | 1 |
| 101_B_one_b | 100 | 556.38 | 2850.5 | — | 1 | 575.14 | 2817.1 | — | 1 | 594.61 | 1680.7 | — | 1 | 614.98 | 504.6 | — | 1 |
| 101_B_two_b | 100 | 556.38 | 1930.9 | — | 1 | 578.07 | t.lim. | 0.56 | 1 | 594.61 | 1409.4 | — | 1 | 614.98 | 696.6 | — | 1 |

**Table 36 Detailed results for GLS-Multi set and very tight capacities, $15 \leq n \leq 30$ ($n = |PD|$).**

| | | $\alpha = 0.52, k = 2$ | | | | $\alpha = 0.35, k = 3$ | | | | $\alpha = 0.27, k = 4$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | UB | sec | gap | iter | UB | sec | gap | iter | UB | sec | gap | iter |
| 016_B_p_two | 15 | 246.23 | 0.33 | — | 1 | 280.5 | 368.24 | — | 7 | 305.21 | 761.12 | — | 7 |
| 016_B_half | 15 | 264.3 | 2.39 | — | 2 | 299.03 | 868.12 | — | 6 | 320.12 | 1435.10 | — | 6 |
| 016_B_one | 15 | 278.57 | 1.99 | — | 2 | 312.8 | 589.25 | — | 5 | 333.89 | 800.86 | — | 4 |
| 016_B_two | 15 | 306.1 | 1.58 | — | 2 | 340.33 | 807.45 | — | 5 | 361.42 | 972.17 | — | 4 |
| 021_B_p_two | 20 | 292.96 | 17.83 | — | 2 | 327.77 | 1279.75 | — | 4 | 358.91 | 1062.58 | — | 3 |
| 021_B_half | 20 | 310.74 | 5.16 | — | 2 | 355.84 | t.lim. | 1.58 | 5 | 426.71 | t.lim. | 10.47 | 4 |
| 021_B_one | 20 | 326.36 | 7.47 | — | 2 | 463.43 | t.lim. | 21.06 | 5 | 473.57 | t.lim. | 16.57 | 3 |
| 021_B_two | 20 | 357.59 | 5.11 | — | 2 | 404.38 | t.lim. | 1.81 | 5 | 567.28 | t.lim. | 24.40 | 4 |
| 022_B_p_two | 21 | 303.09 | 6.80 | — | 2 | 357.72 | 46.34 | — | 2 | 385.02 | 152.39 | — | 3 |
| 022_B_half | 21 | 327.5 | 7.36 | — | 2 | 379.58 | 74.24 | — | 2 | 416.1 | t.lim. | 1.86 | 3 |
| 022_B_one | 21 | 348.8 | 5.53 | — | 2 | 417.58 | t.lim. | 1.73 | 5 | 529.12 | t.lim. | 19.38 | 3 |
| 022_B_two | 21 | 385.22 | 4.92 | — | 2 | 460.51 | t.lim. | 2.98 | 5 | 472.71 | t.lim. | 2.06 | 3 |
| 023_B_p_two | 22 | 547.05 | 19.12 | — | 4 | 550.5 | 6.66 | — | 3 | 583.16 | 0.89 | — | 3 |
| 023_B_half | 22 | 567.59 | 20.53 | — | 4 | 568.14 | 9.63 | — | 3 | 583.16 | 0.86 | — | 3 |
| 023_B_one | 22 | 596.99 | 16.46 | — | 4 | 569.2 | 4.70 | — | 3 | 583.16 | 0.87 | — | 3 |
| 023_B_two | 22 | 655.79 | 14.20 | — | 4 | 569.2 | 3.21 | — | 3 | 583.16 | 0.85 | — | 3 |
| 026_B_p_two | 25 | 343.73 | 16.23 | — | 2 | 376.54 | 21.88 | — | 1 | 421.48 | 438.70 | — | 3 |
| 026_B_half | 25 | 368.68 | 226.78 | — | 5 | 398.36 | 680.73 | — | 3 | 474.01 | t.lim. | 7.90 | 3 |
| 026_B_one | 25 | 386.22 | 180.77 | — | 5 | 415.79 | 136.60 | — | 3 | 508.89 | t.lim. | 10.79 | 3 |
| 026_B_two | 25 | 421.1 | 178.00 | — | 5 | 450.67 | 136.29 | — | 3 | 578.65 | t.lim. | 15.51 | 3 |
| 030_B_p_two | 29 | 488.95 | 1757.71 | — | 6 | 563.12 | t.lim. | 6.67 | 4 | 574.8 | 80.60 | — | 3 |
| 030_B_half | 29 | 547.19 | t.lim. | 6.85 | 8 | 597.32 | t.lim. | 9.41 | 4 | 593.69 | 15.67 | — | 2 |
| 030_B_one | 29 | 554.8 | t.lim. | 3.36 | 7 | 652.57 | t.lim. | 11.77 | 4 | 738.71 | t.lim. | 18.32 | 8 |
| 030_B_two | 29 | 634.93 | t.lim. | 6.67 | 8 | 763.06 | t.lim. | 18.23 | 4 | 611.55 | t.lim. | 1.05 | 8 |
| 031_B_p_two | 30 | 341.71 | 150.83 | — | 2 | 367.47 | 1461.52 | — | 3 | 383.5 | 529.68 | — | 2 |
| 031_B_half | 30 | 365.26 | 56.15 | — | 1 | 388.94 | t.lim. | 0.43 | 5 | 429.89 | t.lim. | 7.53 | 2 |
| 031_B_one | 30 | 385.21 | 82.00 | — | 3 | 411.5 | t.lim. | 1.09 | 4 | 483.79 | t.lim. | 14.46 | 2 |
| 031_B_two | 30 | 415.37 | 59.82 | — | 3 | 441.68 | t.lim. | 0.58 | 5 | 574.28 | t.lim. | 22.70 | 2 |

**Table 37 Detailed results for GLS-Multi set and very tight capacities, $32 \leq n \leq 50$ ($n = |PD|$).**

| | | $\alpha = 0.52, k = 2$ | | | | $\alpha = 0.35, k = 3$ | | | | $\alpha = 0.27, k = 4$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | UB | sec | gap | iter | UB | sec | gap | iter | UB | sec | gap | iter |
| 033_B_p_two | 32 | 645.42 | t.lim. | 11.11 | 8 | 693.95 | 121.64 | — | 2 | 901.78 | t.lim. | 6.78 | 2 |
| 033_B_half | 32 | 674.41 | t.lim. | 11.61 | 11 | 719.33 | 554.30 | — | 6 | 948.96 | t.lim. | 9.79 | 2 |
| 033_B_one | 32 | 626.95 | t.lim. | 1.73 | 10 | 739.36 | 225.96 | — | 6 | 1036.79 | t.lim. | 15.50 | 2 |
| 033_B_two | 32 | 1615.56 | t.lim. | 59.38 | 11 | 779.43 | 332.81 | — | 6 | 1116.93 | t.lim. | 17.98 | 2 |
| 036_B_p_two | 35 | 369.83 | 59.92 | — | 1 | 425.93 | t.lim. | 6.80 | 2 | 426.53 | t.lim. | 1.28 | 2 |
| 036_B_half | 35 | 398.8 | t.lim. | 0.19 | 5 | 459.04 | t.lim. | 8.86 | 2 | 482.02 | t.lim. | 11.76 | 1 |
| 036_B_one | 35 | 417.22 | 755.64 | — | 5 | 520.47 | t.lim. | 16.29 | 1 | 528.4 | t.lim. | 16.29 | 1 |
| 036_B_two | 35 | 447.28 | 386.98 | — | 5 | 640.74 | t.lim. | 27.90 | 1 | 618.61 | t.lim. | 24.37 | 1 |
| 041_B_p_two | 40 | 386.64 | 1253.25 | — | 2 | 426.48 | t.lim. | 5.45 | 1 | 465.49 | t.lim. | 8.33 | 1 |
| 041_B_half | 40 | 425.55 | 990.11 | — | 3 | 474.11 | t.lim. | 7.48 | 1 | 527.95 | t.lim. | 16.12 | 1 |
| 041_B_one | 40 | 473.15 | t.lim. | 3.44 | 5 | 533.37 | t.lim. | 12.68 | 1 | 604.47 | t.lim. | 23.94 | 1 |
| 041_B_two | 40 | 547.71 | t.lim. | 5.32 | 6 | 655.83 | t.lim. | 19.81 | 1 | 727.19 | t.lim. | 33.03 | 1 |
| 045_B_p_two | 44 | 681.46 | 2159.22 | — | 5 | 765 | t.lim. | 4.59 | 3 | 877.09 | t.lim. | 13.29 | 1 |
| 045_B_half | 44 | 697.61 | 241.23 | — | 5 | 967.82 | t.lim. | 22.69 | 2 | 952.95 | t.lim. | 18.59 | 1 |
| 045_B_one | 44 | 724.52 | 174.67 | — | 5 | 1050.16 | t.lim. | 27.49 | 1 | 1033.31 | t.lim. | 22.27 | 1 |
| 045_B_two | 44 | 778.35 | 155.31 | — | 5 | 1211.67 | t.lim. | 31.98 | 1 | 1194.8 | t.lim. | 28.29 | 1 |
| 048_B_p_two | 47 | 36954.83 | 2209.98 | — | 5 | 42404.33 | t.lim. | 10.86 | 1 | 49813.87 | t.lim. | 19.04 | 1 |
| 048_B_half | 47 | 38843.26 | t.lim. | 0.84 | 6 | 45579.48 | t.lim. | 14.90 | 1 | 53317.5 | t.lim. | 23.97 | 1 |
| 048_B_one | 47 | 41928.33 | t.lim. | 0.21 | 6 | 53791.08 | t.lim. | 24.95 | 1 | 59886.76 | t.lim. | 33.49 | 1 |
| 048_B_two | 47 | 48497.61 | t.lim. | 0.27 | 6 | 70214.28 | t.lim. | 37.50 | 2 | 73025.32 | t.lim. | 44.35 | 1 |
| 051_B_p_two | 50 | 454.62 | 1130.83 | — | 2 | 503.51 | t.lim. | 6.30 | 1 | 531.06 | t.lim. | 8.07 | 1 |
| 051_B_half | 50 | 489.19 | 1805.59 | — | 1 | 560.11 | t.lim. | 10.35 | 1 | 579.14 | t.lim. | 12.94 | 1 |
| 051_B_one | 50 | 516.91 | t.lim. | 2.04 | 2 | 624.95 | t.lim. | 16.88 | 1 | 640.08 | t.lim. | 18.60 | 1 |
| 051_B_two | 50 | 686.43 | t.lim. | 21.50 | 2 | 755.29 | t.lim. | 27.02 | 1 | 737.36 | t.lim. | 24.79 | 1 |

**Table 38 Detailed results for GLS-Multi set and very tight capacities, $71 \leq n \leq 100$ ($n = |PD|$)).**

| | | $\alpha = 0.52, k = 2$ | | | | $\alpha = 0.35, k = 3$ | | | | $\alpha = 0.27, k = 4$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| instance | $\|PD\|$ | UB | sec | gap | iter | UB | sec | gap | iter | UB | sec | gap | iter |
| 072_B_p_two | 71 | 224.18 | 1926.29 | — | 2 | 258.86 | t.lim. | 7.12 | 1 | 271.43 | t.lim. | 8.65 | 1 |
| 072_B_half | 71 | 231.57 | t.lim. | 0.78 | 3 | 325.1 | t.lim. | 24.34 | 1 | 291.42 | t.lim. | 12.37 | 1 |
| 072_B_one | 71 | 241.23 | t.lim. | 0.93 | 3 | 333.16 | t.lim. | 23.56 | 1 | 328.57 | t.lim. | 23.54 | 1 |
| 072_B_two | 71 | 259.24 | t.lim. | 0.72 | 5 | 369.39 | t.lim. | 26.10 | 1 | 381.51 | t.lim. | 26.23 | 1 |
| 076_B_p_two | 75 | 588.98 | t.lim. | 4.41 | 1 | 631.77 | t.lim. | 9.28 | 1 | 640.11 | t.lim. | 7.39 | 1 |
| 076_B_half | 75 | 678.73 | t.lim. | 5.99 | 1 | 720 | t.lim. | 11.16 | 1 | 706.07 | t.lim. | 11.04 | 1 |
| 076_B_one | 75 | 856.75 | t.lim. | 19.71 | 1 | 803.06 | t.lim. | 16.18 | 1 | 806.17 | t.lim. | 19.00 | 1 |
| 076_B_two | 75 | 804.67 | t.lim. | 2.07 | 1 | 969.2 | t.lim. | 23.73 | 1 | 1005.55 | t.lim. | 31.62 | 1 |
| 101_B_p_two_a | 100 | 713.51 | t.lim. | 8.56 | 1 | 743.83 | t.lim. | 10.75 | 1 | 732.15 | t.lim. | 6.20 | 1 |
| 101_B_half_a | 100 | 848.43 | t.lim. | 12.71 | 1 | 846.82 | t.lim. | 21.95 | 1 | 849.06 | t.lim. | 15.01 | 1 |
| 101_B_one_a | 100 | 972.24 | t.lim. | 20.12 | 1 | 1016.6 | t.lim. | 34.94 | 1 | 922.52 | t.lim. | 20.57 | 1 |
| 101_B_two_a | 100 | 1175.94 | t.lim. | 24.37 | 1 | 1356.1 | t.lim. | 51.27 | 1 | 1058.32 | t.lim. | 27.61 | 1 |
| 101_B_p_two_b | 100 | 629.58 | t.lim. | 14.60 | 1 | 641.82 | t.lim. | 12.61 | 1 | 680.05 | t.lim. | 15.06 | 1 |
| 101_B_half_b | 100 | 712.97 | t.lim. | 21.02 | 2 | 738.21 | t.lim. | 22.36 | 1 | 719.77 | t.lim. | 20.18 | 1 |
| 101_B_one_b | 100 | 753.43 | t.lim. | 22.99 | 1 | 792.05 | t.lim. | 26.62 | 1 | 759.24 | t.lim. | 23.93 | 1 |
| 101_B_two_b | 100 | 832.37 | t.lim. | 25.24 | 2 | 910.46 | t.lim. | 29.91 | 1 | 838.18 | t.lim. | 31.09 | 1 |

## References

Applegate, D.L., R.E. Bixby, V. Chvátal, W.J. Cook. 2006. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press.

Baldacci, R., A. Mingozzi, R. Roberti. 2011. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research* **59** 1269–1283.

Battarra, M., J.-F. Cordeau, M. Iori. 2014. Pickup and delivery problems for goods transportation. P. Toth, D. Vigo, eds., *Vehicle Routing: Problems, Methods, and Applications*, 2nd ed. MOS-SIAM Series on Optimization, SIAM, 161–192.

Benders, J.F. 1962. Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik* **4** 238–252.

Berbeglia, G., J.-F. Cordeau, I. Gribkovskaia, G. Laporte. 2007. Static pickup and delivery problems: A classification scheme and survey. *TOP* **15** 1–31.

Bruck, B.P., A.G. dos Santos. 2012. Hybrid approach for the multiple vehicle routing problem with deliveries and selective pickups. *12th International Conf. on Hybrid Intelligent Systems, (HIS 2012)*. 265–270.

Bruck, B.P., A.G. dos Santos, J.E.C. Arroyo. 2012. Metaheuristics for the single vehicle routing problem with deliveries and selective pickups. *12th International Conference on Intelligent Systems Design and Applications (ISDA 2012)*. 723–728.

Chemla, D., F. Meunier, R. Wolfler Calvo. 2013. Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization* **10** 120–146.

Codato, G., M. Fischetti. 2006. Combinatorial benders' cuts for mixed-integer linear programming. *Operations Research* **54** 756–766.

Coelho, I.M., P.L.A. Munhoz, M.N. Haddad, M.J.F. Souza, L.S. Ochi. 2012. A hybrid heuristic based on general variable neighborhood search for the single vehicle routing problem with deliveries and selective pickups. *Electronic Notes in Discrete Mathematics* **39** 99–106.

Cordeau, J.F., M. Dell'Amico, S. Falavigna, M. Iori. 2015. A rolling horizon algorithm for auto-carrier transportation. *Transportation Research Part B* **76** 68–80.

Côté, J.-F., M. Dell'Amico, M. Iori. 2014. Combinatorial benders' cuts for the strip packing problem. *Operations Research* **62** 643–661.

Daniel, V., R. Guide Jr., L.N. Van Wassenhove. 2009. The evolution of closed-loop supply chain research. *Operations Research* **57** 10–18.

Doerner, K., J.-J. Salazar-González. 2014. Pickup and delivery routing problems for people transportation. P. Toth, D. Vigo, eds., *Vehicle Routing: Problems, Methods, and Applications*, 2nd ed. MOS-SIAM Series on Optimization, SIAM, 193–212.

Feillet, D., P. Dejax, M. Gendreau. 2001. Traveling salesman problems with profits: An overview. *Transportation Science* **39** 188–205.

Golden, B.L., A.A. Assad. 1986. OR forum – perspectives on vehicle routing: Exciting new developments. *Operations Research* **34**(5) 803–810.

Golden, B.L., A.A. Assad, E.A. Wasil. 2002. Routing vehicles in the real world: Applications in the solid waste, beverage, food, dairy, and newspaper industries. P. Toth, D. Vigo, eds., *The Vehicle Routing Problem*. SIAM, Philadelphia, PA, 245–286.

Gribkovskaia, I., Ø. Halskau, G. Laporte, M. Vlcek. 2007. General solutions to the single vehicle routing problem with pickups and deliveries. *European Journal of Operational Research* **180**(2) 568–584.

Gribkovskaia, I., G. Laporte. 2008. One-to-many-to-one single vehicle pickup and delivery problems. B. Golden, S. Raghavan, E. Wasil, R. Sharda, S. Voss, eds., *The Vehicle Routing Problem: Latest Advances and New Challenges*, vol. 43. Springer, Berlin, 359–377.

Gribkovskaia, I., G. Laporte, A. Shyshou. 2008. The single vehicle routing problem with deliveries and selective pickups. *Computers & Operations Research* **35** 2908–2924.

Gutiérrez-Jarpa, G., G. Desaulniers, G. Laporte, V. Marianov. 2010. A branch-and-price algorithm for the vehicle routing problem with deliveries, selective pickups and time windows. *European Journal of Operational Research* **206** 341–349.

Gutiérrez-Jarpa, G., V. Marianov, C. Obreque. 2009. A single vehicle routing problem with fixed delivery and optional collections. *IIE Transactions* **41** 1067–1079.

Hernández-Pérez, H., J.-J. Salazar-González. 2004. A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery. *Discrete Applied Mathematics* **145** 126–139.

Irnich, S., C. Laganà, D.and Schlebusch, F. Vocaturo. 2015. Two-phase branch-and-cut for the mixed capacitated general routing problem. *European Journal of Operational Research* **243** 17–29.

Jepsen, M., B. Petersen, S. Spoorendonk, D. Pisinger. 2008. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research* **56** 497–511.

Kaparis, K., A. Letchford. 2010. Separation algorithms for 0-1 knapsack polytopes. *Mathematical Programming* **124** 69–91.

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

Miller, C.E., A.W. Tucker, R.A. Zemlin. 1960. Integer programming formulations and traveling salesman problems. *Journal of the ACM* **7** 326–329.

Nagy, G., S. Salhi. 2005. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research* **162** 126–141.

Nowak, M., Ö. Ergun, C.C. White. 2008. Pickup and delivery with split loads. *Transportation Science* **42** 32–43.

Padberg, M., G. Rinaldi. 1991. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review* **33** 60–100.

Pecin, D., A. Pessoa, M. Poggi, E. Uchoa. 2016. Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation* (forthcoming).

Pferschy, U., R. Staněk. 2016. Generating subtour elimination constraints for the TSP from pure integer solutions. *Central European Journal of Operations Research* (forthcoming).

Privé, J., J. Renaud, F. Boctor, G. Laporte. 2006. Solving a vehicle-routing problem arising in soft-drink distribution. *Journal of Operational Research Society* **57** 1045–1052.

Righini, G., M. Salani. 2008. New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks* **51** 155–170.

Salazar-González, J.-J., B. Santos-Hernández. 2015. The split-demand one-commodity pickup-and-delivery travelling salesman problem. *Transportation Research Part B* **75** 58–73.

Subramanian, A., E. Uchoa, A.A. Pessoa, L.S. Ochi. 2011. Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery. *Operations Research Letters* **39** 338–341.

Süral, H., J.H. Bookbinder. 2003. The single-vehicle routing problem with unrestricted backhauls. *Networks* **41** 127–136.

Toth, P., D. Vigo. 2002. An overview of vehicle routing problems. P. Toth, D. Vigo, eds., *The Vehicle Routing Problem*. SIAM, Philadelphia, PA, 1–24.

Van Roy, T.J., L.A. Wolsey. 1987. Solving mixed integer programming problems using automatic reformulation. *Operations Research* **35** 45–57.

Vigo, D. 1999. VRPLIB: A Vehicle Routing Problem LIBrary. Tech. Rep. OR/99/9, DEIS, University of Bologna. `http://or.dei.unibo.it/library/vrplib-vehicle-routing-problem-library` (last accessed: 2016-12-15).

Wolsey, L.A. 1998. *Integer Programming*, *Wiley Series in Discrete Mathematics and Optimization*, vol. 52. Wiley, New York, NY.