

Caspar, Marvin

Article — Published Version

A matheuristic for the split delivery capacitated profitable tour problem with incomplete service

Central European Journal of Operations Research

Provided in Cooperation with:

Springer Nature

Suggested Citation: Caspar, Marvin (2025) : A matheuristic for the split delivery capacitated profitable tour problem with incomplete service, Central European Journal of Operations Research, ISSN 1613-9178, Springer, Berlin, Heidelberg, Vol. 34, Iss. 1, pp. 211-238, <https://doi.org/10.1007/s10100-025-01002-w>

This Version is available at:

<https://hdl.handle.net/10419/334876>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by/4.0/>



A matheuristic for the split delivery capacitated profitable tour problem with incomplete service

Marvin Caspar¹ 

Received: 18 July 2024 / Accepted: 26 September 2025 / Published online: 18 October 2025
© The Author(s) 2025

Abstract

This paper investigates the *Split Delivery Capacitated Profitable Tour Problem with Incomplete Service* (SDCPTP-IS), an extension of the *Capacitated Profitable Tour Problem* (CPTP) that introduces two key relaxations: *split delivery* (SD), allowing a request to be fulfilled by multiple vehicles, and *incomplete service* (IS), permitting partial fulfillment of request demands. The objective of the CPTP is to select a subset of requests and determine feasible tours for a fleet of homogeneous vehicles, maximizing the net collected prize (revenue minus delivery cost). We formulate the SDCPTP-IS as a *Mixed-Integer Linear Program* (MILP). A numerical study using benchmark instances from the literature to investigate the effect of split delivery or/and incomplete service in the CPTP with a state-of-the-art solver and a *Matheuristic* is conducted. This matheuristic exploits the problem structure of the *Capacitated Vehicle Routing Problem* (CVRP) to solve the SDCPTP-IS as a MILP. More precisely, we use a *Guided Local Search* (GLS) heuristic to obtain a CVRP solution that is embedded in the matheuristic to solve the SDCPTP-IS. For managerial insights, we conduct a sensitivity analysis on the CPTP, CPTP with split delivery (SDCPTP), CPTP with incomplete service (CPTP-IS), and the SDCPTP-IS, which investigates a varying number of requests, vehicles, and vehicle capacity.

Keywords Mixed-integer linear programming · Capacitated profitable tour problem · Split delivery · Incomplete service · Matheuristic · Guided local search

✉ Marvin Caspar
marvin.caspar@wiwi.uni-kl.de
<https://wiwi.rptu.de/fgs/bisor>

¹ Business Information Systems & Operations Research (BISOR), University of Kaiserslautern-Landau, Postfach 3049, 67653 Kaiserslautern, Germany

1 Introduction

Due to the increasing diversity of commodities, shorter life cycles, rising energy prices, and competitive environments, request acceptance decisions and transportation planning, can be important decision-making processes for companies (see Noroozi et al. (2018)). For example, in a make-to-order system with next-day delivery, accepting every incoming request may not be profitable. Instead, particularly if non-substitutable resources are scarce (e. g., personnel or materials), a decision maker must trade off the rewards of accepting individual requests against the available resource capacities and the compound effects on the costs (e. g., delivery costs), incurred by fulfilling all requests jointly. From an Operations Research perspective, this trade-off is treated in the *Traveling Salesman Problem with Profits* (TSPPs) (see, e. g., Balas (1989); Dell'Amico et al. (1995); Tsiligrirides (1984)) or, in the case of multiple vehicles, *Vehicle Routing Problem with Profits* (VRPPs) (refer to Toth and Vigo (2014)). The common feature of such problems is the possibility of accepting or rejecting requests; however, their application domains can differ. VRPPs occur in many practical scenarios, e. g., tourist trips, home fuel delivery, mobile crowdsourcing, and freight transport (see, e. g., Feillet et al. (2005); Gunawan et al. (2016)). Especially in transportation areas such as liner shipping (e.g., Karsten et al. (2015)), railway (e.g., Azadi Moghaddam Arani et al. (2019)), or internet and satellite networks (e.g., Wang et al. (2019); Zhang et al. (2018)), the task is to bundle accepted requests (also called orders) in a way that maximizes profit. In VRPPs, a standard assumption is that each demand must be completely satisfied. In the literature, the assumption that demand is only partly satisfied is known as *incomplete service* (IS) while satisfying demand by more than one vehicle is referred to *split delivery* (SD). However, as addressed in Archetti et al. (2014b) (and as we will see in more detail in Sect. 2.2), there is a research gap in the investigation of split delivery and incomplete service in the context of VRPPs. These extensions introduce additional computational effort, raising the question of under what conditions the trade-off between computational effort and operational improvements justifies their implementation.

To address this research gap, we investigate the *Split Delivery Capacitated Profitable Tour Problem with Incomplete Service* (SDCPTP-IS), a variant of VRPPs (see Toth and Vigo (2014)). In the SDCPTP-IS, a fleet of homogeneous vehicles with limited capacity is available to fulfill a set of potential request demands from different locations. Each request can only be delivered once and is associated with a prize (i. e., reward) that results proportionally from the amount of satisfied demand. The objective is to select a subset of requests and determine a feasible tour for each vehicle (that visits some of the selected requests exactly once and then returns to the depot) so that the prize collected by partially or completely fulfilling these requests serviced by one or more vehicles net of the delivery costs is maximized. A key feature of the SDCPTP-IS is the allowance for split deliveries, meaning an accepted request can be fulfilled by multiple vehicles through different vehicles. Additionally, due to incomplete service, accepted requests may not be completely served but only satisfied proportionally to the extent it is profitable. Practical applications of the SDCPTP-IS are observed in last-mile delivery (see Boysen et al. (2021)), particularly in scenarios, where fulfilling an entire request demand in a single tour is infeasible

due to vehicle constraints. This occurs when items are excessively bulky or heavy, necessitating split tours with incomplete service. Another relevant application is in waste collection operations, where different waste types generated at a single location (e.g., recyclables and organics) require separate transportation on split tours due to sorting requirements or capacity limitations.

In this paper, we propose the SDCPTP-IS where the demand of requests can be partially or fully satisfied. Following a literature review, we make the following additional contributions. We propose a *Mixed-Integer Linear Program* (MILP) formulation for the SDCPTP-IS, SDCPTP, and the CPTP-IS. We also present a matheuristic that exploits the problem structure of the *Capacitated Vehicle Routing Problem* (CVRP) to solve these three problems as a MILP. More precisely we use a *Guided Local Search* (GLS) heuristic to obtain a CVRP solution that is embedded in the matheuristic to solve the SDCPTP-IS, SDCPTP, and CPTP-IS afterward. In particular, we provide a detailed numerical study demonstrating the effectiveness of such a matheuristic compared to a state-of-the-art solver in tackling such models. Moreover, we perform computational experiments on these different problems with several instances from the literature for managerial insights.

The remainder of this paper is structured as follows. In Sect. 2, we begin with a brief overview of the related literature concerning the PTP. In particular, we examine different PTP model extensions and show different solution methods for them. Moreover, we discuss the split delivery and incomplete service concept in the context of routing problems. Afterward, in Sect. 3, we describe the SDCPTP-IS, SDCPTP, and CPTP-IS in detail and present a MILP model formulation. Within Sect. 4 first the underlying GLS heuristic for solving the CVRP is explained and then the matheuristic for the specific problems is discussed. Section 5 is dedicated to the computational experiments and their numerical results. Finally, in Sect. 6, we conclude and discuss implications for future research.

2 Related literature

In this section, we briefly discuss profitable tour problems from the literature and their solution methods. With Sect. 2.2, we sketch relevant contributions to routing problems with split delivery or incomplete service.

2.1 Profitable tour problems

The TSPPs is a family of problems related to the TSP (and the VRPPs) in which each request is associated with a reward that is commonly referred to as a *prize* (or, conversely, a penalty if left unvisited) whereas the arcs, that connect accepted requests in a feasible tour, are associated with a cost (refer to Feillet et al. (2005)). The essential characteristic of TSPPs is that there is a trade-off between the cumulative prize and the cost by traversing arcs. In general, there are three main variants of TSPPs:

- In the *Prize-Collecting TSP* (see Balas (1989)) there is a lower bound concerning the cumulative prize that must be collected. Under this constraint, the objective

becomes to minimize the total travel costs (and penalties for uncollected prizes).

- In contrast, in the *Orienteering Problem* (OP) (refer to Tsiligirides (1984)), we have an upper bound for the tour costs (e. g., the length of the tour), and the objective becomes to maximize the cumulative prize.
- In the *Profitable Tour Problem* (PTP), there are no bounds concerning the cumulative prize or the tour cost. Here the objective is to maximize the profit, i.e., the cumulative prize net of the costs (see Dell’Amico et al. (1995)).

Based on these three main categories, several variants and extensions of TSPPs and VRPPs exist (see also Toth and Vigo (2014)). Especially, the PTP finds numerous extensions in the literature as the following listing shows. The authors in Feillet et al. (2005) treat a profitable arc tour problem, where profits are assigned to arcs instead of vertices. The authors in Dasari et al. (2021) treat specifically the single vehicle case of the PTP. In recent years, stochastic variants of the PTP have also become common. For example, in Bruni et al. (2019), the authors analyze a PTP with stochastic costs. Similarly, Zhang et al. (2017) treats a stochastic PTP, where each request asks for service only with a given probability.

In the context of this paper, the *Capacitated Profitable Tour Problem* (CPTP) (refer to Archetti et al. (2009)) is of particular interest. The CPTP is a variant of the PTP in which each request is associated with a demand quantity, and the vehicle’s capacity is limited. The authors in Chbichib et al. (2012) consider a CPTP with multiple trips, where each vehicle is allowed to perform several routes under a strict workday duration limit. Paper (Lahyani et al. 2017) introduces a rich CPTP that includes several temporal and physical constraints, where requests need not be fully satisfied. Specifically, the vehicle is assumed to have multiple compartments with different capacities, and incompatible products must be kept separate during transportation. The authors in Bulhões et al. (2018) extend the CPTP by incorporating service level constraints. In Chentli et al. (2018), the CPTP with simultaneous pickup and delivery services is introduced. Study (Sun et al. 2018) introduces a variant of the CPTP that includes time windows and precedence constraints. Paper (Cortés-Murcia et al. 2019) considers a variant of the CPTP where restaurants are visited during lunchtime by a driver to charge electric vehicles. The work in Lera-Romero and Miranda-Bront (2021) extends the time-dependent CPTP from Sun et al. (2018) by several resource constraints. Paper (Cortés-Murcia et al. 2023) extends the model from Cortés-Murcia et al. (2019), by considering multi-periods and making agreements for recharge services. The authors in Caspar et al. (2023) provide three different MILP model formulations for the CPTP with split delivery and conclude through intensive experiments that Bienstock subtour elimination constraints (see Bienstock et al. (1993)) lead to the best performance within the CPTP.

2.2 Split delivery and incomplete service in routing problems

Split delivery leads to more freedom in fulfilling demands and becomes necessary when the demand of a request exceeds the vehicle capacity (Toth and Vigo 2014). The *Split Delivery Vehicle Routing Problem* (SDVRP) was first introduced by Dror and Trudeau (1989) and has been extensively analyzed over the last three decades.

Particularly, several worst-case analyses and empirical studies have demonstrated that when split delivery is permitted, significant savings can accrue (Toth and Vigo 2014). As well, several exact and heuristic algorithms have been presented to efficiently solve the SDVRP and its variants (refer to Toth and Vigo (2014) and references therein). Examples of such variants include the inclusion of time windows (Frizzell and Giffin 1995), minimum delivery amounts (Gulczynski et al. 2010), or three-dimensional loading constraints (Bortfeldt and Yi 2020).

Incomplete service can find practical application in situations where the service can be shared between different carriers and also becomes necessary when the demand of a request exceeds the vehicle capacity (Toth and Vigo 2014). The possibility to partially serve a request to increase the collected profits thanks to a better exploration of the vehicle capacity (Archetti et al. 2013a) makes incomplete service an important concept.

To the best of our knowledge, Archetti et al. (2013a, 2014a, 2014b); Wang et al. (2014) are the only contributions that consider split delivery or incomplete service in routing problems with profits. Within (Archetti et al. 2014a) 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 *Split Delivery Capacitated Team Orienteering Problem* (SDCTOP) and its extension with incomplete service, called SDCTOP-IS. A detailed study in Archetti et al. (2013a) and Archetti et al. (2014a) revealed that the combination of IS and SD does not significantly increase profits in the *Capacitated Team Orienteering Problem* (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 Archetti et al. (2014b), 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 Wang et al. (2014), 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.

These works demonstrate that split delivery and incomplete service can increase profits in routing problems with profits. However, to the best of our knowledge and as stated in Archetti et al. (2014b), no literature on the CPTP investigates SD (besides (Caspar et al. 2023)) or IS. This provides further motivation for the SDCPTP-IS presented in the following section.

2.3 Solution methods for routing problems with profits

The authors in Toth and Vigo (2014) provide a comprehensive overview of solution methods including exact, heuristic, and approximation algorithms for the CPTP and the closely related OP published up to the year 2013. For the standard CPTP, exact algorithms based on branch-and-price (see Archetti et al. (2009), Archetti et al. (2013b)) and branch-and-cut (see Jepsen et al. (2014)) have emerged as suitable solution methods. However, CPTP extensions often introduce significant additional com-

plexity, and such exact algorithms are only capable of solving small instances of such problems. Consequently, metaheuristic approaches have gained prominence when it comes to addressing large instances of such problems effectively. Genetic algorithms (see Zhang et al. (2017) and Bruni et al. (2019)) and tabu search (see Archetti et al. (2009) and Bruni et al. (2019)) have shown promising results for various CPTP variants. Additionally, variable neighborhood search algorithms have proven effective for solving the CPTP (see Archetti et al. (2009) and Dasari et al. (2021)) and the closely related OP (see Sevkli and Sevilgen (2006), Labadie et al. (2012), Pěnička et al. (2019)). Furthermore, local search heuristics have been successfully applied to OPs and similar problems (see Vansteenwegen et al. (2009), Labadie et al. (2011) and Gunawan et al. (2018)).

However, such metaheuristics do not guarantee finding the optimal solution and may sometimes suffer from poor performance. In contrast, matheuristics leverage the global exploration capabilities of metaheuristics while integrating the precision of mathematical programming to outperform metaheuristics (Fischetti and Fischetti 2018). Several studies have successfully applied matheuristics to various CPTP and OP variants, demonstrating their superiority over other solution methods (see Archetti et al. (2015), Lahyani et al. (2017), and Yu et al. (2019)). In this paper, we design a matheuristic that incorporates a guided local search heuristic to generate a CVRP solution, which is then integrated into a matheuristic framework to solve the SDCPTP-IS.

3 Problem definition and mathematical programming formulations

In this section, we begin with a concise problem definition of the SDCPTP-IS in Sect. 3.1 and derive some interesting properties for the problem. Afterward, in Sect. 3.3, we present a mathematical programming formulation for the problem.

3.1 Problem definition

Let $G(V, A)$ be a complete graph, where $V = \{0, 1, \dots, |N|\}$ is the set of vertices and $A = \{(i, j) \mid i, j \in V (i \neq j)\}$ is the set of arcs. The vertex $0 \in V$ represents the depot and $N = \{1, \dots, |N|\} = V \setminus \{0\}$ 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 positive travel cost $c_{i,j}$ is associated with each arc $(i, j) \in A$ and we assume that the triangle inequality is valid for these values. We say that the problem is symmetric only if $c_{i,j} = c_{j,i}$ for all $(i, j) \in A$. In the SDCPTP-IS, we are given a fleet K of homogeneous vehicles, each with a maximum capacity C . The tour of each vehicle $k \in K$ starts and ends at the depot and, unlike in the CPTP, each request $i \in N$ may be served by more than one vehicle (split delivery) and the demand d_i of an accepted request $i \in N$ may not be completely served (incomplete service). The objective is to find feasible tours for the fleet so that the total profit Z (i. e., the cumulative prize, net of the cost of delivery) is maximized. Fig. 1 illustrates the problem.

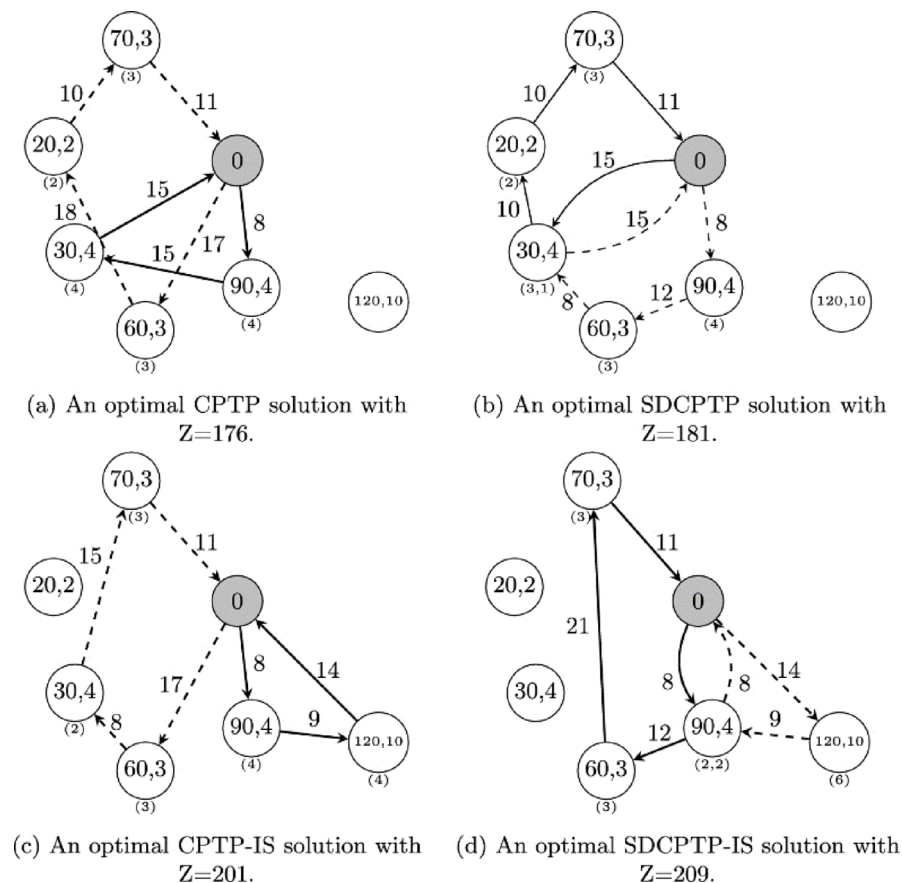


Fig. 1 An illustration of optimal CPTP (a), SDCPTP (b), CPTP-IS (c) and SDCPTP-IS (d) solutions in a problem with $|N| = 6$ requests and a fleet of two vehicles, each with a capacity of $C = 8$. The depot is labeled with 0 and the remaining vertices are labeled with a tuple (p_i, d_i) . The tours are indicated by the solid and dashed arcs, respectively, and the costs $c_{i,j}$ are shown next to each arc. The delivered quantity $q_{i,k}$ to request i by vehicle k is specified in round brackets under the node

Figure 1 shows how the objective function value can be increased from the optimal CPTP solution $Z = 176$ (a) with split delivery to $Z = 181$ (b) and with incomplete service to $Z = 201$ (c). The combination of SD and IS (d) leads to the best objective function value with $Z = 209$.

3.2 Properties for the SDCPTP-IS

The authors of Archetti et al. (2014a) and Archetti et al. (2014b) explored different properties of the SDCTOP and SDCTOP-IS. These properties concern the number of requests that obtain SD and IS in an optimal solution and the amount of additional profit that can be collected if SD or/and IS are possible. This section demonstrates that a subset of these properties is also valid for the SDCPTP-IS. To establish such properties, we make two definitions.

Definition 1 Given $|K|$ requests $i_1, i_2, \dots, i_{|K|}$ and $|K|$ tours. Tour 1 visits requests i_1 and i_2 , tour 2 visits requests i_2 and i_3 , ..., and tour $|K| - 1$ visits requests $i_{|K|-1}$ and $i_{|K|}$, and tour $|K|$ visits requests $i_{|K|}$ and $i_{|K|+1} = i_1$. The subset of requests $i_1, i_2, \dots, i_{|K|}$ is called a *k-split cycle*.

Definition 2 Given a complete graph $G(V, A)$, the non-negative cost matrix describes the travel cost between any two vertices $(i, j) \in A$ and satisfies the triangle inequality if for any three vertices i, j and l in G :

$$c_{i,j} + c_{j,l} \geq c_{i,l} \quad (1)$$

The authors in Dror and Trudeau (1989) proved that there is an optimal solution to the SDVRP without such k-split cycles. Now we show that this property is also valid for the SDCPTP-IS.

Theorem 1 *Given an SDCPTP-IS instance that satisfies the triangle inequality, there exists an optimal solution for the SDCPTP-IS without any k-split cycle (for any k).*

Proof Suppose there exists an optimal SDCPTP-IS solution that contains k-split cycles. Consider such a k-split cycle and let q_{min} be the minimum quantity delivered to any of the requests of the k-split cycle by any tour. Denote by i_{min} the request that receives quantity q_{min} in this tour r_{min} . Now remove this request i_{min} from tour r_{min} and reconnect the preceding and subsequent request of i_{min} from this tour r_{min} to obtain the new tour r_{new} . Due to the cost triangle inequality, r_{new} has lower transportation costs and additional residual capacity q_{min} . This residual capacity can serve at least q_{min} units of the demand of request i_{pre} previously served by tour r_{pre} of the k-split cycle. Hence, q_{min} units of request i_{min} are transformed from tour r_{pre} to tour r_{new} . As a result, tour r_{pre} has an additional residual capacity of q_{min} units that can be used to deliver request i_{min} . This transformation leads to a new solution with lower transportation costs while maintaining the same collected prize, contradicting the assumption that the original solution with a k-split cycle was optimal. Thus, no optimal solution can contain a k-split cycle, completing the proof by contradiction. $\square$

The following Fig. 2 illustrates an example of this Theorem 1 with $q_a, q_b, q_c \geq q_{min} > 0$.

An immediate consequence of Theorem 1 follows.

Corollary 1 *Given an SDCPTP-IS instance that satisfies the triangle inequality, there exists an optimal solution for the SDCPTP-IS where no two tours have more than one request with a split delivery in common.*

Theorem 2 *Given an SDCPTP-IS instance that satisfies the triangle inequality, there exists an optimal solution for the SDCPTP-IS where each tour has at most one request with incomplete service.*

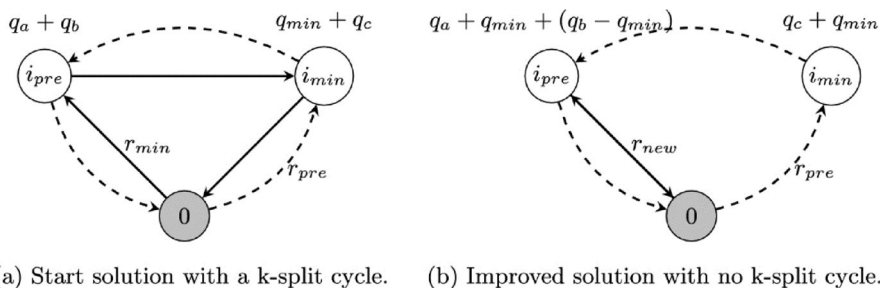


Fig. 2 An illustration of Theorem 1. The tour r_{new} in (b) substitutes tour r_{min} in (a) to obtain a better SDCPTP-IS solution without a k-split cycle

Proof Assume there is an optimal SDCPTP-IS solution, with a tour where more than one request receives incomplete service. Without loss of generality two of these requests i and j fulfill $\frac{p_i}{d_i} \geq \frac{p_j}{d_j}$. Now we move as much as possible demand from the least profitable request j to the more profitable request i . If the served demand of request j was greater than or equal to the non-served demand of request i , the demand of i will be fully satisfied, and thus we reduce the requests that obtain incomplete service by one. This new solution is better since the collected prize stays the same and transportation costs are reduced. Otherwise, if the demand served of request j was smaller than the not served demand of i , the demand of request i remains partially served, but again, we reduce the number of requests with incomplete service by one and obtain an improved solution. The contradiction is made. $\square$

Theorem 3 Given an SDCPTP-IS instance that satisfies the triangle inequality, there exists an optimal solution for the SDCPTP-IS where the total number of splits is less than the number of tours.

Proof Analog to the proof of Theorem 2 in Archetti et al. (2014b) for the SDCTOP. $\square$

Theorem 4 Given an SDCPTP-IS instance that satisfies the triangle inequality, there exists an optimal solution for the SDCPTP-IS where the total number of splits and incomplete services is not greater than the number of tours.

Proof Analog to the proof of Theorem 4 in Archetti et al. (2014a) for the SDCTOP-IS. $\square$

In the following, we want to analyze how much additional profit is possible when we allow or disallow split delivery or/and incomplete service in the CPTP. We can prove the following theorems.

Theorem 5 If the demand of each request $i \in N$ satisfies $d_i \leq C$ and the cost matrix satisfies the triangle inequality, any solution:

$$\frac{Z(\text{SDCPTP})}{Z(\text{SDCPTP} - \text{IS})} \geq \frac{1}{2} \quad (2)$$

Proof Given an optimal SDCPTP-IS solution, where according to Theorem 2, in each tour at most one request obtains IS. We take a tour (called the origin tour) of this solution and split it into the first tour, which only includes all requests with complete service and a second tour, including only the partially served requests. Both tours have lower or equal transportation costs than the origin tour due to the cost triangle inequality and both tours can completely satisfy their requests. If the cumulative prizes of this first tour are greater than the cumulative prizes of the second tour we substitute the origin tour with the first tour and else, the second tour. This substitution leads to an objective value of greater or equal to half of the objective value as before, since the sum of the cumulative prizes of the first and second tours equals the cumulative prizes of the origin tour. We repeat this procedure for all tours and gain a feasible SDCPTP solution that fulfills inequality (2).

Given a vehicle with capacity C and two requests i and j that both demand quantity $\frac{C}{2} + 1$ and have profit $\frac{C}{2} + 1$, while their travel costs satisfy $c_{0,i} = c_{0,j} < \epsilon$ (and thus $c_{i,j} < 2\epsilon$) for sufficiently small $\epsilon > 0$. In the SDCPTP, we can only satisfy one request and in the SDCPTP-IS, we can deliver both requests with quantity $\frac{C}{2}$. When capacity C tends to infinity the transport costs become negligible for the objective function value and the ratio $\frac{Z(\text{SDCPTP})}{Z(\text{SDCPTP-IS})}$ tends to $\frac{1}{2}$, making this bound tight (there exists no better bound). $\square$

The following Fig. 3 illustrates an example of this procedure, where both SDCPTP solutions in (b) completely satisfy their accepted requests (no incomplete service) and one of them fulfills inequality (2).

Theorem 6 *If each request $i \in N$ demand satisfies $d_i \leq C$ and the cost matrix satisfies the triangle inequality, any solution:*

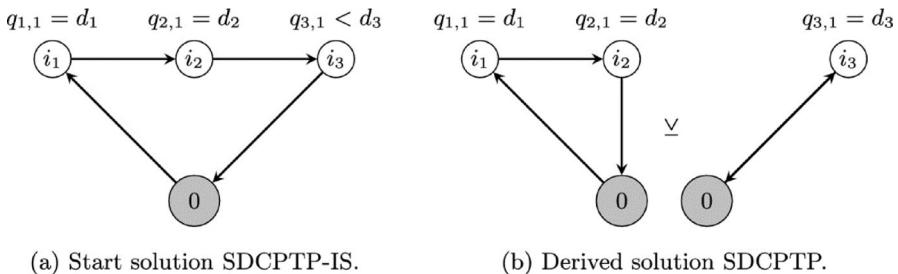


Fig. 3 An illustration of Theorem 5. Given the SDCPTP-IS solution in (a) where a vehicle completely serves requests i_1 and i_2 and incompletely serves i_3 . Without allowing incomplete services, we obtain two possible mutually exclusive solutions for the SDCPTP in (b). However, at least one of them satisfies the inequality (2)

$$\frac{Z(CPTP - IS)}{Z(SDCPTP - IS)} \geq \frac{1}{2} \quad (3)$$

and this objective bound is tight.

Proof Given an optimal SDCPTP-IS solution, Corollary 1, states that no two tours share more than one request with a split delivery. For any given tour (called the origin tour) of this solution, we can split it into two separate tours:

1. A first tour containing only requests that were delivered via split delivery.
2. A second tour containing only requests that were fully served by a single vehicle (i.e., no split delivery).

Both resulting tours are free of split delivery and have lower or equal transportation costs than the origin tour due to the cost triangle inequality (1). If the cumulative prizes of this first tour are greater than the cumulative prizes of the second tour, we substitute the origin tour for the first tour; otherwise, we substitute the second tour. This substitution leads to an objective value of greater or equal to half of the objective value as before, since the sum of the cumulative prizes of the first and second tours equals the cumulative prizes of the origin tour. Repeating this process for all tours results in a feasible CPTP-IS solution that satisfies the inequality (3).

Now, consider a vehicle with capacity C and a request i that demands a quantity of $2C$ and offers profit $2C$, while $c_{0,i} < \epsilon$ for sufficiently small $\epsilon > 0$. In the CPTP-IS, the maximum deliverable quantity per request is C units, whereas in the SDCPTP-IS, due to split delivery, the entire $2C$ units can be delivered. As capacity C approaches infinity, transportation costs become negligible in the objective function, causing the ratio $\frac{Z(CPTP-IS)}{Z(SDCPTP-IS)}$ to converge to $\frac{1}{2}$. This demonstrates that the bound is tight, meaning no stronger bound exists. $\square$

Theorem 7 *If each request $i \in N$ demand satisfies $d_i \leq C$ and the cost matrix satisfies the triangle inequality, any solution:*

$$\frac{Z(CPTP)}{Z(SDCPTP - IS)} \geq \frac{1}{2} \quad (4)$$

and this bound is tight.

Proof Given an optimal SDCPTP-IS solution, where the total number of splits and incomplete services does not exceed the number of tours (as established in Theorem 4). For each request that obtains split delivery or incomplete service, (Fig. 4) we remove this request from the solution and create a separate out-and-back tour to serve this request. By Theorem 4, the number of such out-and-back tours is at most $|K|$. This new solution, which includes all these additional tours, may be infeasible since it could result in up to $2|K|$ tours. To address this, we order all these tours by non-increasing objective value (the collected prize in this tour minus their transportation

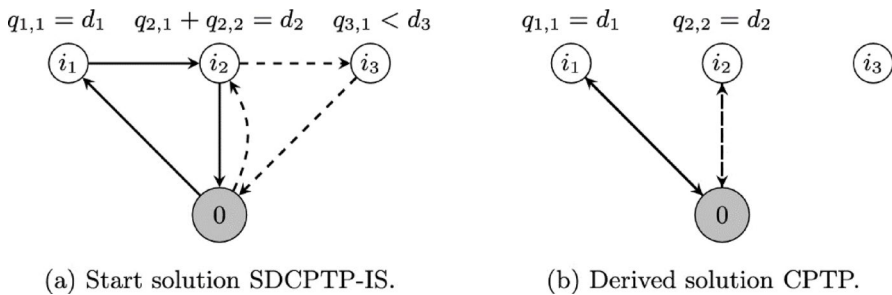


Fig. 4 An illustration of Theorem 7. Given the SDCPTP-IS solution in (a) that completely serves requests i_1 and i_2 , while request i_2 gets split delivery. Request i_3 is incompletely served. Without allowing incomplete services and split delivery, we still obtain a possible CPTP solution in (b) that satisfies the inequality (4) by completely serving requests i_1 and i_2

costs) and only take the first $|K|$ tours. Even in the special case where all tours have the same objective value, the objective value of the selected $|K|$ tours is at least half of the optimal solution $Z(\text{SDCPTP-IS})$. The tightness of this bound $\frac{1}{2}$ follows from Theorem 5. $\square$

The following Fig. 4 illustrates an example procedure for Theorem 7, where $p_1 - 2c_{0,1} \geq p_3 - 2c_{0,3} \leq p_2 - 2c_{0,2}$ and thus the derived CPTP solution in (b) satisfies inequality (4) by a given SDCPTP-IS solution in (a).

3.3 Mathematical programming formulations

In this section, we propose a MILP formulation for the SDCPTP-IS. This model is an undirected three-index vehicle flow and formulation based on inspiration from Jepsen et al. (2014) and Caspar et al. (2023). As an undirected MILP, it enables computational advantages, particularly for symmetric problems commonly studied in the literature. Building on the findings of Caspar et al. (2023) that have shown that the Bienstock subtour elimination constraints seem to perform best for the CPTP, we incorporate these subtour elimination constraints in the following SDCPTP-IS formulation. At the end of this section, we will also demonstrate how this SDCPTP-IS formulation can be adjusted to obtain a corresponding formulation where incomplete service or/and split delivery is not allowed. The notation for the model is summarized in Table 1.

Let us define $\delta(S \subseteq N) := \{(i, j) \mid i \in S, j \in V \setminus S\}$. The MILP model for the SDCPTP-IS (5)–(19) is:

$$\max \sum_{i \in N} \sum_{k \in K} p_i \cdot \frac{q_{i,k}}{d_i} - \sum_{k \in K} \sum_{i \in V} \sum_{\substack{j \in V \\ i < j}} c_{i,j} \cdot x_{i,j,k} - \sum_{k \in K} \sum_{j \in N} 2 \cdot c_{0,j} \cdot y_{j,k} \quad (5)$$

$$x_{i,j,k} = x_{j,i,k} \quad \forall i, j \in V (i < j), k \in K \quad (6)$$

Table 1 Common notation for the MILP models in Sect. 3.3

N	Set of requests $N = \{1, \dots, N \}$
V	Set of all vertices $V = \{0, \dots, N \}$
A	Set of arcs in the complete graph $A = \{(i, j) \mid i, j \in V (i \neq j)\}$
K	Set of vehicles $K = \{1, \dots, K \}$
C	Maximum vehicle capacity (identical for all vehicles $k \in K$)
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 $k \in K$. If $y_{0,k} = 1$ then vehicle k remains idle at the depot
$h_{i,k}$	Binary variable, equals 1 iff request $i \in N$ is visited by vehicle $k \in K$
$q_{i,k}$	Variable, that indicates the quantity delivered to $i \in N$ by $k \in K$

$$\sum_{i \in V} 2 \cdot y_{i,k} + \sum_{j \in N} x_{0,j,k} = 2 \quad \forall k \in K \quad (7)$$

$$2 \cdot y_{i,k} + \sum_{\substack{j \in V \\ i \neq j}} x_{i,j,k} = 2 \cdot h_{i,k} \quad \forall i \in N, k \in K \quad (8)$$

$$\sum_{i \in V} \sum_{\substack{j \in V \\ i \neq j}} x_{i,j,k} \leq 2 \cdot (|N| + 1) \cdot (1 - \sum_{i \in V} y_{i,k}) \quad \forall k \in K \quad (9)$$

$$\sum_{k \in K} h_{i,k} \leq z_i \cdot |K| \quad \forall i \in N \quad (10)$$

$$\sum_{k \in K} q_{i,k} \leq d_i \cdot z_i \quad \forall i \in N \quad (11)$$

$$\sum_{i \in N} q_{i,k} \leq C \quad \forall k \in K \quad (12)$$

$$h_{i,k} \leq q_{i,k} \leq h_{i,k} \cdot C \quad \forall i \in N, k \in K \quad (13)$$

$$\sum_{(i,j) \in \delta(S)} x_{i,j,k} \geq 2 \cdot h_{i,k} \quad \forall S \subseteq N, i \in S, k \in K \quad (14)$$

$$z_i \in \{0, 1\} \quad \forall i \in N \quad (15)$$

$$y_{i,k} \in \{0, 1\} \quad \forall i \in V, k \in K \quad (16)$$

$$h_{i,k} \in \{0, 1\} \quad \forall i \in N, k \in K \quad (17)$$

$$x_{i,j,k} \in \{0, 1\} \quad \forall i, j \in V (i \neq j), k \in K \quad (18)$$

$$q_{i,k} \in \mathbb{N}_0 \quad \forall i \in N, k \in K \quad (19)$$

The objective function (5) maximizes the difference between the gained prize, depending on the delivered and demanded quantity ratio over all requests and the transportation costs. The second term states the costs of all vehicle tours with two or more vertices besides the depot, while the third term accounts for the costs of vehicle tours that visit exactly one vertex before returning to the depot. Constraints (6) enforce the symmetry of the decision variables $x_{i,j,k}$. The equations in (7) guarantee that every vehicle k leaves the depot exactly once to start a tour that serves two or more requests (exactly two $x_{0,j,k}$ have value 1), serves a single request (exactly one $y_{i,k}, i \neq 0$ has value 1), or remains idle at the depot ($y_{0,k} = 1$). For each request i that is visited by vehicle k , constraints (8) force $h_{i,k}$ to take value 1. Constraints (9) are valid inequalities that express that at most $2 \cdot (|N| + 1)$ arcs (due to the symmetry of the decision variables $x_{i,j,k}$) can be selected to form a feasible tour for each vehicle k if and only if vehicle k does neither serve a single request nor remain idle at the depot (i. e., all $y_{i,k}$ have value 0). In principle, these valid inequalities are not required to formulate the problem; however, preliminary experiments have shown that they are quite effective in strengthening the linear relaxation of the formulation. Constraints (10) force z_i to take the value 1 if at least one vehicle visits request i and constraints (11) assure that the delivered quantities do not exceed the demand for each accepted request. The capacity constraints (12) limit the total delivered quantity of each vehicle k to be at most C . Consistency between the variables $q_{i,k}$ and $h_{i,k}$ is ensured by constraints (13). The subtour elimination constraints (14) 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. Finally, the domains of the variables are defined by (15)–(19). As the authors in Archetti et al. (2014b) note, the integrality requirement in (19) 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.

To obtain a CPTP-IS formulation, disallowing split delivery in the SDCPTP-IS formulation (5)–(19) is straightforward by replacing constraints (10) with inequalities (20).

$$\sum_{k \in K} h_{i,k} \leq z_i \quad \forall i \in N \quad (20)$$

To gain an SDCPTP formulation, we need to disallow incomplete service in the SDCPTP-IS formulation (5)–(19) by changing constraints (11) to equations (21):

$$\sum_{k \in K} q_{i,k} = d_i \cdot z_i \quad \forall i \in N \quad (21)$$

A corresponding CPTP formulation can be obtained, given the previous SDCPTP formulation and substituting inequalities (10) through inequalities (20).

4 A matheuristic for the SDCPTP-IS

Matheuristics are solution methods that embed mathematical programming inside a heuristic framework (Schermer et al. 2019) and have been applied to many different routing problems (see Archetti and Speranza (2014) and references therein). In this section, we first present a GLS heuristic to solve the CVRP. Afterward, we show a matheuristic that exploits this CVRP solution to solve the SDCPTP-IS (5)–(19) as a MILP. The notations for the algorithms are summarized in Table 2.

4.1 Guided local search for the CVRP

Local search $LS(\mathcal{S}, g)$ acts on a search space $\mathcal{S}$ under an objective function $g : \mathcal{S} \rightarrow \mathbb{R}$ to iteratively investigate neighboring solutions in $\mathcal{N}(s) \subset \mathcal{S}$ (solutions that can be

Table 2 Common notation for the algorithms in Sect. 4

$\mathcal{S}$	Search space for the guided local search
$g(\cdot)$	Objective function to investigate solutions during the search process
$LS(\mathcal{S}, g)$	Local search, acting on search space $\mathcal{S}$ with objective function g
$\mathcal{N}(\cdot)$	Neighboring solutions that a local search operator can reach
$(l_x, l_y)_i$	Cartesian coordinates of request locations or depot $i \in V$
(x)	CVRP solution, containing all binary variables $x_{i,j,k}$
$f_{i,j}$	Features characterize a solution to guide the search process
$\varphi_{i,j}$	Penalties associated with features $f_{i,j}$
$I_{i,j}(\cdot)$	Indicator function that checks if a specific solution has feature $f_{i,j}$
λ	Parameter that weighs the relative importance of penalties
$u_{i,j}(\cdot)$	Utility function that controls penalties during the search process
$H^q \kappa, H^z \kappa$	Embedding strategy for variables q or z depending on the number $\kappa (\kappa \leq K)$
$dist(\cdot)$	Distance function measures straight-line distances in Euclidean space
av_i	Average request value regarding prize, demand, and distance

reached from current solution $s \in S$ by an *LS* operator with small modifications) (see Aarts and Lenstra (2003)). Such *LS* operators define, how a new improved solution is generated in the neighborhood of the current solution $\mathcal{N}(s)$, typically by performing operations such as swapping two elements, adding or removing an element, or changing a property.

An advanced version of this *LS* is the *Guided Local Search* (GLS) (see Voudouris and Tsang (1999)) where features provide additional information to guide the *LS* during specific choices of solutions in the neighborhood. GLS penalizes, based on a utility function, unwanted solution features during each *LS* iteration and may escape from local optima to allow the search to continue. GLS has already shown very good results for various VRPs such as the VRP with time windows (Kilby et al. 1999), VRP with backhauls (Zhong and Cole 2005), or the OP (Vansteenwegen et al. 2009). However, to the best of our knowledge, GLS has not been used before to solve profitable tour problems.

Let l_x and l_y be the Cartesian coordinates of the request locations and the depot. In the context of the CVRP (for a MILP formulation see Toth and Vigo (2014)) as well as in the SDCPTP-IS, the distances between these request locations are suitable features (see Voudouris and Tsang (1999)), given as directed arc $f_{i,j} = (i, j) \in F$ that are traversed in the solution or not with $f_{i,j} \in \{0, 1\}$. In the following, the CVRP solution is stated by (x) , representing binary variables $x_{i,j,k}$, that equals 1 iff arc (i, j) is part of the tour of vehicle $k \in K$. Such a feature $f_{i,j}$ is represented by an indicator function:

$$I_{i,j}((x)) = \begin{cases} 1 & \text{if solution } (x) \text{ has feature } f_{i,j}, \\ 0 & \text{otherwise,} \end{cases} \quad \forall (x) \in S$$

Moreover, these features $f_{i,j}$ have a penalty $\varphi_{i,j}$ and a travel cost $c_{i,j}$. An augmented cost function g extends the CVRP objective function

$$\sum_{k \in K} \sum_{i \in V} \sum_{\substack{j \in V \\ j < i}} c_{i,j} \cdot x_{i,j,k} \quad \text{to:}$$

$$g((x)) = \sum_{k \in K} \sum_{i \in V} \sum_{\substack{j \in V \\ j < i}} c_{i,j} \cdot x_{i,j,k} + \lambda \sum_{(i,j) \in F} I_{i,j}((x)) \cdot \varphi_{i,j} \cdot c_{i,j}$$

where λ is a parameter that represents the relative importance of penalties and is 1 by default. Penalties $\varphi_{i,j}$ start with value 0.

$LS(S, g)$ returns the optimized routes and the total cost associated with these routes. The GLS iteratively uses $LS(S, g)$ to modify these penalties each time a local minimum is found. The penalties $\varphi_{i,j}$ are incremented by one that maximizes the utility expression:

$$u_{i,j}((x)) = I_{i,j}((x)) \frac{c_{i,j}}{1 + \varphi_{i,j}}$$

If a feature $f_{i,j}$ is penalized many times over several iterations, the term $\frac{c_{i,j}}{1+\varphi_{i,j}}$ decreases and enables other features to be penalized. Thus, a local minimum can be escaped and better solutions can be found until a termination criterion, e.g., a time limit is met (see Voudouris and Tsang (1999)).

Algorithm 1 Pseudo algorithm for the guided local search for solving a CVRP

Data: Given an instance
 $K, V, C, N = \{(p, d, (l_x, l_y))_0, \dots, (p, d, (l_x, l_y))_{|N|}\}$ with features F ,
parameter $\lambda = 1$, binary indicator function $I_{i,j}$, penalties $\varphi_{i,j}$,
time limit T and a local search LS

Result: CVRP solution $(x)_{it}$ (set N') with variables q^{GLS} and z^{GLS}
(set N)

- 1 start solution is determined by $LS(\mathcal{S}, \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{i,j} \cdot x_{i,j,k})$
- 2 $it = 1$
- 3 for $(i, j) \in F$ do:
- 4 $\varphi_{i,j} = 0$
- 5 $g := \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{i,j} \cdot x_{i,j,k} + \lambda \sum_{(i,j) \in F} I_{i,j}(x) \cdot \varphi_{i,j} \cdot c_{i,j}$
- 6 while computation time $< T$ do:
- 7 $(x)_{it} = LS(\mathcal{S}, g)$
- 8 for $(i, j) \in F$ do:
- 9 $u_{i,j}((x)_{it}) = I_{i,j}((x)_{it}) \frac{c_{i,j}}{1+\varphi_{i,j}}$
- 10 for $\arg \max_{(i,j) \in F} u_{i,j}$ do:
- 11 $\varphi_{i,j} = \varphi_{i,j} + 1$
- 12 $it = it + 1$
- 13 $(x)_{it}$ best found CVRP solution
- 14 define q^{GLS} and z^{GLS} according to the CVRP solution $(x)_{it}$
- 15 return q^{GLS} and z^{GLS} variables

Given an SDCPTP-IS instance with (K, V, C, N) , Algorithm 1 describes how to solve the corresponding CVRP instance (K, V, C, N') with a valid subset $N' \subseteq N$ to obtain integer variables q^{GLS} and binary variables z^{GLS} . A valid subset $N' \subseteq N$ must satisfy condition $\sum_{i \in N'} d_i \leq C \cdot |K|$ and there must be a partition of these demand quantities

$d_i \in N'$ between the vehicles $|K|$. The following example clarifies a valid subset for using Algorithm 1 to obtain a CVRP solution. Given two vehicles each with capacity $C = 5$ and a request set $N = \{(8, 4, (1, 0))_1, (6, 3, (0, 1))_2, (5, 3, (-1, 0))_3\}$ (in the form of $(p, d, (l_x, l_y))_i$). Then $N'_a = \{(8, 4, (1, 0))_1, (6, 3, (0, 1))_2, (5, 3, (-1, 0))_3\}$ is not a valid subset, since there is no possible partition of the demands between the two vehicles. However, a valid subset that satisfies both requirements is $N'_b = \{(8, 4, (1, 0))_1, (6, 3, (0, 1))_2\}$.

Algorithm 1 generates a start solution for the CVRP with local search. All feature penalties $\varphi_{i,j}$ are initialized by zero and the augmented cost function g is defined according to the CVRP. While the time limit T is not reached (see line 6 of Algo-

rithm 1) every iteration it tries to improve the current CVRP solution. The local search updates the current solution $(x)_{it}$ and updates the penalties $\varphi_{i,j}$ of the features $f_{i,j}$ to maximize the utility $u_{i,j}$. After the last iteration, the best CVRP solution $(x)_{it}$ (set N) is taken to determine integer variables q^{GLS} and binary variables z^{GLS} (set N'). These variables can be embedded into a matheuristic that solves an SDCPTP-IS instance.

4.2 The matheuristic

The matheuristic that exploits the CVRP solution from Algorithm 1 to solve the SDCPTP-IS (5)–(19) as a MILP is given with Algorithm 2.

Algorithm 2 Pseudo algorithm for the matheuristic solving the SDCPTP-IS MILP formulation

Data: Given an instance with

$K, V, C, N = \{(p, d, (l_x, l_y))_0, \dots, (p, d, (l_x, l_y))_{|N|}\}$, Algorithm 1 and an embedding strategy with $\kappa \leq |K|$

Result: SDCPTP-IS solution

- 1 calculate $av_i = \frac{p_i}{d_i \cdot dist((l_x, l_y)_i, (l_x, l_y)_0)} \forall i \in N$ and sort it in descending order
 - 2 define new request set $N' = \{(p, d, (l_x, l_y))_i\}$ by inserting according to the sorted av_i as long as $\sum_{i \in N'} d_i \leq C \cdot |K|$ and there is a valid partition of these demands $d_i \in N'$ to the $|K|$ vehicles
 - 3 use Algorithm 1 to solve $CVRP(K, V, C, N')$ to obtain the integer variables q^{GLS} and z^{GLS}
 - 4 solve the $SDCPTP-IS(K, V, C, N) \{q^{GLS} \vee z^{GLS}\}_\kappa$ with the MILP formulation of the SDCPTP-IS (5)–(19)
 - 5 return SDCPTP-IS solution
-

Algorithm 2 starts by determining the subset $N' \subseteq N$ based on the number of available vehicles $|K|$ and their capacity C that can be solved in a CVRP instance (K, V, C, N') with Algorithm 1. First, for each request $i \in N$ we calculate the distance of its location to the depot. This value influences av_i , which weights the request's prize with its associated demand d_i and depot distance (determined with the straight-line function $dist(\cdot)$ in Euclidean space). Sorting these values av_i in descending order, we sort requests into this new request set N' as long as the corresponding CVRP is solvable. Solving such a CVRP instance (K, V, C, N') with

Algorithm 1, we get the variables $q_{i,k}^{GLS}$ and $z_{i,k}^{GLS}$ of each vehicle $k \in K$ and request location $i \in N'$ (see line 3). Now we embed these variables (either q^{GLS} or/and z^{GLS}) into the SDCPTP-IS MILP formulation (5)–(19) (with set N) to obtain a feasible solution for the SDCPTP-IS. Depending on the objective and the available computation time, we must decide which and how many of these variables are embedded

(called embedding strategy). If we only embed variables $q_{i,k}^{GLS}$ or $z_{i,k}^{GLS}$ with $k \leq \kappa$ ($\kappa \leq |K|$) into the MILP formulation of the SDCPTP-IS, we call it $H^q\kappa$ or $H^z\kappa$ embedding strategy. Only embedding z^{GLS} variables does not fix any delivery quantities in the SDCPTP-IS MILP formulation; thus there is potential to optimize this

solution. Following an embedding strategy where the sum of q^{GLS} variables is lower than $C \cdot |K|$ leads to a residual capacity that can further improve this solution in the SDCPTP-IS MILP formulation (5)–(19).

With $\kappa = |K|$ all q^{GLS} or z^{GLS} variables are embedded and with $\kappa = 0$, no variables are embedded and the embedding strategy becomes equal to solving the original SDCPTP-IS formulation (5)–(19). While decreasing κ ($\kappa \leq |K|$) a larger objective function value is expected at the expense of longer computation times. In total, there are up to $|K| + 1$ possible embedding strategies for either q^{GLS} or z^{GLS} variables (see line 4 in Algorithm 2).

Analogously to the SDCPTP-IS formulation, the CPTP, SDCPTP, and CPTP-IS formulation can be solved with this matheuristic by adapting line 4 in Algorithm 2.

5 Computational experiments and numerical results

This section outlines our computational experiments and discusses their numerical results. All experiments were examined on a shared compute cluster equipped with Intel Xeon Gold 6126 processors. For our experiments, we used 8 cores and 64 GB of RAM. We implemented the MILP models for the CPTP, SDCPTP, CPTP-IS, and SDCPTP-IS with Python 3.10 and used the branch-and-cut solver Gurobi Optimizer 10.0 Gurobi Optimization (2023) with a time limit of 3600 s using 24 threads. Within our experiments, we also used highly optimized Google OR-tools v9.7 Perron and Furnon (2013) with different local search techniques.

In all MILP formulations, the subtour elimination constraints (see Equation (14)) are dynamically generated, by adding them only as necessary and not before needed. More precisely, at each node of the branch-and-cut tree that yields an integer-feasible solution, we use Gurobi's *callback* procedure to check for the presence of subtours. If subtours are detected, we identify the smallest subset of vertices S that forms a subtour and apply the corresponding constraint (14) as a *lazy constraint* during the callback. If no subtour is detected, we can be sure that the solution is valid.

The remainder of this section is structured as follows. We start in Sect. 5.1 with a description of the problem instances. Afterward, in Sect. 5.2, we discuss the computational results.

5.1 Instance selection

There exist well-established benchmark instances for vehicle routing problems in the literature, and we base our experiments on two different sets of instances. The first set of instances is based on CVRP instances from the CVRP-library (available at <http://vrp.atd-lab.inf.puc-rio.br/index.php/en/>). In particular, a total of 25 CVRP instances are chosen from different classes and analog to Archetti et al. (2009) the integer profits p_i of request $i \in N$ are defined as $(0.5 + b)d_i$ with b a random number uniformly generated over the interval $[0, 1]$. The CPTP formulation of Sect. 3 was used to obtain optimal CPTP solutions as benchmarks.

The second set was originally proposed for the CPTP by Archetti et al. (2009) (and also used in the split delivery CPTP by Caspar et al. (2023)), where the authors

adapted 10 instances from the well-known benchmark set for the vehicle routing problem from Christofides (1979). We choose 15 of these instances with between 50 to 199 requests (each associated with a demand and a prize) with a fleet size of $|K| \in \{2, 3, 4\}$, and a vehicle capacity of $C = 100$. These instances were optimally solved for the CPTP in Archetti et al. (2009).

We analyze to what extent the solution of the CPTP can be increased by split delivery and incomplete service as well as a combination of both within this context. Furthermore, we compare how each of the three MILP formulations SDCPTP-IS, SDCPTP, and CPTP-IS performs compared to the corresponding matheuristic (see Algorithm 2). All instances are available at <https://doi.org/10.5281/zenodo.8394665>.

5.2 Computational results

We begin by presenting the numerical results for the MILP formulation of the SDCPTP and a corresponding matheuristic (see Algorithm 2), where only split delivery is allowed, and incomplete service is not permitted.

The first column lists the instance according to the number of requests N and vehicles K , while the corresponding optimal CPTP solution Z^* (taken from Archetti et al. (2009)) is shown in the second column. In the third and fourth columns, we see the performance of the MILP formulation for the SDCPTP (5)–(19), specifically the best feasible solution Z and the computation time in seconds. If the solution was proven to be optimal, i.e., the remaining *Mixed-Integer Programming (MIP)* gap is zero (within the default tolerance), the exact time is displayed. Otherwise, the remaining MIP gap after 3600 s is reported as a percentage. The same applies to the next double column, where the performance of the matheuristic MH^q with embedding q^{GLS} variables into the MILP formulation (see Algorithm 2) is shown using the best embedding strategy. Subsequently, the double column MH^z shows the maximum objective function value Z under all possible embedding strategies in Algorithm 2 within 900 s of computation time after the GLS variables have been determined in 60 s with Algorithm 1. If no value is shown in the columns MH^q and MH^z , the optimal solution was already found within 900 s under the MILP formulation and the value follows from the column *SDCPTP*. Column v_{SD} indicates the number of accepted requests delivered through split delivery. The final column (Imp.) quantifies the improvement of the SDCPTP 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 the SDCPTP MILP formulation and the matheuristic within Algorithm 2.

In Table 3, we see that the MILP formulation for the SDCPTP could not find optimal solutions in 12 out of 40 cases within 3600 s of computation time. In 6 of these 12 cases, the matheuristic with the best embedding strategy found a qualitative better solution (higher objective value or same value in less computation time) in even a quarter of the MILP computation time. The matheuristic with embedding q^{GLS} variables (see column MH^q) already finds solutions that are approximately 0.9 % better (compared to the objective function value) than the optimal CPTP solution after only 347 s on average. By embedding z^{GLS} variables (see column MH^z), the matheuristic finds roughly 0.7 % better solutions after 455 s compared to the optimal

Table 3 Numerical results of the SDCPTP MILP formulation and matheuristic for the SDCPTP on instances based on the CVRP-library and Archetti et al. (2009)

Instance ($ N , K $)	CPTP		SDCPTP		MH^q		MH^z		v_{SD}	Imp. (%)
	Z^*	Z	t or g	Z	t or g	Z	t or g	Z		
Ar3-n101-k2	158.21	162.20	53.9 s						1/10	2.5
Ar3-n101-k3	218.63	222.62	1461 s	222.62 H0	2.3 %	222.62 H0	2.3 %		1/16	1.8
Ar3-n101-k4	255.33	246.61	21.4 %	267.42 H1	301 s	267.15 H2	7.8 %		1/23	4.7
Ar6-n51-k2	100.27	100.27	3.2 s						0/12	0
Ar6-n51-k3	134.72	134.72	23.2 s						0/16	0
Ar6-n51-k4	153.30	154.59	691 s						2/21	0.8
Ar7-n76-k2	132.70	133.20	22.0 s						1/11	0.4
Ar7-n76-k3	185.25	187.37	183 s						2/15	1.1
Ar7-n76-k4	233.40	234.86	4.3 %	234.86 H1	174 s	234.86 H1	6.0 %		1/18	0.6
Ar9-n151-k2	161.23	161.23	415 s						0/11	0
Ar9-n151-k3	230.49	229.80	5.9 %	226.13 H1	867 s	224.03 H0	13.0 %		1/19	-0.3
Ar9-n151-k4	290.54	271.85	21.5 %	265.68 H2	12.3 %	259.32 H2	23.6 %		1/23	-6.4
Ar10-n200-k2	171.24	171.24	1537 s	168.98 H0	5.4 %	168.98 H0	5.4 %		0/10	0
Ar10-n200-k3	250.18	244.78	11.5 %	242.84 H1	720 s	239.20 H2	846 s		1/15	-2.2
Ar10-n200-k4	324.02	315.57	17.1 %	293.74 H2	487 s	298.03 H3	4.2 %		2/22	-2.6
A-n32-k5	18.52	18.52	2.2 s						0/6	0
A-n33-k6	52.11	52.46	40 s						1/8	0
A-n34-k5	18.88	18.88	17.6 s						0/5	0
A-n37-k5	22.74	22.74	7.6 s						0/5	0
A-n37-k6	18.84	18.84	22.5 s						0/11	0
A-n38-k5	71.04	71.04	1316 s	71.04 H0	8.8 %	71.04 H0	8.8 %		0/20	0
A-n44-k6	50.88	50.88	10.0 s						0/11	0
A-n45-k6	18.61	18.61	57.9 s						0/11	0
A-n46-k7	26.48	26.48	38.4 %	26.48 H0	81.3 %	26.48 H0	81.3 %		0/7	0
A-n54-k7	31.86	31.86	64.3 %	31.86 H0	106 %	31.86 H0	106 %		0/8	0
E-n22-k4	4464.7	5062.4	0.4 s						2/4	13.4
E-n23-k3	2853.4	2956.7	0.6 s						2/10	3.6
E-n30-k3	2490.8	2602.0	1.2 s						2/5	4.5
E-n51-k5	206.20	219.00	1475 s	219.00 H0	2.4 %	219.00 H0	2.4 %		3/26	6.2
E-n76-k7	614.62	544.44	29.6 %	607.47 H2	3.8 %	473.93 H0	50 %		1/69	-1.2
F-n45-k4	3998.6	4505.0	1.3 %	4503.6 H0	2.4 %	4503.6 H0	2.4 %		2/14	12.7
F-n72-k4	98134	98138	97.1 s						1/71	0
P-n16-k8	248.05	253.63	35.2 s						1/10	2.2
P-n19-k2	77.75	77.75	0.3 s						0/7	0
P-n22-k8	18409	18433	3532 s	18433 H2	8 s	18433 H0	1.1 %		2/21	0.1
P-n40-k5	241.85	243.57	129 s						1/28	0.7
P-n45-k5	225.44	225.56	233 s						1/33	0.1
P-n76-k4	782.69	782.69	1.3 %	778.43 H1	66 s	770.33 H0	4.8 %		0/71	0
P-n76-k5	847.34	846.88	2.7 %	847.34 H1	209 s	842.38 H2	3.8 %		0/71	0
P-n101-k4	886.35	886.35	1781 s	875.97 H0	3.1 %	875.97 H0	3.1 %		0/84	0
Avg.	3448.4	3477.0	5.5 %	3477.8	347 s	3473.9	455 s		0.8/21	1.1

CPTP solution by mean. Regarding the matheuristic, in 20 out of 36 cases embedding no variables leads to the best results under all possible embedding strategies. With the best-known solution from the MILP formulation and the matheuristic, the optimal CPTP solution can be increased by allowing split delivery by approximately 1.1 %. It should be noted that these best-known SDCPTP solutions still have an MIP gap of around 2.8 % on average. However, only 0.8 of circa 21 accepted requests by means are provided with split delivery in these solutions.

Next, we illustrate the numerical results for the CPTP-IS in Table 4. Analog to Table 3, we describe in column v_{IS} how many of the accepted requests are delivered with incomplete service.

In Table 4, we see that the MILP formulation for the CPTP-IS could not find optimal solutions in 9 out of 40 cases within our computational burden of 3600 s. In 4 of these 9 cases, the matheuristic with the best embedding strategy found a qualitative better solution in even a quarter of the MILP computation time. The matheuristic with embedding q^{GLS} variables (see column MH^q) already finds approximately 0.9 % better solutions than the optimal CPTP solution after only 234 s on average. By embedding z^{GLS} variables (see column MH^z), the matheuristic finds roughly 0.8 % better solutions after 346 s compared to the optimal CPTP solution by means. Regarding the matheuristic, the best embedding strategies vary between embedding zero and the first two vehicle variables in this context. With the best-known solution from the MILP formulation and the matheuristic, the optimal CPTP solution can be increased by allowing incomplete service by approximately 4.4 %. It should be noted that these best-known CPTP-IS solutions still have an MIP gap of around 1.5 % on average. However, 1.8 of circa 22 accepted requests by means are provided with enabling incomplete service in these solutions.

Finally, we allow both split delivery and incomplete service and obtain the following numerical results for the SDCPTP-IS in Table 5. As illustrated in the Tables 3 and 4, we describe in column v_{SD-IS} how many of the accepted requests are delivered with split delivery, respectively, incomplete service.

In Table 5, we see that the MILP formulation for the SDCPTP-IS could not find optimal solutions in 11 out of 40 cases within 3600 s of computation time. In 10 of these 12 cases, the matheuristic with the best embedding strategy found a qualitative better solution (higher objective value or same value in less computation time) in even a quarter of the MILP computation time. The matheuristic with embedding q^{GLS} variables (see column MH^q) already finds approximately 1.1 % better solutions than the optimal CPTP solution after only 252 s on average. By embedding z^{GLS} variables (see column MH^z), the matheuristic finds roughly 0.8 % better solutions after 407 s compared to the optimal CPTP solution by means. Regarding the matheuristic, in 19 out of 23 cases the H1 embedding strategy, where only variables of the first vehicle are embedded leads to the best results under all possible embedding strategies. With the best-known solution from the MILP formulation and the matheuristic, the optimal CPTP solution can be increased by allowing split delivery and incomplete service by approximately 4.5 %. In this context, it should be noted that these best-known SDCPTP-IS solutions still have an MIP gap of around 1.1 % on average. However, only 0.4 of circa 21 accepted requests by means are provided

Table 4 Numerical results of the CPTP-IS MILP formulation and matheuristic for the CPTP-IS on instances based on the CVRP-library and Archetti et al. (2009)

Instance ($ N , K $)	CPTP		CPTP-IS		MH^q		MH^z		v_{IS}	Imp. (%)
	Z^*	Z	t or g	Z	t or g	Z	t or g	Z		
Ar3-n101-k2	158.21	163.46	285 s						2/10	3.3
Ar3-n101-k3	218.63	226.83	803 s						3/16	3.8
Ar3-n101-k4	255.33	273.03	9.2 %	270.97 H1	103 s	260.80 H0	15.4 %		4/26	6.9
Ar6-n51-k2	100.27	105.75	1.6 s						2/11	5.5
Ar6-n51-k3	134.72	136.34	13.3 s						1/17	1.2
Ar6-n51-k4	153.30	156.90	83.3 s						2/23	2.3
Ar7-n76-k2	132.70	137.32	7.1 s						2/11	3.5
Ar7-n76-k3	185.25	191.30	55.9 s						3/17	3.3
Ar7-n76-k4	233.40	240.68	410 s						4/20	3.1
Ar9-n151-k2	161.23	164.81	861 s						1/10	2.2
Ar9-n151-k3	230.49	230.80	7.4 %	224.59 H2	2.6 %	230.29 H0	9.1 %		2/17	0.1
Ar9-n151-k4	290.54	218.15	53.7 %	276.59 H2	9.6 %	222.50 H2	53.5 %		4/28	-4.8
Ar10-n200-k2	171.24	175.69	3308 s	175.69 H0	2.5 %	175.69 H0	2.5 %		1/10	2.6
Ar10-n200-k3	250.18	250.31	8.5 %	237.55 H2	244 s	244.08 H0	12.7 %		3/17	0.1
Ar10-n200-k4	324.02	296.70	23.7 %	288.00 H2	7.3 %	196.04 H0	89.8 %		3/21	-8.4
A-n32-k5	18.52	21.88	2.3 s						1/5	18.1
A-n33-k6	52.11	59.37	13.3 s						2/10	13.9
A-n34-k5	18.88	22.68	11.4 s						1/6	20.1
A-n37-k5	22.74	23.72	4.9 s						1/6	4.3
A-n37-k6	18.84	21.31	12.2 s						1/11	13.1
A-n38-k5	71.04	72.64	7.5 s						2/21	2.3
A-n44-k6	50.88	54.22	28.7 s						1/12	6.6
A-n45-k6	18.61	19.39	78.8 s						1/7	4.2
A-n46-k7	26.48	30.18	263 s						1/8	14.0
A-n54-k7	31.86	34.87	239 s						1/8	9.4
E-n22-k4	4464.7	5130.5	0.2 s						4/7	14.9
E-n23-k3	2853.4	2964.8	8.3 s						3/13	3.9
E-n30-k3	2490.8	2600.5	3.4 s						3/5	4.4
E-n51-k5	206.20	218.81	96 s						4/29	6.1
E-n76-k7	614.62	601.66	9.2 %	609.74 H2	8.5 %	587.52 H0	23.7 %		1/69	-0.8
F-n45-k4	3998.6	4453.7	161 s						4/23	11.4
F-n72-k4	98134	98134	242 s						0/71	0
P-n16-k8	248.05	258.52	8.3 s						2/11	4.2
P-n19-k2	77.75	79.08	0.2 s						1/8	1.7
P-n22-k8	18409	18409	0.1 %	18409 H4	0.6 s	18409 H0	0.2 %		0/21	0
P-n40-k5	241.85	245.28	78 s						1/28	1.4
P-n45-k5	225.44	229.97	96 s						2/34	2.0
P-n76-k4	782.69	782.69	0.6 %	778.43 H1	56 s	776.92 H0	3.4 %		0/71	0
P-n76-k5	847.34	841.03	4.6 %	847.34 H1	150 s	847.34 H2	3.0 %		0/72	0
P-n101-k4	886.35	886.35	1716 s	865.34 H1	352 s	881.91 H1	2.5 %		0/84	0
Avg.	3448.4	3479.1	11.9 %	3480.6	234 s	3476.8	346 s		1.8/22	4.4

Table 5 Numerical results of the SDCPTP-IS MILP formulation and matheuristic for the SDCPTP-IS on instances based on the CVRP-library and Archetti et al. (2009)

Instance	CPTP	SDCPTP-IS		MH^q		MH^z		v_{SD-IS}	Imp.
(N , K)	Z^*	Z	t or g	Z	t or g	Z	t or g		(%)
Ar3-n101-k2	158.21	163.46	162 s					(0,2)/10	3.3
Ar3-n101-k3	218.63	226.83	0.1 %	211.12 H1	47 s	226.83 H1	3.2 %	(0,3)/16	3.8
Ar3-n101-k4	255.33	262.20	14.6 %	271.26 H1	147 s	246.66 H2	24.1 %	(1,2)/24	6.2
Ar6-n51-k2	100.27	105.75	2.2 s					(0,2)/11	5.5
Ar6-n51-k3	134.72	136.34	11.3 s					(0,1)/17	1.2
Ar6-n51-k4	153.30	156.90	136 s					(0,2)/23	2.3
Ar7-n76-k2	132.70	137.32	9.7 s					(0,2)/11	3.5
Ar7-n76-k3	185.25	191.30	66.2 s					(0,3)/17	3.3
Ar7-n76-k4	233.40	240.68	892 s					(0,4)/20	3.1
Ar9-n151-k2	161.23	164.81	867 s					(0,1)/10	2.2
Ar9-n151-k3	230.49	223.88	13.1 %	226.63 H1	6.2 %	224.76 H1	14.1 %	(0,2)/21	−1.7
Ar9-n151-k4	290.54	124.09	176 %	271.54 H1	16.3 %	49.84 H1	594 %	(0,4)/26	−6.5
Ar10-n200-k2	171.24	168.99	11.1 %	169.03 H1	855 s	173.90 H1	9.0 %	(0,3)/13	1.6
Ar10-n200-k3	250.18	233.59	19.0 %	237.55 H2	232 s	212.91 H3	31.8 %	(0,2)/18	−5.0
Ar10-n200-k4	324.02	26.29	1314 %	306.11 H1	13.0 %	297.73 H3	26.7 %	(0,3)/20	−5.5
A-n32-k5	18.52	21.88	1.4 s					(0,1)/5	18.1
A-n33-k6	52.11	59.37	10.9 s					(0,2)/10	13.9
A-n34-k5	18.88	22.68	7.5 s					(0,1)/6	20.1
A-n37-k5	22.74	23.72	5.2 s					(0,1)/6	4.3
A-n37-k6	18.84	21.30	13.9 s					(0,1)/11	13.1
A-n38-k5	71.04	73.41	508 s					(1,1)/20	3.3
A-n44-k6	50.88	54.22	17.5 s					(0,1)/12	6.6
A-n45-k6	18.61	19.39	35.4 s					(0,1)/7	4.2
A-n46-k7	26.48	30.18	326 s					(0,1)/8	14.0
A-n54-k7	31.86	34.87	882 s					(0,1)/8	9.4
E-n22-k4	4464.7	5251.4	1.3 s					(2,3)/6	17.6
E-n23-k3	2853.4	2968.4	1.0 s					(2,1)/12	4.0
E-n30-k3	2490.8	2638.7	18.4 s					(1,2)/5	5.9
E-n51-k5	206.20	221.85	331 s					(2,2)/27	7.6
E-n76-k7	614.62	588.33	14.3 %	610.14 H2	3.3 %	544.80 H1	35.1 %	(0,1)/69	−0.7
F-n45-k4	3998.6	4319.9	7.3 %	4391.2 H1	40 s	4478.1 H3	4.2 %	(2,1)/19	12.0
F-n72-k4	98134	98138	90.2 s					(1,0)/71	0.0
P-n16-k8	248.05	258.52	25.6 s					(0,2)/11	4.2
P-n19-k2	77.75	79.08	0.2 s					(0,1)/8	1.7
P-n22-k8	18409	18434	0.1 %	18433 H4	3.2 s	18434 H0	0.2 %	(2,0)/21	0.1
P-n40-k5	241.85	245.28	99.8 s					(0,1)/28	1.4
P-n45-k5	225.44	229.97	101 s					(0,2)/34	2.0
P-n76-k4	782.69	779.18	2.5 %	778.43 H1	59 s	775.63 H1	4.2 %	(1,0)/71	−0.4
P-n76-k5	847.34	818.82	8.1 %	847.34 H1	71 s	841.94 H1	5.0 %	(0,0)/72	0.0
P-n101-k4	886.35	886.35	2774 s	865.34 H1	424 s	886.35 H1	841 s	(0,0)/84	0.0
Avg.	3448.4	3472.4	158 %	3482.7	252 s	3477.0	407 s	(0.4,1.5)/22	4.5

with split delivery in these solutions, while around 1.5 of them obtain incomplete service.

Summarizing the experimental results, it appears that allowing only incomplete service in the CPTP yields significantly better objective function values compared to allowing only split delivery. As Table 5 has shown, a combination of SD and IS in the CPTP only minorly improves objective function values on average compared to the CPTP-IS. Moreover, we can see that the matheuristic (see Algorithm 2) consistently provides better solutions within just 900 s of computation time, outperforming the corresponding MILP formulations after 3600 s, as well as the optimal CPTP solution. Especially for larger instances, the MILP formulations exhibit relatively poor performance after one hour of computation, highlighting the need for a matheuristic. For all three problems, respectively SDCPTP-IS, SDCPTP, and CPTP-IS, the q^{GLS} embedding strategy in the matheuristic has shown better performance on average than the z^{GLS} embedding strategy and the respective standard MILP formulation. In conclusion, embedding only the delivered quantity variables q^{GLS} of the first vehicle (H1 embedding strategy) into the matheuristic tends to provide the best performance across all experiments.

As managerial insights, it can be concluded to prioritize enabling IS over SD for transportation planning in problems such as the CPTP. On the one hand, our computational results show that enabling IS improves the CPTP solution by 4.4 %, compared to only 1.1 % with SD on average; on the other hand, in cases where incorporating SD leads to significant improvements, IS achieves a similar improvement, though not vice versa. Thus, split delivery in the context of the CPTP should rather be considered under specific conditions, such as low vehicle capacity or high-demand requests, and not for the sole purpose of improving the CPTP's objective value. The sensitivity analysis also suggests that relying only on MILP formulations for solving the SDCPTP-IS, SDCPTP, and CPTP-IS may not be advisable, as our matheuristic provides comparatively better solutions on average. Since the matheuristic performs better than the MILP formulations while requiring only a quarter of the computation time, we recommend using matheuristics to solve the SDCPTP-IS, SDCPTP, and CPTP-IS, especially for time-sensitive transportation planning. However, incorporating SD and IS increases the computational effort for solving the CPTP, necessitating careful evaluation to determine whether their integration is worthwhile.

6 Conclusions and future research directions

This paper introduces the *Split Delivery Capacitated Profitable Tour Problem with Incomplete Service* (SDCPTP-IS). The SDCPTP-IS extends the profitable tour problem by a fleet of vehicles and the possibility of split delivery and incomplete service. A concise overview of the contributions from the literature on the profitable tour problem is provided, highlighting a scientific gap in exploring split delivery and incomplete service in this context. To address this gap, this paper introduces a MILP formulation for the SDCPTP-IS and identifies key properties regarding objective bounds and optimal solutions with their maximum number of split deliveries and incomplete services. A *Guided Local Search* (GLS) heuristic solves a related capaci-

tated vehicle routing problem and embeds this solution into a matheuristic framework for solving the SDCPTP-IS, SDCPTP, and CPTP-IS.

Computational experiments have shown that this matheuristic can provide better solutions for the SDCPTP-IS, SDCPTP, and CPTP-IS in significantly shorter computation times compared to the corresponding MILP models. Among the tested embedding strategies in the matheuristic, embedding only the quantity delivery variables of the first vehicle has yielded the best results across all experiments. In summary, the experiments indicate that incomplete service contributes significantly more to improving a CPTP solution than split delivery in terms of objective function value. Solutions of the SDCPTP-IS with SD and IS do not appear to offer substantial improvements over CPTP-IS solutions where only IS is possible. However, our computational experiments suggest that while SD and IS can enhance solution quality, their implementation should be carefully evaluated to ensure that the additional computational effort is justified.

Despite these promising results, this work has some limitations. One key limitation is the fact that there are instances where the best matheuristic solution still exhibits computational gaps, indicating that even better solutions may be achievable with more computational resources. A more refined embedding strategy that embeds variables based on requests rather than just vehicles could offer an opportunity to further improve SDCPTP, CPTP-IS, and SDCPTP-IS solutions. Another limitation lies in the MILP formulations, which may lack computational efficiency or be sensitive to specific problem instances.

Based on this paper, there are also several future research directions. The matheuristic that embeds the GLS heuristic has shown promising results for SDCPTP-IS, SDCPTP, and CPTP-IS, warranting further testing with a broader range of instances to verify if the incorporation of SD and IS in such problems is worthwhile. Moreover, in the context of practical applications, where information about requests might not be deterministic, it would be worthwhile to consider stochastic variants of the SDCPTP-IS and the search for solution methods for such problems.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aarts EH, Lenstra JK (2003) Local search in combinatorial optimization. Princeton University Press
- Archetti C, Bianchessi N, Speranza MG (2013) The capacitated team orienteering problem with incomplete service. *Optimization Lett* 7(7):1405–1417

- Archetti C, Bianchessi N, Speranza MG, Hertz A (2014) Incomplete service and split deliveries in a routing problem with profits. *Networks* 63(2):135–145
- Archetti C, Bianchessi N, Speranza MG (2013) Optimal solutions for routing problems with profits. *Discret Appl Math* 161(4–5):547–557
- Archetti C, Bianchessi N, Speranza MG, Hertz A (2014) The split delivery capacitated team orienteering problem. *Networks* 63(1):16–33
- Archetti C, Corberán Á, Plana I, Sanchis JM, Speranza MG (2015) A matheuristic for the team orienteering arc routing problem. *Eur J Oper Res* 245(2):392–401
- Archetti C, Feillet D, Hertz A, Speranza MG (2009) The capacitated team orienteering and profitable tour problems. *J Op Res Soc* 60(6):831–842
- Archetti C, Speranza MG (2014) A survey on matheuristics for routing problems. *EURO J Comput Optimiz* 2(4):223–246
- Azadi Moghaddam Arani A, Jolai F, Nasiri MM (2019) A multi-commodity network flow model for railway capacity optimization in case of line blockage. *Int J Rail Transp* 7(4):297–320
- Balas E (1989) The prize collecting traveling salesman problem. *Networks* 19(6):621–636
- Bienstock D, Goemans MX, Simchi-Levi D, Williamson D (1993) A note on the prize collecting traveling salesman problem. *Math Program* 59(1):413–420
- Bortfeldt A, Yi J (2020) The split delivery vehicle routing problem with three-dimensional loading constraints. *Eur J Oper Res* 282(2):545–558
- Boysen N, Fedtke S, Schwerdfeger S (2021) Last-mile delivery concepts: a survey from an operational research perspective. *OR Spectrum* 43(1):1–58
- Bruni ME, Brusco L, Ielpa G, Beraldi P (2019) The risk-averse profitable tour problem. In: ICORES. pp. 459–466
- Bulhões T, Ha MH, Martinelli R, Vidal T (2018) The vehicle routing problem with service level constraints. *Eur J Oper Res* 265(2):544–558
- Caspar M, Schermer D, Wendt O (2023) Formulations for the split delivery capacitated profitable tour problem. In: International Conference on Computational Science and Its Applications. pp. 82–98. Springer
- Chbichib A, Mellouli R, Chabchoub H (2012) Profitable vehicle routing problem with multiple trips: modeling and variable neighborhood descent algorithm. *Am J Op Res* 2(6):104–119
- Chentli H, Ouafi R, Cherif-Khettaf WR (2018) Behaviour of a hybrid ils heuristic on the capacitated profitable tour problem. In: ICORES. pp. 115–123
- Christofides N (1979) The vehicle routing problem. *Combinatorial optimization*
- Cortés-Murcia DL, Afsar HM, Prodhon C (2019) A branch and price algorithm for the electric capacitated profitable tour problem with mandatory stops. *IFAC-PapersOnLine* 52(13):1572–1577
- Cortés-Murcia DL, Afsar HM, Prodhon C (2023) Multi-period profitable tour problem with electric vehicles and mandatory stops. *Int J Sustain Transp* 17(5):473–489
- Dasari KV, Pandiri V, Singh A (2021) Multi-start heuristics for the profitable tour problem. *Swarm Evol Comput* 64:100897
- Dell'Amico M, Maffioli F, Värbrand P (1995) On prize-collecting tours and the asymmetric travelling salesman problem. *Int Trans Oper Res* 2(3):297–308
- Dror M, Trudeau P (1989) Savings by split delivery routing. *Transp Sci* 23(2):141–145
- Feillet D, Dejax P, Gendreau M (2005) Traveling salesman problems with profits. *Transp Sci* 39(2):188–205
- Fischetti M, Fischetti M (2018) Matheuristics. In: *Handbook of heuristics*, pp. 121–153. Springer
- Frizzell PW, Giffin JW (1995) The split delivery vehicle scheduling problem with time windows and grid network distances. *Comput Op Res* 22(6):655–667
- Gulczynski D, Golden B, Wasil E (2010) The split delivery vehicle routing problem with minimum delivery amounts. *Transp Res Part E Logist Transp Rev* 46(5):612–626
- Gunawan A, Lau HC, Vansteenwegen P (2016) Orienteering problem: a survey of recent variants, solution approaches and applications. *Eur J Oper Res* 255(2):315–332
- Gunawan A, Ng KM, Kendall G, Lai J (2018) An iterated local search algorithm for the team orienteering problem with variable profits. *Eng Optim* 50(7):1148–1163
- Gurobi Optimization LLC (2023) Gurobi Optimizer Reference Manual
- Jepsen MK, Petersen B, Spoorendonk S, Pisinger D (2014) A branch-and-cut algorithm for the capacitated profitable tour problem. *Discret Optim* 14:78–96
- Karsten CV, Pisinger D, Ropke S, Brouer BD (2015) The time constrained multi-commodity network flow problem and its application to liner shipping network design. *Transp Res Part E Logist Transp Rev* 76:122–138

- Kilby P, Prosser P, Shaw P (1999) Guided local search for the vehicle routing problem with time windows. *Meta-heuristics: Advances and trends in local search paradigms for optimization* pp. 473–486
- Labadie N, Mansini R, Melechovský J, Calvo RW (2012) The team orienteering problem with time windows: an lp-based granular variable neighborhood search. *Eur J Oper Res* 220(1):15–27
- Labadie N, Melechovský J, Wolfler Calvo R (2011) Hybridized evolutionary local search algorithm for the team orienteering problem with time windows. *J Heurist* 17:729–753
- Lahyani R, Khemakhem M, Semet F (2017) A unified matheuristic for solving multi-constrained traveling salesman problems with profits. *EURO J Comput Optimiz* 5(3):393–422
- Lera-Romero G, Miranda-Bront JJ (2021) A branch and cut algorithm for the time-dependent profitable tour problem with resource constraints. *Eur J Oper Res* 289(3):879–896
- Noroozi A, Mazdeh MM, Heydari M, Rasti-Barzoki M (2018) Coordinating order acceptance and integrated production-distribution scheduling with batch delivery considering third party logistics distribution. *J Manuf Syst* 46(2):29–45
- Pěnička R, Faigl J, Saska M (2019) Variable neighborhood search for the set orienteering problem and its application to other orienteering problem variants. *Eur J Oper Res* 276(3):816–825
- Perron L, Furnon V. Or-tools, (2013) <https://developers.google.com/optimization/>
- Schermer D, Moeini M, Wendt O (2019) A matheuristic for the vehicle routing problem with drones and its variants. *Transp Res Part C Emerg Technol* 106:166–204
- Sevklı Z, Sevilgen FE (2006) Variable neighborhood search for the orienteering problem. In: *international symposium on computer and information sciences*. pp. 134–143. Springer
- Sun P, Veelenturf LP, Dabia S, Van Woensel T (2018) The time-dependent capacitated profitable tour problem with time windows and precedence constraints. *Eur J Oper Res* 264(3):1058–1073
- Toth P, Vigo D (eds.) (2014) *Vehicle routing: problems, methods, and applications*. Society for Industrial and Applied Mathematics (SIAM)
- Tsiligrirides T (1984) Heuristic methods applied to orienteering. *J Op Res Soc* 35(9):797–809
- Vansteenwegen P, Souffriau W, Berghe GV, Van Oudheusden D (2009) A guided local search metaheuristic for the team orienteering problem. *Eur J Oper Res* 196(1):118–127
- Voudouris C, Tsang E (1999) Guided local search and its application to the traveling salesman problem. *Eur J Oper Res* 113(2):469–499
- Wang P, Li H, Zhang T, Zhang S, Shi K (2019) A novel dynamic multi-source multi-sink flow algorithm over the satellite networks. In: *international conference on wireless and satellite systems*. pp. 303–311. Springer
- Wang X, Golden B, Gulczynski D (2014) A worst-case analysis for the split delivery capacitated team orienteering problem with minimum delivery amounts. *Optimiz Lett* 8(8):2349–2356
- Yu Q, Fang K, Zhu N, Ma S (2019) A matheuristic approach to the orienteering problem with service time dependent profits. *Eur J Oper Res* 273(2):488–503
- Zhang M, Wang J, Liu H (2017) The probabilistic profitable tour problem. *Int J Enterprise Inf Syst (IJEIS)* 13(3):51–64
- Zhang T, Li H, Li J, Zhang S, Shen H (2018) A dynamic combined flow algorithm for the two-commodity max-flow problem over delay-tolerant networks. *IEEE Trans Wireless Commun* 17(12):7879–7893
- Zhong Y, Cole MH (2005) A vehicle routing problem with backhauls and time windows: a guided local search solution. *Transp Res Part E Logist Transp Rev* 41(2):131–144

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.