

Formulations for the Split Delivery Capacitated Profitable Tour Problem

Marvin Caspar(✉), Daniel Schermer, and Oliver Wendt

Business Information Systems and Operations Research (BISOR), University of Kaiserslautern-Landau, Postfach 3049, 67653 Kaiserslautern, Germany
{marvin.caspar,daniel.schermer,wendt}@wiwi.uni-kl.de
https://wiwi.rptu.de/fgs/bisor

Abstract. In this paper, we study the *Split Delivery Capacitated Profitable Tour Problem* (SDCPTP) that extends the profitable tour problem by a fleet of vehicles and the possibility of split deliveries. In the SDCPTP, the objective is to select a subset of requests and find for each vehicle a feasible tour such that the prize collected by fulfilling the selected requests net of the cost of delivery is maximized. We propose three mixed-integer linear programming formulations for modeling the SDCPTP. These formulations differ primarily in their treatment of the subtour elimination constraints within the branch-and-cut algorithm. The first of these formulations is applicable to the more general asymmetric case, whereas the other two are computationally more efficient in presence of symmetric variants of the problem. Using benchmark instances from the literature, we provide a detailed numerical study that demonstrates the effectiveness of a state-of-the-art solver in tackling these models. For managerial insights, we perform a sensitivity analysis on the SDCPTP with respect to the number of vehicles and requests as well as the benefits of enabling split delivery.

Keywords: Mixed-Integer Linear Programming · Branch-and-Cut Algorithm · Capacitated Profitable Tour Problem · Split Delivery

1 Introduction

The *Traveling Salesman Problem* (TSP) is one of the most well-known combinatorial optimization problems that has driven the advancement of branch-and-cut algorithms for several decades [15]. Many practical problems can be expressed in terms of a TSP, and therefore it comes as no surprise that many variants of the problem exist. For example, in the context of scheduling and routing, an important generalization of the problem relaxes the standard assumption that every vertex must be visited exactly once (see, e. g., [8,16,18]). Instead, the vertices are viewed as requests that yield a *prize* (i. e., a reward) that is gained for fulfilling a given demand; and it falls to the decision maker to determine which requests to accept and in which order to process them. From the standpoint of operations

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023, LNCS 13956, pp. 82–98, 2023.
https://doi.org/10.1007/978-3-031-36805-9_6

research, this trade-off is known in the literature as *Traveling Salesman Problem with Profits* (TSPPs) (refer to [18]) or, in the case of multiple vehicles, *Vehicle Routing Problem with Profits* (VRPPs) (refer to [27]). In VRPPs, a standard assumption is that each demand must be fulfilled by at most one vehicle. However, especially in the fields of transportation and logistics, where demands may exceed the capacity of a single vehicle, there are many applications in which it becomes beneficial or even necessary to fulfill a demand by more than one vehicle. Such applications include, e. g., the distribution of livestock feed [24], waste management [7], routing helicopters for crew exchanges [25], or watering services [30]. In the literature, the assumption that a demand may be fulfilled by more than one vehicle is known as *split delivery*, which can make logistics more efficient and therefore also more sustainable. However, as addressed in [5] (and as we will see in more detail in Sect. 2), there is a scientific gap concerning the investigation of split delivery in the context of VRPPs.

In order to address this scientific gap, we study the *Split Delivery Capacitated Profitable Tour Problem* (SDCPTP), which is a variant of VRPPs (see [27]). In the SDCPTP, we are given a fleet of homogenous vehicles with a limited capacity and a set of requests. Each request is associated with a prize that is gained only if the corresponding demand is fully satisfied. The objective is to select a subset of requests and find a feasible tour for each vehicle (that visits some of the selected requests exactly once and then returns to the depot) such that the prize collected by fulfilling these requests net of the cost of delivery is maximized. By the nature of split delivery, accepted requests may be fulfilled jointly, i. e., by more than one vehicle.

In our paper, we make the following contributions. After a brief literature review, we provide three *Mixed-Integer Linear Programming* (MILP) formulations for the SDCPTP. These formulations differ primarily in their treatment of the subtour elimination constraints within the branch-and-cut algorithm. In particular, whereas the first of these formulations is applicable to the more general asymmetric case, the other two are computationally more efficient in presence of symmetric variants of the problem, which are commonly studied in the literature. Using these formulations, we provide a detailed numerical study that demonstrates the effectiveness of a state-of-the-art solver in tackling these models. For managerial insights, we perform a sensitivity analysis on the SDCPTP with respect to the number of vehicles and requests as well as the benefits of enabling split delivery.

The remainder of this paper is organized as follows. We begin with a brief overview of the related literature in Sect. 2. In particular, we discuss the foundations of the SDCPTP in more detail, i. e., the *Profitable Tour Problem* (PTP) and split delivery in the context of routing problems. Afterwards, in Sect. 3, we begin with a concise problem definition of the SDCPTP and provide the MILP formulations. In Sect. 4 we discuss the computational experiments and their numerical results. Finally, with Sect. 5, we draw concluding remarks on our work and discuss further research implications.

2 Related Literature

In what follows, we briefly address PTP problems in Sect. 2.1. Afterwards, in Sect. 2.2, we outline relevant works of routing problems with split delivery.

2.1 Profitable Tour Problems

Routing problems with profits are a family of problems in which each request is associated with a prize (i. e., a reward) that is gained only if the demand that is associated with that request is satisfied; and the arcs, that connect accepted requests in a feasible tour, are associated with a cost (see [18]). Within this family of problems, the essential characteristic is that it becomes necessary to trade off the cumulative prize against the cost that is incurred by traversing arcs. In general, we can differentiate three main variants of TSPPs:

- In the *Prize-Collecting TSP* (see [8,16]) there is a lower bound on the cumulative prize that must be collected. The objective then becomes to minimize the total travel costs while respecting the prize limit.
- In contrast, in the *Orienteering Problem* (refer to [28]) there is an upper bound on the costs (e. g., the length of the tour). Here, the objective is to maximize the collected prize by a tour that does not exceed the cost limit.
- In the *Profitable Tour Problem* (PTP) there is neither a lower bound on the total prize nor an upper bound on the costs. Instead, the objective consists in maximizing the profit, i. e., the cumulative prize net of the costs (see [16]).

For the purpose of this paper, we restrict ourselves to the PTP in what follows, as the inherent trade-off in that problem is of particular interest for a variety of routing and scheduling problems (see [27] and references therein). Furthermore, the PTP is also worthwhile because it relates to fundamental problems such as the elementary shortest path problem with resource constraints (refer to [22]). While most existing literature on the PTP does not consider a fleet of vehicles (for further references, see [27]), generalizations of the problem have become increasingly common in recent years.

In particular, the *Capacitated* PTP (CPTP) was introduced in [6] as a generalization of the PTP, in which we are given a capacitated fleet of vehicles. The authors introduced a branch-and-price algorithm, a variable neighborhood search algorithm, and two tabu search algorithms for solving the problem and their computational experiments show that the heuristic procedures often find the same optimal solutions as the exact method. Later, in [4], an improved branch-and-price algorithm is proposed that includes a modified branching rule and further acceleration techniques, to speed up the pricing problem. This enabled the authors to find optimal solutions to several benchmark instances that were previously unsolved. Apart from these fundamental works, there exist various extensions to the CPTP such as the introduction of time windows [26] or electric vehicles [13]. The latter variant has also been refined through the inclusion of multi-periods with recharge services and mandatory stops [14].

Further variants of the PTP include time-dependent travel times and resource constraints [23] or stochastic variants of the problem [1], in which accepted requests show up only with a given probability during the execution of the tour.

2.2 Split Delivery in Routing Problems

The assumption that a request may be fulfilled by more than one vehicle is known as *Split Delivery* (SD). The *Split Delivery Vehicle Routing Problem* (SDVRP) was first introduced by [17] and has been extensively analyzed in the last three decades. In particular, several worst-case analyses and empirical studies have shown that considerable savings can occur when split delivery is allowed [27]. Furthermore, several exact and heuristic algorithms have been proposed to effectively solve the SDVRP and its variants (refer to [27] and references therein). Such variants include, e. g., the inclusion of time windows [19], minimum delivery amounts [20], or three-dimensional loading constraints [11].

Naturally, SD may also be beneficial in the context of routing problems with profits, as this allows for more flexibility when it comes to fulfilling demands. To the best of our knowledge, [2,3,5,29] are the only contributions that consider SD within routing problems with profits. In [3] the authors investigate the *Split Delivery Capacitated Team Orienteering Problem* (SDCTOP) with the option of *Incomplete Service* (IS), i. e., the assumption that requests may only be partially served. Apart from theoretical contributions that analyze the maximum possible profit, the authors also propose a branch-and-price approach as well as a variable neighborhood search and tabu search heuristic and verify their effectiveness on benchmark instances for the SDCTOP and SDCTOP-IS. A detailed study in [2] revealed that the combination of IS and SD does not significantly increase the profits in the CTOP compared to only allowing either IS or SD. Furthermore, the experiments show that the largest increase in profits in the SDCTOP and the SDCTOP-IS is observed when the average demand exceeds half the vehicle capacity. In [5], a branch-and-price algorithm and a hybrid heuristic based on tabu search are used to solve four different sets of instances for the CTOP and SDCTOP. It was shown that, depending on the instances, the hybrid heuristic could be significantly more effective than branch-and-price in terms of computation time. In [29], the *SDCTOP with Minimum Delivery Amounts* (SDCTOP-MDA) is discussed, and in a worst-case analysis correlations between the CTOP and the SDCTOP-MDA are analyzed.

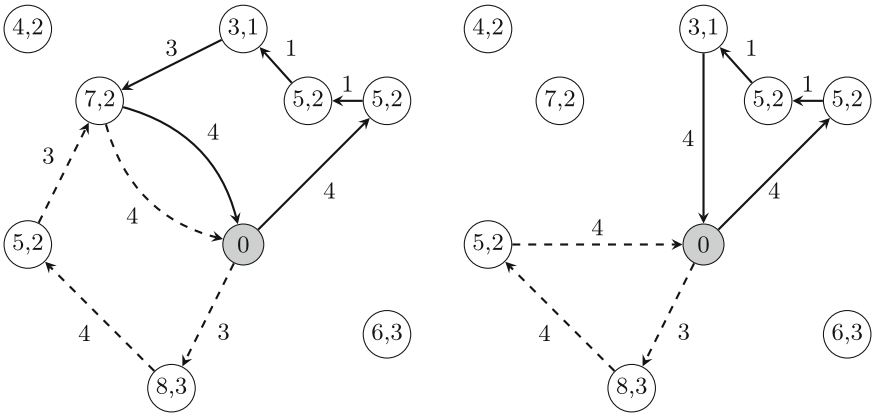
In general, these works demonstrate that the option of SD has been intensively studied in the context of team orienteering problems. However, to the best of our knowledge and as stated in [5], there is no literature on the CPTP that investigates SD. This provides further motivation for the SDCPTP presented in the following section.

3 Problem Definition and Mathematical Programming Formulations

In this section, we begin with a concise problem definition of the SDCPTP in Sect. 3.1. Afterwards, in Sect. 3.2, we present three mathematical programming formulations for the problem.

3.1 Problem Definition

Let $G(V, A)$ be a complete graph, where $V = \{0, 1, \ldots, |N|\}$ is the set of vertices and $A = \{(i, j) \,|\, i, j \in V (i \neq j)\}$ is the set of arcs. The vertex $0 \in V$ represents the depot and $N = \{1, \ldots, |N|\} = V \setminus \{0\}$ represents a set of requests. Each request $i \in N$ is associated with a prize $p_i \in \mathbb{R}_{>0}$ (i. e., a reward) that is only obtained if the corresponding demand $d_i \in \mathbb{N}$ is completely satisfied. A non-negative travel cost $c_{i,j}$ is associated with each arc $(i, j) \in A$ and we assume that the triangle inequality holds for these values. We say that the problem is symmetric only if $c_{i,j} = c_{j,i}$ holds for all $(i, j) \in A$. In the SDCPTP, we are given a fleet K of homogenous vehicles, each with a maximum capacity C. The tour of each vehicle $k \in K$ starts and ends at the depot and, in contrast to the CPTP, each request $i \in N$ may be served by more than one vehicle. The objective is to find feasible tours for the fleet such that the cumulative prize net of the cost of delivery is maximized. Figure 1 illustrates the problem.



(a) A SDCPTP solution. (b) A CPTP solution.

Fig. 1. An illustration of optimal SDCPTP (a) and CPTP (b) solutions on a problem with $|N| = 8$ requests and a fleet of two vehicles, each with a capacity of $C = 6$. The depot is labelled with 0 and the remaining vertices are labelled with a tuple (p_i, d_i). The tours are indicated by the solid and dashed arcs, respectively, and the costs $c_{i,j}$ are shown next to each arc. A single request receives a split delivery in (a), increasing the profit by 1 unit compared to (b).

3.2 Mathematical Programming Formulations

In this section, we propose three novel MILP formulations for the SDCPTP. The first model follows up on the work in [5] and the other two models draw some inspiration from [22]. In general, all of these formulations are three-index vehicle flow formulations that differ primarily in their treatment of the subtour elimination constraints within the branch-and-cut algorithm. Whereas the first of these formulations is applicable to the more general asymmetric case, the other two are only suitable in presence of symmetric variants of the problem, which are commonly studied in the literature. At the end of this section, we will also demonstrate how these formulations can be adjusted to model the CPTP, i. e., for special cases where split delivery is not allowed. The common notation for the models is summarized in Table 1.

Table 1. Common notation for the MILP models in Sect. 3.2.

N	Set of requests $N = \{1, ..., \|N\|\}$
V	Set of all vertices $V = \{0, ..., \|N\|\}$
A	Set of arcs in the complete graph $A = \{(i,j) \mid i,j \in V (i \neq j)\}$
K	Set of vehicles $K = \{1, ..., \|K\|\}$
C	Maximum capacity of each vehicle $k \in K$
p_i	Prize associated with request $i \in N$
d_i	Demand associated with request $i \in N$
$c_{i,j}$	Travel cost between two vertices $(i,j) \in A$
$x_{i,j,k}$	Binary variable, equals 1 iff arc $(i,j) \in A$ is part of the tour of vehicle $k \in K$
z_i	Binary variable, equals 1 iff request $i \in N$ is accepted
$y_{i,k}$	Binary variable, equals 1 iff request $i \in N$ is the only request that is served by vehicle $k \in K$. Iff $y_{0,k} = 1$ then vehicle k remains idle at the depot
$h_{i,k}$	Binary variable, equals 1 iff request $i \in N$ is visited by vehicle $k \in K$
$q_{i,k}$	Integer decision variable, indicates the quantity delivered to $i \in N$ by $k \in K$

The first MILP model for the SDCPTP is (1)–(11) (referred to as SDCPTP1 in what follows):

$$\max \quad \sum_{i \in N} p_i \cdot z_i - \sum_{k \in K} \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} c_{i,j} \cdot x_{i,j,k} \tag{1}$$

$$\sum_{j \in N} x_{0,j,k} \leq 1 \quad \forall k \in K \tag{2}$$

$$\sum_{\substack{j \in V \\ i \neq j}} x_{i,j,k} = \sum_{\substack{j \in V \\ i \neq j}} x_{j,i,k} \quad \forall i \in V,\, k \in K \tag{3}$$

$$\sum_{i \in V} \sum_{\substack{j \in V \\ i \neq j}} x_{i,j,k} \leq (|N| + 1) \cdot \sum_{j \in N} x_{0,j,k} \quad \forall k \in K \tag{4}$$

$$\sum_{k \in K} q_{i,k} = d_i \cdot z_i \quad \forall i \in N \tag{5}$$

$$\sum_{i \in N} q_{i,k} \leq C \quad \forall k \in K \tag{6}$$

$$\sum_{\substack{j \in V \\ i \neq j}} x_{i,j,k} \leq q_{i,k} \leq C \cdot \sum_{\substack{j \in V \\ i \neq j}} x_{i,j,k} \quad \forall i \in N,\, k \in K \tag{7}$$

$$\sum_{i \in S} \sum_{\substack{j \in S \\ i \neq j}} x_{i,j,k} \leq |S| - 1 \quad \forall S \subseteq N\, (|S| \geq 2),\, k \in K \tag{8}$$

$$z_i \in \{0, 1\} \quad \forall i \in N \tag{9}$$

$$x_{i,j,k} \in \{0, 1\} \quad \forall i, j \in V\, (i \neq j),\, k \in K \tag{10}$$

$$q_{i,k} \in \mathbb{N}_0 \quad \forall i \in N,\, k \in K \tag{11}$$

The objective function (1) maximizes the difference between the prize gained from accepted requests and the transportation costs. The latter are determined by the total length of all tours. Constraints (2) impose that each vehicle k may leave the depot at most once and conservation of flow for each vehicle k at each vertex i is guaranteed by constraints (3). Constraints (4) are valid inequalities that express that at most $(|N|+1)$ arcs can be selected to form a feasible tour for each vehicle k if and only if vehicle k leaves the depot. In principle, these valid inequalities are not required for correctly formulating the problem; however, preliminary experiments have shown that they are quite effective in strengthening the linear relaxation of the formulation. For all requests i, constraints (5) ensure that, in case the request is accepted (i. e., $z_i = 1$), the cumulative delivered quantities $q_{i,k}$ must match the demand d_i. The capacity constraints (6) limit the total delivered quantity of each vehicle k to be at most C. Constraints (7) guarantee the consistency between the variables $q_{i,k}$ and $x_{i,j,k}$. For every vehicle k and every subset of requests $S \subseteq N$, the subtour elimination constraints are given by (8). More precisely, these constraints guarantee connectivity to the depot and they are based on the *Danzig-Fulkerson-Johnson* (DFJ) subtour elimination constraints (refer to [15]). Finally, (9)–(11) formalize the domains of the decision variables. As [5] note, the integrality requirement in (11) may be relaxed if all demands d_i and the vehicle capacity C are integer-valued, as optimal solutions would still exhibit integral values for $q_{i,k}$ in such cases.

The second MILP model for the SDCPTP is (12)–(26) (referred to as SDCPTP2 in what follows). In contrast to SDCPTP1, the three-indexed variables $x_{i,j,k}$ in this model represent undirected edges. Therefore, the SDCPTP2 is limited to symmetric problems. However, as we will see in Sect. 4, this gives the formulation a computational advantage in presence of symmetric problems, which are commonly studied in the literature.

$$\max \quad \sum_{i \in N} p_i \cdot z_i - \sum_{k \in K} \sum_{i \in V} \sum_{\substack{j \in V \\ i<j}} c_{i,j} \cdot x_{i,j,k} - \sum_{k \in K} \sum_{j \in N} 2 \cdot c_{0,j} \cdot y_{j,k} \tag{12}$$

$$x_{i,j,k} = x_{j,i,k} \quad \forall i, j \in V\,(i < j), k \in K \tag{13}$$

$$\sum_{i \in V} 2 \cdot y_{i,k} + \sum_{j \in N} x_{0,j,k} = 2 \quad \forall k \in K \tag{14}$$

$$2 \cdot y_{i,k} + \sum_{\substack{j \in V \\ i \neq j}} x_{i,j,k} = 2 \cdot h_{i,k} \quad \forall i \in N,\, k \in K \tag{15}$$

$$\sum_{i \in V} \sum_{\substack{j \in V \\ i \neq j}} x_{i,j,k} \leq 2 \cdot (|N| + 1) \cdot (1 - \sum_{i \in V} y_{i,k}) \quad \forall k \in K \tag{16}$$

$$\sum_{k \in K} h_{i,k} \leq z_i \cdot |K| \quad \forall i \in N \tag{17}$$

$$\sum_{k \in K} q_{i,k} = d_i \cdot z_i \quad \forall i \in N \tag{18}$$

$$\sum_{i \in N} q_{i,k} \leq C \quad \forall k \in K \tag{19}$$

$$h_{i,k} \leq q_{i,k} \leq h_{i,k} \cdot C \quad \forall i \in N,\, k \in K \tag{20}$$

$$\sum_{i \in S} \sum_{\substack{j \in S \\ i<j}} x_{i,j,k} \leq |S| - 1 \quad \forall S \subseteq N\,(|S| \geq 2),\, k \in K \tag{21}$$

$$z_i \in \{0, 1\} \quad \forall i \in N \tag{22}$$

$$y_{i,k} \in \{0, 1\} \quad \forall i \in V,\, k \in K \tag{23}$$

$$h_{i,k} \in \{0, 1\} \quad \forall i \in N,\, k \in K \tag{24}$$

$$x_{i,j,k} \in \{0, 1\} \quad \forall i, j \in V\,(i \neq j),\, k \in K \tag{25}$$

$$q_{i,k} \in \mathbb{N}_0 \quad \forall i \in N,\, k \in K \tag{26}$$

The objective function (12) is analogous to (1) with the difference that the variables $y_{i,k}$ are included. Constraints (13) enforce the symmetry of the decision variables $x_{i,j,k}$. The equations in (14) guarantee that every vehicle k leaves the depot exactly once to start a tour that serves two or more customers (exactly two $x_{0,j,k}$ have value 1), serves a single customer (exactly one $y_{i,k}, i \neq 0$ has value 1), or remains idle at the depot ($y_{0,k} = 1$). For each customer i that is visited by vehicle k, constraints (15) force $h_{i,k}$ to take value 1. Constraints (16) are valid inequalities that are structurally identical to those presented for SDCPTP1 in (4). They express that at most $(|N|+1)$ edges can be selected to form a feasible tour for each vehicle k if and only if that vehicle does neither serve a single customer nor remain idle at the depot (i. e., all $y_{i,k}$ have value 0). Constraints (17) force z_i to take value 1 if at least one vehicle visits request i and constraints (18) assure that the delivered quantities match the demand for each accepted request.

Consistency between the variables $q_{i,k}$ and $h_{i,k}$ is ensured by constraints (20). Analogously to the SDCPTP1 formulation, for every vehicle and every subset of requests, the subtour elimination constraints are given by (21). Finally, the domains of the variables are defined by (22)–(26). Note that the considerations that apply to (11) also apply to (26).

Another representation of subtour elimination constraints, which to the best of our knowledge was first presented in [10] for a prize-collecting TSP, can also be applied to the SDCPTP. Let us define $\delta(S \subseteq N) := \{(i,j) \mid i \in S, j \in V \setminus S\}$. Then, the third MILP formulation of the SDCPTP (referred to as SDCPTP3 in what follows) is given by (12)–(26) with the only difference that the subtour elimination constraints (21) are substituted for the following constraints (27):

$$\sum_{(i,j)\in\delta(S)} x_{i,j,k} \geq 2 \cdot h_{i,k} \quad \forall S \subseteq N,\, i \in S,\, k \in K \tag{27}$$

The subtour elimination constraints (27) guarantee that, for each request i in every subset of requests ($S \subseteq N$), iff i is served (i. e., $h_{i,k} = 1$) there must be at least two edges that connect vertices in S to the remaining vertices in $V \setminus S$, such that connectivity to the depot is enforced and no subtours occur. As in constraints (8) in SDCPTP1 and (21) in SDCPTP2, note that the number of these constraints grows exponentially with the number of requests. This implies that these constraints should be generated dynamically in the branch-and-cut framework (see Sect. 4).

Disallowing SD in these formulations is straightforward. In SDCPTP1, it becomes necessary to extend the model by constraints (28):

$$\sum_{k\in K} \sum_{\substack{j\in N \\ i\neq j}} x_{i,j,k} \leq z_i \quad \forall i \in N \tag{28}$$

And in SDCPTP2 or SDCPTP3 it suffices to replace constraints (17) with (29):

$$\sum_{k\in K} h_{i,k} \leq z_i \quad \forall i \in N \tag{29}$$

4 Computational Experiments and Numerical Results

In this section, we describe our computational experiments and discuss their numerical results. All experiments were performed on a shared compute cluster equipped with Intel Xeon Gold 6126 processors. For our experiments, we used 4 cores and 16 GB of RAM. We implemented the MILP models with Python 3.9 and used the branch-and-cut solver Gurobi Optimizer 10.0 [21] with a time limit of 3600 s using 4 threads. The subtour elimination constraints are implemented as lazy constraints, i. e., at every node of the branch-and-cut tree that yields an integer-feasible solution, we examine if there exist any subtours. If this is the case, we identify the smallest subset of vertices S that constitutes a subtour and apply the corresponding constraint (8), (21), or (27) for that subset.

The remainder of this section is structured as follows. We begin in Sect. 4.1 with a description of the problem instances. Afterwards, in Sect. 4.2, we discuss the numerical results.

4.1 Instance Generation

We base our experiments on two sets of instances. The first set was originally proposed for the CPTP in [6], where the authors adapted 10 instances from the well-known benchmark set for the vehicle routing problem in [12]. In these proposed CPTP instances, there are between 50 to 199 requests $|N|$ (each of which is associated with a demand and a prize) on a complete graph, a fleet size $|K| \in \{2, 3, 4\}$, and vehicle capacity $C \in \{50, 75, 100\}$. All of these instances were optimally solved for the CPTP in [6]; therefore, it is interesting to investigate the same instances from the perspective of the SDCPTP.

The second set of instances are motivated by the experiments in [5], where the SDCTOP has been studied. In that paper, the authors observed that SD typically only has a marginal impact if the customer demands are small w. r. t. the capacity (which applies to the first set of instances discussed above). To investigate if this observation also holds for the SDCPTP, we generate the second set based on the methodology in [9,17]. In particular, the demands are drawn uniformly at random from the interval $[\eta \cdot C, \nu \cdot C]$. Following [5], we consider the following eleven pairs of values for (η, ν): $(0.01, 0.1)$, $(0.1, 0.3)$, $(0.1, 0.5)$, $(0.1, 0.7)$, $(0.1, 0.9)$, $(0.3, 0.5)$, $(0.3, 0.7)$, $(0.3, 0.9)$, $(0.5, 0.7)$, $(0.5, 0.9)$, $(0.7, 0.9)$.

4.2 Experimental Results

As the SDCPTP contains the capacitated vehicle routing problem as a special case, the problem is NP-hard (refer to [27]). Therefore, finding optimal solutions to larger instances quickly becomes intractable. However, we can still solve small test instances optimally in a reasonable time, and compare the efficiency of SDCPTP1 to 3 (refer to Sect. 3.2).

Table 2 illustrates the numerical results on the first set of test instances. The first column displays the instance and the corresponding optimal CPTP solution z^* (taken from [6]) is shown in the second column. For SDCPTP1 to 3, the objective value z of the best feasible solution that was found is reported. If the solver proved z to be optimal, we report the runtime t in seconds; otherwise, we report the remaining *Mixed-Integer Programming* (MIP) gap g after 3600 s as a percentage value. For each row, the smallest runtime or MIP gap (in case none of the formulations returned the optimal solution) is displayed in bold. Additionally, the best SDCPTP solution is highlighted in bold (z') and this value is used to calculate the improvement w. r. t. z^*, i. e., the potential benefit of enabling split delivery. These percentages are calculated as $\frac{z' - z^*}{z^*}$, where z' is the best feasible solution found by SDCPTP1 to 3.

Table 2. Numerical results for the first set of instances (based on [6]).

Instance	CPTP	SDCPTP1		SDCPTP2		SDCPTP3		Imp.
$(\|N\|, C, \|K\|)$	z^*	z	t or g	z	t or g	z	t or g	(%)
6 (50,50,2)	33.88	**38.72**	45.5 s	**38.72**	**1.8** s	**38.72**	1.8 s	14.3
6 (50,75,2)	72.28	**72.28**	170.2 s	**72.28**	4.3 s	**72.28**	**2.7** s	0.0
6 (50,100,2)	100.27	**100.27**	961.2 s	**100.27**	9.0 s	**100.27**	**5.5** s	0.0
6 (50,50,3)	40.95	**48.62**	35.8%	**48.62**	23.4 s	**48.62**	**17.2** s	18.7
6 (50,75,3)	92.32	91.68	29.1%	**92.32**	106.0 s	**92.32**	**36.3** s	0.0
6 (50,100,3)	134.72	133.65	23.9%	**134.72**	239.1 s	**134.72**	**42.0** s	0.0
6 (50,50,4)	45.43	**50.50**	105.0%	**50.50**	2506 s	**50.50**	**1295** s	11.2
6 (50,75,4)	99.37	90.55	79.7%	**109.01**	2594 s	**109.01**	**379.6** s	9.7
6 (50,100,4)	153.30	136.05	60.9%	**154.59**	10.9%	**154.59**	**2216** s	0.8
7 (75,50,2)	49.18	**50.13**	82.5 s	**50.13**	4.2 s	**50.13**	**4.2** s	1.9
7 (75,75,2)	92.44	**95.90**	192.4 s	**95.90**	4.6 s	**95.90**	**4.0** s	3.7
7 (75,100,2)	132.70	**133.20**	2211 s	**133.20**	24.9 s	**133.20**	**15.0** s	0.4
7 (75,50,3)	69.94	73.23	21.5%	**73.67**	43.9 s	**73.67**	**41.4** s	5.3
7 (75,75,3)	131.12	**135.51**	15.1%	**135.51**	149.8 s	**135.51**	**106.1** s	3.4
7 (75,100,3)	185.25	179.44	25.8%	**187.37**	2328 s	**187.37**	**458.7** s	1.1
7 (75,50,4)	90.65	**96.16**	39.0%	**96.16**	594.8 s	**96.16**	**387.9** s	6.1
7 (75,75,4)	158.11	163.26	36.5%	164.70	10.6%	**167.14**	**4.0**%	5.7
7 (75,100,4)	233.40	213.37	42.3%	232.91	13.3%	**234.86**	**7.0**%	0.6
3 (100,50,2)	57.75	**57.75**	2237 s	**57.75**	**8.1** s	**57.75**	14.1 s	0.0
3 (100,75,2)	106.15	**109.01**	19.9%	**109.01**	61.3 s	**109.01**	**26.5** s	2.7
3 (100,100,2)	158.21	**168.15**	10.5%	162.20	161.6 s	162.20	**50.0** s	6.3
3 (100,50,3)	80.82	76.36	66.4%	**80.82**	461.1 s	**80.82**	**164.3** s	0.0
3 (100,75,3)	147.55	124.28	66.1%	**151.63**	10.9%	**151.63**	**1679** s	2.8
3 (100,100,3)	218.63	187.87	45.1%	**222.62**	6.3%	**222.62**	**1.0**%	1.8
3 (100,50,4)	100.36	74.72	130.8%	99.45	24.8%	**100.36**	**9.5**%	0.0
3 (100,75,4)	185.27	133.53	96.1%	**181.41**	22.8%	179.07	**20.8**%	−2.1
3 (100,100,4)	255.33	195.91	74.9%	254.92	20.8%	**268.33**	**11.9**%	5.1
8 (100,50,2)	57.75	**57.75**	2104 s	**57.75**	**7.9** s	**57.75**	13.9 s	0.0
8 (100,75,2)	106.15	108.88	20.3%	**109.01**	68.5 s	**109.01**	**29.5** s	2.7
8 (100,100,2)	158.21	**168.15**	10.6%	162.20	178.3 s	162.20	**54.7** s	6.3
8 (100,50,3)	80.82	76.36	66.5%	**80.82**	467.8 s	**80.82**	**167.2** s	0.0
8 (100,75,3)	147.55	124.28	66.1%	**151.63**	10.7%	**151.63**	**1596** s	2.8
8 (100,100,3)	218.63	175.59	55.3%	**222.62**	6.3%	**222.62**	**1.2**%	1.8
8 (100,50,4)	100.36	74.72	130.7%	99.45	24.5%	**100.36**	**9.1**%	0.0
8 (100,75,4)	185.27	133.53	96.2%	**181.41**	22.8%	179.07	**20.9**%	−2.1

(*continued*)

Table 2. (*continued*)

Instance	CPTP	SDCPTP1		SDCPTP2		SDCPTP3		Imp.
$(\|N\|, C, \|K\|)$	z^*	z	t or g	z	t or g	z	t or g	(%)
8 (100,100,4)	268.34	195.91	74.9%	254.92	20.8%	**268.33**	**11.9**%	−0.0
14 (100,50,2)	43.26	**43.26**	501.1 s	**43.26**	11.7 s	**43.26**	**5.7** s	0.0
14 (100,75,2)	77.09	75.76	33.7%	**77.09**	1798 s	**77.09**	**222.6** s	0.0
14 (100,100,2)	125.29	112.34	41.9%	**125.29**	3076 s	**125.29**	**203.5** s	0.0
14 (100,50,3)	59.43	**59.43**	41.3%	**59.43**	559.1 s	**59.43**	**105.1** s	0.0
14 (100,75,3)	112.56	66.36	160.8%	**112.81**	16.3%	**112.81**	**2531** s	0.2
14 (100,100,3)	182.31	159.44	59.5%	176.25	22.7%	**182.31**	**3.5**%	0.0
14 (100,50,4)	68.63	14.69	785.8%	**68.63**	33.2%	**68.63**	**8.4**%	0.0
14 (100,75,4)	139.88	44.75	423.2%	99.24	105.0%	**137.21**	**19.7**%	−1.9
14 (100,100,4)	237.68	50.73	558.1%	204.54	52.0%	**237.68**	**5.6**%	0.0
Avg. ($n \leq 100$)	124.32	106.05	81.1%	123.93	9.7%	126.27	3.0%	2.4
13 (120,50,2)	64.12	**69.77**	1773 s	**69.77**	**40.4** s	**69.77**	43.2 s	8.8
13 (120,75,2)	110.12	102.79	26.1%	**110.62**	444.9 s	**110.62**	**133.1** s	0.5
13 (120,100,2)	145.75	80.47	119.9%	**145.75**	7.4%	**145.75**	**3.6**%	0.0
13 (120,50,3)	87.25	38.57	217.4%	**91.04**	8.9%	**91.04**	**1492** s	4.3
13 (120,75,3)	139.37	**108.17**	79.5%	62.24	186.7%	97.09	**67.3**%	−22.4
13 (120,100,3)	181.63	99.22	161.3%	**141.00**	**74.5**%	100.70	125.2%	−22.4
13 (120,50,4)	104.18	45.13	270.7%	46.98	208.4%	**65.75**	**99.5**%	−36.9
13 (120,75,4)	161.62	47.12	438.5%	**109.52**	**116.3**%	52.77	306.3%	−32.2
13 (120,100,4)	200.62	75.12	343.6%	115.58	178.3%	**133.50**	**117.4**%	−33.5
9 (150,50,2)	65.03	**66.13**	1532 s	**66.13**	**29.9** s	**66.13**	30.2 s	1.7
9 (150,75,2)	117.66	110.58	16.6%	**117.66**	93.2 s	**117.66**	**57.6** s	0.0
9 (150,100,2)	161.23	147.70	24.8%	**161.23**	5.4%	**161.23**	**1058** s	0.0
9 (150,50,3)	96.16	70.77	75.8%	**98.41**	448.0 s	**98.41**	**178.3** s	2.3
9 (150,75,3)	160.96	101.57	99.9%	**161.20**	12.5%	**161.20**	**6.8**%	0.1
9 (150,100,3)	230.49	129.65	114.5%	195.89	34.3%	**228.33**	**9.8**%	−0.9
9 (150,50,4)	121.35	42.66	296.8%	**125.71**	12.2%	**125.71**	**6.7**%	3.6
9 (150,75,4)	204.25	80.04	241.1%	182.54	37.3%	**186.41**	**28.7**%	−8.7
9 (150,100,4)	290.54	128.23	188.3%	**260.24**	**35.0**%	203.52	65.9%	−10.4
15 (150,50,2)	64.98	**66.96**	6.5%	**66.96**	56.5 s	**66.96**	**33.3** s	3.0
15 (150,75,2)	120.93	115.54	20.0%	**120.93**	549.1 s	**120.93**	**183.0** s	0.0
15 (150,100,2)	169.71	162.49	21.6%	**171.05**	5.6%	**171.05**	**1858** s	0.8
15 (150,50,3)	96.42	82.41	63.2%	**98.63**	5.5%	**98.63**	**509.7** s	2.3
15 (150,75,3)	174.58	116.57	84.6%	162.94	21.8%	**171.05**	**8.3**%	−2.0
15 (150,100,3)	244.08	173.50	66.9%	228.69	21.3%	**247.78**	**6.6**%	1.5
15 (150,50,4)	124.02	91.73	96.6%	119.38	29.7%	**124.61**	**17.4**%	0.5
15 (150,75,4)	219.22	150.39	87.0%	**194.20**	36.6%	189.58	**35.1**%	−11.4

(*continued*)

Table 2. (*continued*)

Instance	CPTP	SDCPTP1		SDCPTP2		SDCPTP3		Imp.
$(\|N\|, C, \|K\|)$	z^*	z	t or g	z	t or g	z	t or g	(%)
15 (150,100,4)	308.07	206.45	83.0%	244.39	49.0%	**255.14**	**37.7**%	−17.2
10 (199,50,2)	70.87	**70.87**	21.5%	**70.87**	123.6 s	**70.87**	**111.3** s	0.0
10 (199,75,2)	124.85	119.45	26.7%	**124.85**	**432.6** s	**124.85**	442.0 s	0.0
10 (199,100,2)	171.24	157.89	31.7%	169.04	10.1%	**171.24**	**3.6**%	0.0
10 (199,50,3)	103.79	72.94	105.7%	**104.46**	10.8%	**104.46**	**3.2**%	0.6
10 (199,75,3)	177.90	103.24	128.3%	**167.06**	24.0%	161.20	**22.1**%	−6.1
10 (199,100,3)	250.18	183.66	72.7%	**238.36**	21.8%	233.41	**18.9**%	−4.7
10 (199,50,4)	134.81	40.95	397.7%	132.96	24.8%	**134.66**	**17.3**%	−0.1
10 (199,75,4)	229.27	114.98	172.4%	**225.81**	**24.9**%	199.74	33.3%	−1.5
10 (199,100,4)	324.02	169.11	147.2%	230.74	70.9%	**262.36**	**42.8**%	−19.0
16 (199,50,2)	66.81	**69.96**	13.9%	**69.96**	110.5 s	**69.96**	**74.8** s	4.7
16 (199,75,2)	123.38	120.11	21.4%	**125.37**	540.9 s	**125.37**	**371.1** s	1.6
16 (199,100,2)	177.23	154.01	34.1%	176.28	5.4%	**177.23**	**2243** s	0.0
16 (199,50,3)	99.70	88.12	64.9%	101.35	11.4%	**101.63**	**6.8**%	1.9
16 (199,75,3)	179.55	107.34	116.4%	156.69	32.3%	**176.16**	**11.0**%	−1.9
16 (199,100,3)	258.07	156.81	98.3%	245.20	18.1%	**248.03**	**14.1**%	−3.9
16 (199,50,4)	131.37	60.73	231.1%	**133.01**	21.3%	130.36	**18.9**%	1.2
16 (199,75,4)	235.03	133.40	130.2%	**207.68**	**34.0**%	195.38	38.3%	−11.6
16 (199,100,4)	336.24	136.90	201.1%	**285.73**	**34.9**%	273.65	37.4%	−15.0
Avg. ($n \leq 199$)	143.92	106.03	99.0%	135.68	20.7%	136.71	14.9%	−1.3

The results in Table 2 demonstrate that SDCPTP1 has the worst performance overall. In particular, SDCPTP2 or 3 either find optimal solutions faster or yield a lower MIP gap than SDCPTP1 on instances that remain unsolved after 3600 s. This shows that (in presence of these symmetric test instances) the symmetric formulations are much more efficient. Comparing SDCPTP2 to SDCPTP3, the former achieves better performance (by finding optimal solutions faster or returning lower MIP gaps after runtime) in only 12 out of 90 cases. Concerning the benefits of split delivery, for the instances with up to 100 requests, an average improvement of 2.4% is observed. However, when we also consider instances with more than 100 requests, we have negative improvements of −1.3% on average. This is explained by the fact that the MIP gaps after runtime are often large; and the feasible solutions found for the SDCPTP are still worse than the optimal CPTP solutions. Regarding the improvements, [5] already observed that the demands may be too small w. r. t. the vehicle capacity in these instances and thus, split delivery has no significant impact.

The numerical results of our experiments on the second set, where the ratio of demand to capacity is larger, are displayed in Table 3. As in the previous table, we compare the solutions of the SDCPTP after 3600 s of computation time with the corresponding optimal solution of the CPTP. For these instances, there are no existing optimal solutions, which are important to highlight the benefits of split delivery. Therefore, we resorted to the following method for generating reference values. We used SDCPTP3 and integrated constraints (29) to disable SD. We then solved each instance for 48 h on 24 cores (using 24 threads) with sufficient RAM in our Intel Xeon Gold 6126 compute cluster. This allowed us to find all optimal CPTP solutions except for a few cases that are explicitly highlighted in Table 3.

Table 3. Numerical results for the second set of instances (based on [5]).

Instance	CPTP	SDCPTP1		SDCPTP2		SDCPTP3		Imp.
$(\|N\|, C, \|K\|)$	z^*	z	t or g	z	t or g	z	t or g	(%)
6 ($\|N\| = 50, C = 160, \|K\| = 10$)								
$(0.01, 0.1)$	270.04	246.84	24.2%	**270.04**	2.8%	**270.04**	**2.2%**	0.0
$(0.1, 0.3)$	1143.09[1]	65.60	2049%	**1113.57**	**17.1%**	1055.67	23.9%	−3.7
$(0.1, 0.5)$	1363.44[1]	1315.11	28.9%	**1386.67**	**13.6%**	1380.10	13.8%	2.5
$(0.1, 0.7)$	1370.32[1]	763.62	125%	1416.16	12.4%	**1444.18**	**9.4%**	5.4
$(0.1, 0.9)$	1425.15	1456.32	31.7%	1531.04	14.3%	**1550.11**	**11.7%**	8.8
$(0.3, 0.5)$	1299.91	1261.59	43.6%	1455.84	16.6%	**1469.63**	**14.9%**	13.1
$(0.3, 0.7)$	1351.07	489.22	261%	**1377.94**	15.0%	1375.26	**13.4%**	2.0
$(0.3, 0.9)$	1583.81	1657.35	19.3%	1704.94	8.3%	**1717.53**	**6.1%**	8.4
$(0.5, 0.7)$	1107.72	1673.19	19.2%	**1698.54**	**8.4%**	1684.44	8.5%	52.1
$(0.5, 0.9)$	1369.87	1707.55	14.0%	**1714.62**	**5.2%**	1706.67	5.5%	25.2
$(0.7, 0.9)$	1432.22	1651.31	22.9%	1686.67	9.3%	**1712.68**	**7.1%**	19.6
7 ($\|N\| = 75, C = 140, \|K\| = 8$)								
$(0.01, 0.1)$	198.45	0.00	1302%	228.35	10.1%	**234.89**	**5.6%**	18.4
$(0.1, 0.3)$	879.92[2]	395.08	187%	**838.50**	**27.8%**	581.04	82.7%	−4.7
$(0.1, 0.5)$	1051.10[2]	908.20	43.5%	**1076.38**	13.7%	1074.45	**10.1%**	2.4
$(0.1, 0.7)$	1102.18[2]	871.46	50.7%	1144.17	8.8%	**1160.22**	**5.8%**	5.3
$(0.1, 0.9)$	1050.50	1069.98	25.3%	**1136.04**	7.9%	1132.08	**6.5%**	8.1
$(0.3, 0.5)$	1041.70	1064.12	26.9%	1152.12	9.5%	**1155.28**	**8.4%**	10.9
$(0.3, 0.7)$	1034.43	1005.61	35.1%	1080.50	17.3%	**1094.14**	**12.9%**	5.8
$(0.3, 0.9)$	1106.43	1207.11	14.3%	1241.53	4.0%	**1248.28**	**2.5%**	12.8
$(0.5, 0.7)$	761.38	1154.29	15.9%	1172.06	7.3%	**1178.23**	**4.7%**	54.7
$(0.5, 0.9)$	917.96	1151.68	20.3%	**1203.31**	4.7%	**1203.31**	**3.8%**	31.1
$(0.7, 0.9)$	1048.90	1225.84	12.5%	**1235.12**	2.5%	**1235.12**	**2.1%**	17.8
Avg.	1086.80	1015.50	198%	1221.10	11.1%	1211.97	11.9%	13.5

[1] These MIP gaps are up to 5.28% after 48 h of computation.
[2] The MIP gap is 12.2% after 48 h of computation.

In Table 3, even though the average MIP gaps for SDCPTP2 and 3 are more than 11% on average, we can observe that split delivery can drastically increase the objective value for such instances, where there is a high ratio of demand to vehicle capacity. For 2 of the 22 instances, improvements of more than 50% are possible through split delivery and the overall average is 13.5%. Within the 3600 s of computation time, we can see that with an increasing ratio of demand to vehicle capacity, the MIP gaps become smaller in relative terms for all three formulations. As was the case for the first set of test instances, it is once again evident that SDCPTP1 can not compete with SDCPTP2 or SDCPTP3. Regarding the latter formulations, SDCPTP3 provides better solutions than SDCPTP2 in 14 out of 22 cases (see Table 3).

Regarding the experiments as a whole and with respect to Tables 2 and 3, it can be concluded that in some cases with high MIP gaps, SDCPTP2 can perform slightly better than SDCPTP3 in terms of the feasible solution found after 3600 s of computation time. However, for small problem instances, with up to 75 requests and 10 vehicles, both formulations SDCPTP2 and SDCPTP3 typically lead to satisfactory solutions within 3600 s of computation time. Thus with our experimental results and as [5] already stated for the CTOP, the main advantages of split deliveries for the CPTP are also obtained in instances with high request demands, especially when demands exceed half of the vehicle capacity.

5 Conclusion and Future Research

In this paper, we investigated the SDCPTP that extends the profitable tour problem by a fleet of vehicles and the possibility of split deliveries. We began with a brief overview of routing problems with profits and split delivery in vehicle routing in Sect. 2. In particular, that section demonstrated that there is a scientific gap in studying split delivery in the context of profitable tour problems. To address this gap, we proposed three novel MILP formulations for the SDCPTP in Sect. 3. All of these formulations are three-index vehicle flow formulations that differ primarily in their treatment of the subtour elimination constraints within the branch-and-cut algorithm. Afterwards, in Sect. 4, we performed a computational study that consisted of solving the three different MILP formulations with a state-of-the-art solver on different instances from the literature. Our numerical results showed that there is a significant difference in performance when using these formulations. Whereas the performance of SDCPTP1 was comparatively poor, the second and third formulation showed promising results. While SDCPTP3 performed the best overall in our experiments, there are some cases where the SDCPTP2 can perform better. These cases concern especially those with a high ratio of request demand to vehicle capacity. On such instances, it became clear that split delivery can significantly improve the solution.

Based on this paper, there are several future research directions. While SDCPTP2 and SDCPTP3 showed promising performance, attempting to solve the SDCPTP optimally with an off-the-shelf solver quickly becomes intractable on larger instances. Therefore, apart from exact branch-and-price procedures

for the SDCPTP, heuristic methods should be investigated for the problem. As mentioned above, it would also be worthwhile to investigate cases where demand can exceed vehicle capacity, making split delivery indispensable. Moreover, in the context of practical applications, where requests might not be deterministic, it would be worthwhile to consider stochastic variants of the SDCPTP.

References

1. Angelelli, E., Mansini, R., Rizzi, R.: The probabilistic profitable tour problem under a specific graph structure. arXiv preprint arXiv:2204.07378 (2022)
2. Archetti, C., Bianchessi, N., Speranza, M.G.: The capacitated team orienteering problem with incomplete service. Optim. Lett. **7**(7), 1405–1417 (2013)
3. Archetti, C., Bianchessi, N., Speranza, M.G., Hertz, A.: Incomplete service and split deliveries in a routing problem with profits. Networks **63**(2), 135–145 (2014)
4. Archetti, C., Bianchessi, N., Speranza, M.G.: Optimal solutions for routing problems with profits. Discret. Appl. Math. **161**(4–5), 547–557 (2013)
5. Archetti, C., Bianchessi, N., Speranza, M.G., Hertz, A.: The split delivery capacitated team orienteering problem. Networks **63**(1), 16–33 (2014)
6. Archetti, C., Feillet, D., Hertz, A., Speranza, M.G.: The capacitated team orienteering and profitable tour problems. J. Oper. Res. Soc. **60**(6), 831–842 (2009)
7. Archetti, C., Speranza, M.G.: Vehicle routing in the 1-skip collection problem. J. Oper. Res. Soc. **55**(7), 717–727 (2004)
8. Balas, E.: The prize collecting traveling salesman problem. Networks **19**(6), 621–636 (1989)
9. Belenguer, J.M., Martinez, M., Mota, E.: A lower bound for the split delivery vehicle routing problem. Oper. Res. **48**(5), 801–810 (2000)
10. Bienstock, D., Goemans, M.X., Simchi-Levi, D., Williamson, D.: A note on the prize collecting traveling salesman problem. Math. Program. **59**(1), 413–420 (1993)
11. Bortfeldt, A., Yi, J.: The split delivery vehicle routing problem with three-dimensional loading constraints. Eur. J. Oper. Res. **282**(2), 545–558 (2020)
12. Christofides, N.: The vehicle routing problem. In: Combinatorial Optimization (1979)
13. Cortés-Murcia, D.L., Afsar, H.M., Prodhon, C.: A branch and price algorithm for the electric capacitated profitable tour problem with mandatory stops. IFAC-PapersOnLine **52**(13), 1572–1577 (2019)
14. Cortés-Murcia, D.L., Afsar, H.M., Prodhon, C.: Multi-period profitable tour problem with electric vehicles and mandatory stops. Int. J. Sustain. Transp. **17**, 1–17 (2022)
15. Dantzig, G., Fulkerson, R., Johnson, S.: Solution of a large-scale traveling-salesman problem. J. Oper. Res. Soc. Am. **2**(4), 393–410 (1954)
16. Dell'Amico, M., Maffioli, F., Värbrand, P.: On prize-collecting tours and the asymmetric travelling salesman problem. Int. Trans. Oper. Res. **2**(3), 297–308 (1995)
17. Dror, M., Trudeau, P.: Savings by split delivery routing. Transp. Sci. **23**(2), 141–145 (1989)
18. Feillet, D., Dejax, P., Gendreau, M.: Traveling salesman problems with profits. Transp. Sci. **39**(2), 188–205 (2005)
19. Frizzell, P.W., Giffin, J.W.: The split delivery vehicle scheduling problem with time windows and grid network distances. Comput. Oper. Res. **22**(6), 655–667 (1995)

20. Gulczynski, D., Golden, B., Wasil, E.: The split delivery vehicle routing problem with minimum delivery amounts. Transp. Res. Part E Logist. Transp. Rev. **46**(5), 612–626 (2010)
21. Gurobi Optimization, LLC.: Gurobi Optimizer Reference Manual (2023)
22. Jepsen, M.K., Petersen, B., Spoorendonk, S., Pisinger, D.: A branch-and-cut algorithm for the capacitated profitable tour problem. Discret. Optim. **14**, 78–96 (2014)
23. Lera-Romero, G., Miranda-Bront, J.J.: A branch and cut algorithm for the time-dependent profitable tour problem with resource constraints. Eur. J. Oper. Res. **289**(3), 879–896 (2021)
24. Mullaseril, P.A., Dror, M., Leung, J.: Split-delivery routeing heuristics in livestock feed distribution. J. Oper. Res. Soc. **48**(2), 107–116 (1997)
25. Sierksma, G., Tijssen, G.A.: Routing helicopters for crew exchanges on off-shore locations. Ann. Oper. Res. **76**, 261–286 (1998)
26. Sun, P., Veelenturf, L.P., Dabia, S., Van Woensel, T.: The time-dependent capacitated profitable tour problem with time windows and precedence constraints. Eur. J. Oper. Res. **264**(3), 1058–1073 (2018)
27. Toth, P., Vigo, D. (eds.): Vehicle Routing: Problems, Methods, and Applications. Society for Industrial and Applied Mathematics (SIAM) (2014)
28. Tsiligirides, T.: Heuristic methods applied to orienteering. J. Oper. Res. Soc. **35**(9), 797–809 (1984)
29. Wang, X., Golden, B., Gulczynski, D.: A worst-case analysis for the split delivery capacitated team orienteering problem with minimum delivery amounts. Optim. Lett. **8**(8), 2349–2356 (2014)
30. Yu, M., Jin, X., Zhang, Z., Qin, H., Lai, Q.: The split-delivery mixed capacitated arc-routing problem: applications and a forest-based tabu search approach. Transp. Res. Part E Logist. Transp. Rev. **132**, 141–162 (2019)