Applied Mathematical Modelling 80 (2020) 290–318

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

A truck and drones model for last-mile delivery: A mathematical model and heuristic approach

Mohammad Moshref-Javadi[a,*], Ahmad Hemmati[b], Matthias Winkenbach[a]

[a] Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[b] University of Bergen, Bergen 5020, Norway

ARTICLE INFO

Article history:
Received 7 April 2019
Revised 3 November 2019
Accepted 13 November 2019
Available online 18 November 2019

Keywords:
Last-mile delivery
UAV logistics
Mixed-integer programming
Large neighborhood search

ABSTRACT

We present a mathematical formulation and a heuristic solution approach for the optimal planning of delivery routes in a multi-modal system combining truck and Unmanned Aerial Vehicle (UAV) operations. In this system, truck and UAV operations are synchronized, i.e., one or more UAVs travel on a truck, which serves as a mobile depot. Deliveries can be made by both UAVs and the truck. While the truck follows a multi-stop route, each UAV delivers a single shipment per dispatch. The presented optimization model minimizes the waiting time of customers in the system. The model determines the optimal allocation of customers to truck and UAVs, the optimal route sequence of the truck, and the optimal launch and reconvene locations of the UAVs along the truck route. We formulate the problem as a Mixed-Integer Linear Programming (MILP) model and conduct a bound analysis to gauge the maximum potential of the proposed system to reduce customer waiting time compared to a traditional truck-only delivery system. To be able to solve real-world problem size instances, we propose an efficient Truck and Drone Routing Algorithm (TDRA). The solution quality and computational performance of the mathematical model and the TDRA are compared together and with the truck-only model based on a variety of problem instances. Further, we apply the TDRA to a real-world case study for e-commerce delivery in São Paulo, Brazil. Our numerical results suggest significant reductions in customer waiting time to be gained from the proposed multi-modal delivery model.

1. Introduction

Last-mile delivery is arguably one of the most complex and costly to operate parts of a supply chain. According to Goodman [1], the final delivery of goods from distribution centers to customers accounts for up to 28% of the total cost of transportation. Similarly, Jacobs et al. [2] find that last-mile delivery on average accounts for 41% of overall supply chain cost. On an operational level, the optimal planning of last-mile delivery routes constitutes a particularly challenging problem. Especially in urban areas, the planning of efficient delivery routes to a large number of fragmented customers constitutes a highly complex and computationally intensive optimization problem [3].

In light of a continued boom of e-commerce, consumers are expecting increasingly fast and responsive delivery services [2]. According to Jacobs et al. [2], fast and frequent deliveries are becoming a key differentiating factor for many retailers competing for consumer demand and loyalty. Especially highly-responsive, on-demand delivery services, such as two-hour

* Corresponding author.
E-mail addresses: moshref@mit.edu (M. Moshref-Javadi), ahmad.hemmati@uib.no (A. Hemmati), mwinkenb@mit.edu (M. Winkenbach).

https://doi.org/10.1016/j.apm.2019.11.020
0307-904X/

delivery, are shown to be a key driver of customer loyalty and market share. Accordingly, many retailers are investing in the ability to provide fast on-demand deliveries, the majority of which (60-70%) are fulfilled from physical retail stores.

Given the current speed of global urbanization, traffic loads and congestion levels are rising in cities around the world while the rapid growth of e-commerce and customer-centric delivery services require ever more fast, efficient, and reliable urban goods transportation [4,5]. Building on conventional delivery vehicles alone, urban distribution systems may not be able to compensate the increasingly negative impact of traffic and congestion on the service quality and speed that Logistics Service Providers (LSPs) are able to provide [6,7]. Recent literature suggests that a partial shift of urban freight distribution from ground-based to UAVs could help improve overall urban mobility substantially [8]. Therefore, an increasing number of LSPs is experimenting with alternative vehicle technologies to overcome the impending limitations to the efficiency of conventional delivery operations. Most of these efforts are focused on the integration of UAVs, or 'drones', in future urban e-commerce distribution systems. For instance, *DHL* [9], *UPS* [10], and *Amazon* [11] each piloted autonomous drone delivery systems for small packages from fulfillment centers to customers.

The most obvious advantage of UAVs over conventional vehicles is their ability to move flexibly in three dimensions, instead of along a discrete set of static roadways. This enables travel paths that are closer to straight-line connections and allows UAVs to circumvent traffic congestion or accidents. Accordingly, UAVs travel at more constant and higher average travel speeds than conventional vehicles. In Australia, *Alphabet* is operating a network of UAVs to deliver food and medical supplies in rural areas at travel speeds of up to 75 mph [12]. In dense urban environments, lower top speeds are to be expected. Nevertheless, it is reasonable to assume that even here, high travel speeds can be maintained more consistently by UAVs.

However, the relative competitiveness of UAVs suffers from two major limitations. First, based on today's battery technology, UAVs are still highly constrained in their maximum flight time and thus their geographic reach. In recent surveys, the available flight time for freight-bearing UAVs ranges between 30 and 40 min of travel per dispatch [13–15]. Second, their physical carrying capacity in terms of package size and weight is limited. While conventional delivery vehicles can often deliver more than 120 packages per trip [3], UAVs used for urban logistics are typically limited in their physical carrying capacity to no more than one package per dispatch. Especially the latter constraint potentially renders a purely UAV-based urban distribution system economically and operationally infeasible, as well as socially undesirable. The sheer number of daily e-commerce shipments going into a major urban area would require thousands of simultaneously operating drones. In Manhattan alone, more than 120,000 packages are delivered every single weekday [16]. Handling all these deliveries by UAV would create substantial nuisances for urban populations, such as noise and the risk of accidents, as well as massive control problems to safely coordinate these simultaneous flights. To this end, several vehicle manufacturers and logistics service providers, such as Mercedes-Benz and UPS, started to explore multi-modal delivery models combining conventional vehicle operations with UAVs in a joint, synchronized, and collaborative system. In these systems, UAVs are traveling on a conventional vehicle (e.g., a truck) serving as a mobile hub that dispatches and receives UAVs along its own delivery route [10,17]. Compared to a purely UAV-based delivery system, this approach has the potential to reduce the number of simultaneous UAV operations significantly, while increasing the physical reach of UAV-based delivery operations.

To develop an in-depth understanding of the implications of a combined conventional and aerial delivery model on delivery performance and traffic load, and to guide the future design of such systems for real-world applications, we formulate a mathematical model optimizing the combined routes of one truck and ($K \geq 1$) UAVs to serve a given set of customer requests. Our model assumes that the UAVs travel on the truck, can get dispatched to serve a single customer every time the truck stops at a customer location, and return to the truck at one of its subsequent delivery stops. UAVs are assumed to serve only customer requests that do not exceed their physical carrying capacity. All other requests need to be served by the truck. The simultaneous movements of the truck and UAVs need to be synchronized to allow for the UAVs to return to the truck at discrete locations within their allowable flight time. The truck, while serving delivery requests itself, effectively extends the service range of the UAVs beyond their original travel range.

Traditionally, route optimization aims at minimizing the distance and time traveled as well as the size of the required vehicle fleet [18]. Given the increasing consumer expectations and competitive pressure outlined above, recent literature is increasingly concerned with routing problems attempting to minimize delivery lead times [see, e.g., 19]. Minimizing customer waiting times for e-commerce deliveries is becoming an increasingly relevant optimization objective, as it is shown to have a strong influence on customer satisfaction, loyalty, and willingness to pay for delivery services [2]. In a recent study by Joerss et al. [20], almost 25 percent of consumers revealed that they were willing to pay significant price premiums for highly responsive delivery services, such as same-day or instant delivery. According to the same study, this share is likely to increase in the future, as particularly younger consumers show an even greater preference for and willingness to pay for these highly responsive delivery services. Therefore, retailers and their logistics service providers have an increasing incentive to prioritize customer waiting time over cost when designing for highly responsive delivery services.

In general, one could either minimize the maximum waiting time across all customers on a route (referred to as a *Min-Max* objective), or the average waiting time of these customers (which is equivalent to minimizing the sum of the waiting times of all customers along the route and referred to as a *Min-Sum* objective). Since the *Min-Sum* objective minimizes the waiting time of all customers, we focus on presenting our analyses for the latter objective. The *Min-Sum* objective is known to be computationally more difficult since a local change in the route of a vehicle could significantly affect the waiting time of all other subsequent customers in its route [21,22]. The interested reader can find corresponding results for the *Min-Max* objective in Appendix A.

The contributions of this paper are threefold. First, we formulate the Simultaneous Traveling Repairman Problem with Drones (STRPD) as a MILP model and prove the maximum possible savings in customer waiting times by the proposed multi-modal delivery system. Second, we propose a metaheuristic solution algorithm, TDRA, combining several heuristics which allow for the efficient solution of real-world problem size instances of the STRPD. Finally, we assess the commercial and operational benefits of the proposed multi-modal delivery systems by applying the proposed mathematical model and algorithm to several benchmark problem instances and comparing our results to those of the traditional Traveling Repairman Problem (TRP). These problem instances are either generated or adopted from the literature. We further demonstrate the real-world applicability and impact of our proposed method by applying it to a case study from the urban last-mile delivery operations of a major e-commerce platform in São Paulo, Brazil.

The remainder of this paper is structured as follows. Section 2 presents a comprehensive review of the literature on multi-modal UAV-based delivery systems. The multi-modal delivery system proposed in this work is formulated as a MILP model in Section 3. In Section 4, we present, the TDRA, our heuristic solution approach. In Section 5, we apply both the mathematical model and the proposed algorithm to a variety of problem instances for which we summarize and analyze the numerical results and compare the computational performance of both solution approaches. We conclude our work in Section 6 with a discussion of our approach and findings, and an outlook towards future research directions.

2. Literature review

The conventional Vehicle Routing Problem (VRP) and its numerous variations are well-studied problems in the literature. This class of optimization problem originates from a basic problem with a single uncapacitated truck, called the Traveling Salesman Problem [23,24], and then is extended to include multiple capacitated vehicles [25]. To account for real-world constraints in delivery operations, these problems have been extended to incorporate various constraints, such as time windows [26] and travel distance limit [27]. The TRP [28], which is also known as the Minimum Latency Problem, is a basic routing problem which determines the sequence of visits of a single vehicle to a given set of customer locations in an attempt to minimize the sum of waiting times of all customers [22]. The literature contains several studies of the TRP [see, e.g., [29–33]] and relevant extensions thereof [see, e.g., [21,34–37]]. A widely discussed example for multi-modal variations of routing problems is the Truck and Trailer Routing Problem (TTRP) [38]. The Truck and Trailer Routing Problem (TTRP) assumes a truck pulling a trailer. The trailer can be detached and left behind so that the truck can serve a sub-tour of customers without the trailer, before returning to its waiting location and resuming the route with the trailer attached. The most significant difference between the TTRP and the STRPD is that in the latter, UAVs can reconvene with the truck at locations other than their launch locations, i.e., the truck does not have to wait for the UAVs to return to the launch location. While this assumption of synchronized truck and UAV movements considerably improves the performance and flexibility of the distribution system (cf., Section 5.5), it significantly increases the computational complexity of the associated MILP model (see Section 5.4) and consequently calls for a heuristic solution approach to solve the STRPD at larger scales (see Section 4).

Despite the considerable number of works on the numerous variants of the vehicle routing problems, the literature on synchronized multi-echelon routing for UAV-based, multi-modal delivery systems is still very limited, given the novelty of this technology for applications in the logistics domain. The first study on multi-modal UAV-based delivery systems is presented by Murray and Chu [39] with the Flying Sidekick Traveling Salesman Problem (FSTSP). Their model assumes a single truck and a single UAV, serving customers jointly with the objective of minimizing the tour time. It is assumed that once the UAV is launched from the truck, the truck has to move to the next destination to receive the UAV back. The problem is formulated as a mixed integer programming model and a heuristic algorithm is proposed to solve some generated problem instances with size 10 and 20 customers. Agatz et al. [40] propose a new formulation of the FSTSP which allows the truck to wait at the launch location for the returning UAV. They further propose two heuristics and two exact algorithms to solve several problem instances. In a recent study, Ha et al. [8] formulate the Traveling Salesman Problem with Drone (TSP-D) with the objective of minimizing the total cost of truck and UAV travels and the cost of waiting of the UAV for the truck and vice versa. A Greedy Randomize Adaptive Search (GRASP) procedure is developed and used to solve the TSP-D problem instances with up to 100 customers.

Wang et al. [41] conduct several worst-case analyses of various multi-modal UAV-based delivery systems with single or multiple trucks and UAVs. Their particular focus is on the effect of the ratio of truck and UAV speeds. Savuran and Karakaya [42] consider a UAV-based delivery system with the objective of determining the truck stop locations that minimize the length of the UAV route serving all customers. They further present a Genetic Algorithm (GA) to solve the problem with a single truck and a single UAV. In a similar study, Chang and Lee [43] use a nonlinear programming model combined with K-means clustering to group customers and determine the optimal stop locations of the truck to launch the UAVs. Assuming a predefined route for the truck, Othman et al. [44] prove the NP-hardness of the multi-modal delivery problem with one UAV and one truck and propose an approximation algorithm to solve this problem. Using continuous approximation, Campbell et al. [45] develop a cost-based function which can be used to determine the optimal number of truck and UAV deliveries per route and determine the optimal number of UAVs per truck. Their analyses demonstrate that the costs of operations of UAV and truck, and the spatial density of customers play an important role in the optimal operational setup and cost of this UAV-based delivery system. Contrary to many prior works which assume that the truck can only stop at customer locations, the UAV-based delivery system presented by Carlsson and Song [46] allows the truck to launch UAVs

at any point on its route. A continuous approximation approach is used to model this system with several experiments on generated problems. For a comprehensive survey of models and civil applications of UAVs, the reader is referred to Otto et al. [47].

Considering solution approaches, a variety of methods have been developed to address the optimal design and planning of multi-modal UAV-based delivery systems. Several contributions propose novel mathematical formulations of the underlying optimization problems, including [39,40,48]. These models generally suffer from considerable computational complexity, limiting the size of the problem instances that can be solved within reasonable time to no more than 10 to 20 demand nodes (customers). Due to the inefficiency of exact methods in solving larger problem instances, the literature focuses on the development of efficient heuristics, such as the re-assignment heuristic [see, e.g., 39], GA approaches [see, e.g., 42], Dynamic Programming (DP) based heuristics [see, e.g., 40,49], and Greedy Randomize Adaptive Search (GRASP) procedures [see, e.g., 8]. Although these approaches do not prove optimality, they are computationally much more efficient than exact methods and have been used to solve UAV-based delivery problems with up to 100 customer nodes close to optimality [see, e.g., [8,50,51]]. Yurek and Ozmutlu [52] develop a two-stage iterative algorithm to solve the FSTSP. The first stage of their proposed method obtains a single truck route through partial enumeration, while the second stage determines the route of the UAV with the objective of minimizing the total waiting time of the truck. Boysen et al. [53] study the computational complexity of six variants of the TSP-D for a fixed and given truck route by varying the number of UAVs as well as the rejoin locations of the UAVs and the truck.

Another stream of research builds on the use of continuous approximation techniques to enable solutions to such problems by eliminating the explicit routing component and reducing the complexity of the overall optimization problem through closed-form estimates of the travel distances and cost of trucks and UAVs [see, e.g., 45,46,54].

Our review of the literature indicates that additional efforts are required to develop more efficient algorithms that enable the evaluation of the corresponding optimization models of multi-modal UAV-based delivery systems on real-world size problems. Also, although a few articles consider the multi-UAV systems [see, e.g., 55], they do not propose efficient algorithms to solve such problems and they only conduct worst-case analysis. The majority of the extant contributions considers systems with a single truck and a single UAV, and concentrate on solving small-scale problems which yield limited insights for the design and operation of real-world systems. These gaps shall be addressed in the remainder of this paper.

3. The simultaneous traveling repairman problem with drones

3.1. Problem definition

In this section, we define the STRPD and describe its underlying assumptions. The STRPD considers a hybrid system consisting of a single truck and multiple ($K \geq 1$) UAVs. The truck, besides serving delivery requests itself, serves as a moving hub for the UAVs, effectively extending the service range of the UAVs. The UAVs can pick up a package from the truck, take off to deliver it to a customer when the truck stops, and return to the truck at one of its subsequent delivery stops. The truck and UAVs move simultaneously, but not independently. Their movements have to be synchronized to allow for the UAVs to return to the truck at discrete locations within their allowable flight time. After the UAV dispatch, the truck cannot wait for their return at the location of the dispatch, but only at one of the subsequent customer locations. This assumption is somewhat specific to urban settings where trucks often cannot park and block traffic for a long time, but have to proceed to the next delivery destination as soon as possible.

The problem is to find the routes of the truck and the UAVs that minimize the sum of customer waiting times. The problem is called the STRPD since it extends the single-vehicle routing problem TRP by combining it with multiple UAVs. Moreover, we make the following assumptions:

(A-1) All customer demands are known before the start of operations and all customers must be served.
(A-2) UAVs may be dispatched from the truck at the depot before the truck moves.
(A-3) The truck can only stop at customer locations to launch UAVs.
(A-4) When UAVs are launched from the truck, they have to rendezvous with the truck at any of the subsequent stops along the route or at the depot.
(A-5) UAVs can serve only a single customer request per dispatch.
(A-6) UAVs can only serve requests that do not exceed their physical carrying capacity.
(A-7) Demand of each customer can be fulfilled by one visit of either the truck or a UAV.
(A-8) The flight range of the UAVs is limited by time.
(A-9) When a UAV and the truck reconvene at a customer location, the one arriving earlier waits for the other.
(A-10) UAVs are recharged at the truck instantaneously.
(A-11) Multiple UAVs can be launched simultaneously.
(A-12) The return time to the depot is not considered in the objective value since we focus on minimizing the waiting time of the customers.

Fig. 1 illustrates a potential solution to the STRPD.

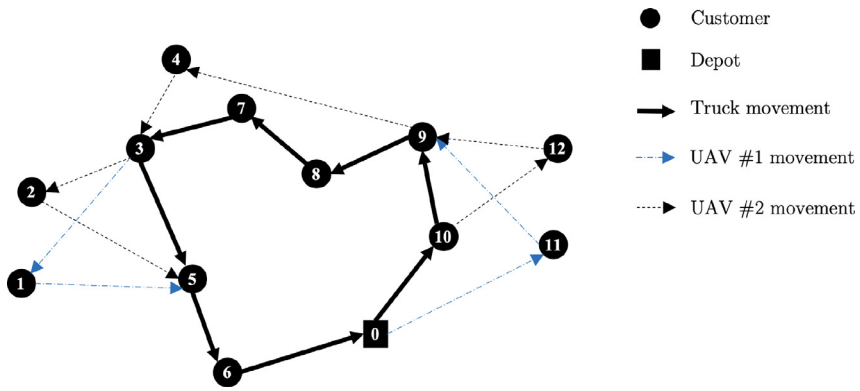

Fig. 1. The simultaneous traveling repairman problem with drones.

Table 1
Notation used in the mathematical model to the STRPD.

Parameters	
d_{ij}	Travel time between node i and j by truck
$\hat{d}_{ij}$	Travel time between node i and j by UAV
M	A sufficiently large number
L	Flight range limit of UAVs
c	Number of customers
K	Number of UAVs
Sets	
V	Set of all nodes in the network. Elements 0 and $c+1$ both represent the depot.
C	Set of customers. $C = \{1, 2, \ldots, c\}$. $\|C\| = c$.
C_0	Set of customers and the depot as the starting location. $C_0 = \{0, 1, 2, \ldots, c\}$.
C_E	Set of customers and the depot as the terminal location. $C_E = \{1, 2, \ldots, c, c+1\}$.
U	Set of UAVs. $\|U\| = K$.
Indexes	
i, j, s, g, v, h	Equivalently used indexes for the nodes of the network (customers and depot)
u	Index of UAVs
Decision Variables	
x_{ij}	Binary variable; 1 if truck moves from node i to node j; 0 otherwise.
y^u_{ijs}	Binary variable; 1 if UAV u travels from node i to visit node j and returns to the truck at node s; 0 otherwise.
p_{ij}	Binary variable; 1 if node i is visited before node j by truck; 0 otherwise.
z_i	Position of node i in the truck route
t_i	Arrival time of truck at node i
l_i	Departure time of truck from node i
$\hat{t}_{iu}$	Arrival time of UAV u at node i
$\hat{l}_{iu}$	Departure time of UAV u from node i

3.2. Mathematical model

In this section we present the mathematical optimization model characterizing the STRPD. Table 1 summarizes the notation used in the model formulation. This formulation extends the model presented by Murray and Chu [39] since it considers multiple UAVs, incorporates exact calculations of arrival and departure times of UAVs from the truck and at customers/depot locations, and follows the objective of minimizing the total waiting time of the customers. The model can be formulated as follows.

$$\textit{Minimize} \quad \sum_{i \in C} t_i + \sum_{i \in C} \sum_{u \in U} \hat{t}_{iu} \tag{1}$$

subject to

$$\sum_{j \in C_E} x_{0j} = 1, \tag{2}$$

$$\sum_{i \in C_0} x_{i,c+1} = 1, \tag{3}$$

$$\sum_{\substack{g\in C_0\\ g\neq i}} x_{gi} = \sum_{\substack{j\in C_E\\ j\neq i}} x_{ij}, \qquad \forall i \in C, \tag{4}$$

$$\sum_{\substack{i\in C_0\\ i\neq j}} x_{ij} + \sum_{\substack{i\in C_0\\ i\neq j}} \sum_{s\in C_E} \sum_{u\in U} y^u_{ijs} = 1, \qquad \forall j \in C, \tag{5}$$

$$\sum_{\substack{j\in C\\ j\neq i}} \sum_{s\in C_E} y^u_{ijs} \leq 1, \qquad \forall i \in C_0, \forall u \in U, \tag{6}$$

$$\sum_{\substack{i\in C_0\\ i\neq s}} \sum_{j\in C} y^u_{ijs} \leq 1, \quad \forall s \in C_E, \forall u \in U, \tag{7}$$

$$\sum_{h\in C_0 h\neq i} x_{hi} + \sum_{\substack{g\in C\\ g\neq s}} x_{gs} \geq 2y^u_{ijs}, \quad \forall i \in C, \forall j \in C, \forall s \in C_E, \forall u \in U, \tag{8}$$

$$y^u_{0js} \leq \sum_{\substack{g\in C_0\\ g\neq s}} x_{gs}, \qquad \forall j \in C, \forall s \in C_E, \forall u \in U, \tag{9}$$

$$l_i + d_{ij} - (1 - x_{ij})M \leq t_j, \qquad \forall i \in C_0, \forall j \in C_E, i \neq j, \tag{10}$$

$$\hat{l}_{iu} + \hat{d}_{ij} - \left(1 - \sum_{\substack{s\in C_E\\ s\neq j}} y^u_{ijs}\right)M \leq \hat{t}_{ju}, \qquad \forall i \in C_0, \forall j \in C, \forall u \in U, \tag{11}$$

$$t_i \leq l_i, \qquad \forall i \in C_0, \tag{12}$$

$$\hat{l}_{iu} - \left(1 - \sum_{\substack{j\in C\\ j\neq i}} \sum_{s\in C_E} y^u_{ijs}\right)M \leq l_i, \qquad \forall i \in C_0, \forall u \in U, \tag{13}$$

$$\hat{t}_{ju} + \hat{d}_{js} - \left(1 - \sum_{\substack{i\in C_0\\ i\neq j}} y^u_{ijs}\right)M \leq l_s, \qquad \forall j \in C, \forall s \in C_E, \forall u \in U, \tag{14}$$

$$t_i - \left(1 - \sum_{\substack{j\in C\\ j\neq i}} \sum_{s\in C_E} y^u_{ijs}\right)M \leq \hat{l}_{iu}, \qquad \forall i \in C_0, \forall u \in U, \tag{15}$$

$$\hat{t}_{ju} + \hat{d}_{js} - \left(2 - \sum_{\substack{i\in C_0\\ i\neq j}} y^u_{ijs} - \sum_{\substack{v\in C\\ v\neq s}} \sum_{h\in C_E} y^u_{svh}\right)M \leq \hat{l}_{su}, \qquad \forall s \in C_E, \forall j \in C, \forall u \in U, \tag{16}$$

$$1 - (c+2)(1 - x_{ij}) \leq z_j - z_i, \qquad \forall i \in C_0 \forall j \in C_E, i \neq j, \tag{17}$$

$$(c+2)(1 - x_{ij}) + 1 \geq z_j - z_i, \qquad \forall i \in C_0, \forall j \in C_E, i \neq j, \tag{18}$$

$$(c+2)p_{ij} \geq z_j - z_i, \qquad \forall i \in C_0, \forall j \in C_E, i \neq j, \tag{19}$$

$$p_{ij} + p_{ji} = 1, \qquad \forall i \in C, \forall j \in C, i \neq j, \tag{20}$$

$$-\left(3 - p_{ig} - y^u_{ijs} - \sum_{\substack{v \in C \\ v \neq i \\ v \neq s \\ v \neq g}} \sum_{\substack{h \in C_E \\ h \neq i \\ h \neq s}} y^u_{gvh}\right) M \leq \hat{l}_{gu} - t_s, \quad \forall i \in C_0, \forall s \in \{C_E : s \neq i\}, \forall g \in \{C : g \neq i, g \neq s\}, \forall j \in \{C : j \neq g\}, u \in U, \tag{21}$$

$$L + (1 - y^u_{ijs})M \geq t_s - \hat{l}_{iu}, \quad \forall i \in C_0, \forall j \in \{C : j \neq i\}, \quad \forall s \in \{C : s \neq i\}, u \in U, \tag{22}$$

$$y^u_{ijs}(\hat{d}_{ij} + \hat{d}_{js}) \leq L, \quad \forall i \in C_0, \forall j \in \{C : j \neq i\}, \quad \forall s \in \{C_E : s \neq i\}, u \in U, \tag{23}$$

$$x_{ij}, p_{ij} \in \{0, 1\}, \quad \forall i \in V, \forall j \in V, \tag{24}$$

$$y^u_{ijs} \in \{0, 1\}, \quad \forall i \in V, \forall j \in V, \forall s \in V, \forall u \in U, \tag{25}$$

$$1 \leq z_i \leq (c + 2), \quad \forall i \in V, \tag{26}$$

$$t_i, l_i \geq 0, \quad \forall i \in V, \tag{27}$$

$$\hat{t}_{iu}, \hat{l}_{iu} \geq 0, \quad \forall i \in V, \forall u \in U. \tag{28}$$

The objective function in Eq. (1) minimizes the sum of the waiting times of customers. Note that each customer is visited by either the truck or a UAV. The set of Constraints (2) to (9) build the routes of the truck and the UAVs and mathematically relate them. Constraint (2) ensures that the truck departs towards exactly one customer from the depot. Constraint (3) guarantees that the truck returns to the depot from exactly one customer. Constraints (4) ensure the conservation of flow in the truck tour, that is, if the truck arrives at a customer, it also has to leave the customer again. Constraints (5) represent that each customer is visited exactly once, either by the truck or a UAV. Constraints (6) reflect that when a UAV is launched, it can visit exactly one customer and returns to the truck exactly once. Similarly, Constraints (7) represent that each UAV must depart from exactly one node and visit exactly one customer before it reconvenes with the truck. Constraints (8) ensure that when a UAV is launched from node i and returns to the truck at node s, the truck route needs to contain both nodes i and s. Similarly, Constraints (9) reflect that if a UAV is launched from the depot and returns to the truck at node s, node s must be on the truck's route for it to receive the UAV.

Constraints (10) and (11) are used to calculate the arrival times of the truck and UAVs at customers, respectively. Constraints (10) ensure consistent arrival times of the truck at its customers. Since we assume that the delivery operations start at time 0, this arrival time at a customer is equivalent to the waiting time of the customer, which is minimized by the objective function. If the truck travels from node i to j, the arrival time of the truck at node j is equal to the sum of the departure time of the truck from node i and the truck travel time between nodes i and j. Note that since the objective is minimizing the sum of arrival times, the values of t_i at customers which are visited by UAVs are set to 0 by the model. To obtain a sufficiently large value for M, we solve the TRP by the Nearest Neighbor algorithm and use the objective function value of the obtained solution for M. Constraints (11) ensure consistent arrival times of the UAVs at their customers. The arrival time of UAV u which is launched from node i to visit customer j is equal to the sum of the departure time of UAV u from node i and the UAV travel time between nodes i and j. Note that the values of $\hat{t}_{iu}$ for those customers that are visited by the truck are set to 0 by the model.

Constraints (12)–(16) ensure consistent departure times of the truck and the UAVs from each customer. Constraints (12) guarantee that the departure time of the truck from any node i must be greater than its arrival time at this node. Constraints (13) ensure that the truck's departure time from any node i must be greater than the departure time of all UAVs that are launched at this node. Constraints (14) guarantee that the truck only leaves node s after all of the UAVs that are scheduled to return to the truck at node s have arrived. Constraints (15) ensure that any given UAV u cannot be launched from node i unless the truck has arrived at this node. Constraints (16) ensure that if UAV u returns to the truck at node s and is scheduled to be launched again from node s, the departure time of UAV u from node s must be greater than its arrival time at node s.

Constraints (17)–(21) guarantee the correct permutation of customers on the truck route and of the associated UAV launches. Constraints (17) and (18) ensure a consistent sequence of customer visits by the truck. Constraints (19) and (20) ensure that if the truck visits customer j after customer i, p_{ij} is set to 1. Constraints (21) relate variables p_{ij} and y^u_{ijs} by ensuring that the launch time of a given UAV u from node g, which is located after node s in the truck route, must be greater than the arrival time of the truck at node s.

Constraints (22) represent the travel range constraints for the UAVs. That is, if UAV u is launched from node i to serve node j and returns to the truck at node s, the travel time of UAV u from node i through node j to node s plus the waiting time of the UAV for the truck at node s must be less than L. These constraints activate if the truck arrives at customer s after UAV u. Similarly, Constraints (23) ensure that the travel time of UAV u from node i through node j to node s must be less than L. These constraints activate if the arrival time of UAV u at node s is greater than the arrival time of the truck (t_s) at this node, that is, if the truck waits for the UAV. Finally, Constraints (24)–(28) account for the types and admissible ranges of the variables.

Further, to account for the service times of the customers, Constraints (10) and (11) become

$$t_j \geq l_i + S_{Tj} + d_{ij} - (1 - x_{ij})M, \qquad \forall i \in C_0, \forall j \in C_E, i \neq j \qquad (10'),$$

$$\hat{t}_{ju} \geq \hat{l}_{iu} + S_{D_j} + \hat{d}_{ij} - \left(1 - \sum_{\substack{s \in C_E \\ s \neq j}} y^u_{ijs}\right)M, \qquad \forall i \in C_0, \forall j \in C, \forall u \in U \qquad (11'),$$

where S_{Tj} and S_{Dj} are the service times of customer j by the truck and the UAVs, respectively.

3.3. Bound analysis

When designing a new delivery system, it is helpful to know the maximum possible savings or improvement that could be obtained from the new system design compared to the conventional truck-only system. The goal of this section is to examine the potential savings, i.e., the maximum reduction in customer waiting time, that can be achieved by the STRPD compared to the TRP. First, we find a solution for the STRPD and then construct a TRP solution (serving all the customers with a single truck without UAVs) based on the obtained STRPD solution. Bound analysis is common in proving the maximum possible savings by algorithms and has been used in [41] and [56]. In the following theorem, we assume that the UAV to truck speed ratio is α and the problem is denoted by $STRPD_\alpha$. The Min-Sum objective is denoted as $MS(.)$. The following theorem proves the bound for the Min-Sum objective.

Theorem 1. $1 \leq \frac{MS(TRP)}{MS(STRPD_\alpha)} \leq (2c + \alpha(2c(K-1)+1), \quad \text{for} \quad K \geq 1.$

Proof. The validity of the left-side inequality is straightforward since UAVs are used only when they can contribute to reducing customer waiting times. That is, in the worst case, if the $STRPD_\alpha$ does not contain any deliveries by UAVs, the $STRPD_\alpha$ optimal solution is identical to the TRP optimal solution and $MS(STRPD_\alpha) = MS(TRP)$. To prove the right-side inequality, we first solve the $STRPD_\alpha$ and obtain K routes of UAVs and one truck route. In this $STRPD_\alpha$ solution, a_i is the waiting time of customer i, l_k is the waiting time of the last visited customer in UAV route k, l_T is the waiting time of the last visited customer in the truck route, and a_m is the largest waiting time among all routes, and $l_m = \max_k\{l_k, l_T\}$. The obtained $STRPD_\alpha$ solution contains $K = 1$ routes, including one truck route and K routes of UAVs. Now, we construct a TRP solution for the truck to visit the customers in the first UAV's route, return to the depot through the same route, then visit the customers in the second UAV's route, return to the depot through the same route, and continue to visit all the customers in all the routes of the UAVs and truck. The waiting time of customer i in UAV route k is $2l_T + \sum_{k'=1}^{k-1} 2\alpha l_{k'} + \alpha a_i$. Since $l_k \leq l_m, \forall k = \{1, \ldots, K\}$, the waiting time at customer i is $2l_T + \sum_{k'=1}^{k-1} 2\alpha l_{k'} + \alpha a_i \leq 2l_m + \sum_{k'=1}^{K-1} 2\alpha l_m + \alpha a_i \leq 2l_m + 2\alpha(K-1)l_m + \alpha a_i$. The sum of waiting times at customers in the constructed TRP solution, $MS(TRP)$, is

$$\begin{aligned} MS(TRP) &\leq \sum_{i=1}^{c}(2l_m + 2\alpha(K-1)l_m + \alpha a_i) = 2cl_m + 2c\alpha(K-1)l_m + \alpha\sum_{i=1}^{c} a_i \\ &\leq 2cl_m + 2c\alpha(K-1)\sum_{i=1}^{c} a_i + \alpha\sum_{i=1}^{c} a_i \\ &\leq (2c + \alpha(2c(K-1)+1)MS(STRPD_\alpha). \end{aligned} \qquad (29)$$

□

The following proposition proves the maximum number of customers that can potentially be served by UAVs in an STRPD solution. This measure is used in Section 5.5 to assess which fraction μ of the maximum potential usage of UAVs to serve customers has been exploited.

Proposition 1. *The maximum number of stops that can be served by UAVs, $c_{\max}$, is given by $c_{\max} = K(\frac{c+1}{K+1})$.*

Proof. Assume that the truck serves m stops and the rest of the stops are served by the UAVs. To obtain the maximum number of deliveries by UAVs, we assume that whenever the truck stops, it launches all the UAVs. These UAVs will return to the truck at the next truck stop since we do not allow the truck to launch each drone more than one time from the same location. Having m stops plus an initial stop at the depot provides $m+1$ potential flights for each UAV. Thus, all UAVs can serve $K(m+1)$ customers where K is the number of UAVs. If c indicates the number of stops, therefore, $m + K(m+1) = c$

and $m = \frac{(c-K)}{(K+1)}$. Thus, $c_{\max}$ is calculated as

$$c_{\max} = K\left(\frac{c-K}{K+1}+1\right) = K\left(\frac{c+1}{K+1}\right). \tag{30}$$

□

4. The truck and drone routing algorithm

To solve the STRPD, we develop a meta-heuristic algorithm, the TDRA. It is inspired by a well-known meta-heuristic, the Adaptive Large Neighborhood Search (ALNS) introduced by Ropke and Pisinger [57], extending the Large Neighborhood Search heuristic proposed by Shaw [58]. ALNS has been successfully applied to different optimization problems including routing problems [see, e.g., 57,59,60]. Algorithm 1 presents the pseudo-code of the TDRA. The TDRA starts by generating an

Algorithm 1: Outline of the TDRA.

Inputs: $d_{ij}, \hat{d}_{ij}, U, C_0, c, \xi, R_{max}, T_0, \beta_{TDRA}$
Generate an initial solution, s, to the STRPD based on the SA approach described in Section 4.2
$s_{best} \leftarrow s$, $f(s_{best}) \leftarrow f(s)$, $iter \leftarrow 0$, $T \leftarrow T_0$
repeat
  Select a heuristic from the list of heuristics (Section 4.3) using the selection method and heuristic weights described in (Section 4.4)
  Apply the selected heuristic to s'
  Save the new solution as s'
  if $f(s') < f(s_{best})$ **then**
    $s_{best} \leftarrow s'$
    $f(s_{best}) \leftarrow f(s')$
  end
  if s' *is accepted as the new solution based on the Boltzmann probability function with current temperature* T, **then**
    $s \leftarrow s'$
  end
  Update weights of heuristics (Section 4.4)
  Change search neighborhood using Shake heuristic after ξc non-improving iterations (Section 4.5)
  $iter \leftarrow iter + 1$
  $T \leftarrow \beta_{TDRA} T$
until $iter = R_{max}$;
return $s_{best}, f(s_{best})$.

initial solution using the Simulated Annealing (SA) method described in Section 4.2. This initial solution undergoes an improvement procedure which iteratively generates new solutions using the heuristics described in Section 4.3 and updates the best found solution. Each heuristic has a weight associated to it, which determines the likelihood of it being selected by the algorithm (see Section 4.4). These weights are periodically updated according to the Adaptive Weight Assignment method described in Section 4.4. To escape from local optima, we allow the algorithm to accept non-improving solutions in both the initial solution generation step and the improvement procedure using the *Boltzmann probability function*, $e^{\frac{-|f(s)-f(s')|}{T}}$, where T is the current iteration temperature, $f(s)$ is the objective function value of the current solution and $f(s')$ is the objective function value of the newly generated solution. We also use a Shake heuristic to change the search neighborhood if the best found solution does not improve after a certain number of iterations, as described in Section 4.5. The search procedure only covers the feasible region of the solution space since all the designed heuristics ensure feasibility. The algorithm runs for a pre-specified number of iterations, $R_{\max}$, which is tuned based on preliminary experiments. For the numerical analyses presented in Section 5, we chose $R_{\max} = 6,000$.

4.1. Solution representation

The STRPD consists of three major decisions. The first decision is the assignment of customers to the truck and the UAVs. The second decision determines the sequence of customers visited by the truck. The third decision determines the sequence of customers visited by each UAV, as well as the launch and reconvene locations of each UAV trip. These locations can be the depot or any of the customer locations that are served by the truck. Note that all of these decisions are made simultaneously due to the inter-connection among these decisions.

The solution representation we discuss in the following reflects all of the three decisions. In Fig. 2, an example solution is illustrated along with its solution representation which is composed of four sections:

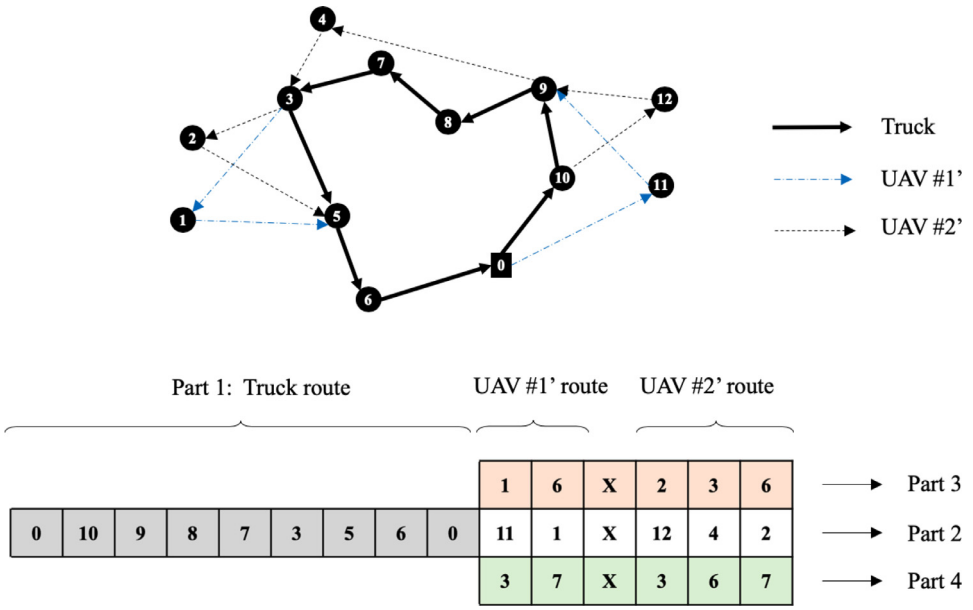

Fig. 2. Solution representation of the TDRA.

- Part 1 represents the sequence of customers visited by the truck.
- Part 2 denotes the assignment of customers to the UAVs and also the sequence of visits by each UAV.
- Parts 3 and 4 represent the launch and reconvene locations of each UAV for every flight, respectively. Note that the numbers in Parts 3 and 4 refer to the cell number in Part 1, not the customer number. For instance, number 2 in Part 3 means that the UAV is launched from the truck at the location of the customer that appears in the second cell of Part 1, which is customer 10. Likewise, number 3 in Part 4 means that the UAV is launched from the truck at the location of the customer that appears in the third cell of Part 1, which is customer 9.

Given c customers and K UAVs, the first two parts have a fixed length of $c + K + 1$. Part 1 starts and ends with a zero, which denotes the depot. The numbers between the two zero values represent the sequence of customers served by the truck. Part 2 consists of the customers that are served by UAVs. These customers are grouped by $K - 1$ separators (denoted by "X" in Fig. 2), generating a total of K sections, one for each UAV. Each section denotes the sequence of customers that are served by a specific UAV. In Fig. 2, customers 11 and 1 are served by one UAV and customers 12, 4, and 2 are served by the second UAV. The length of Parts 3 and 4 is the same as the length of Part 2.

4.2. Initial solution

The proposed TDRA begins with an initial feasible solution. To construct an initial solution, we randomly assign all customers to the truck route (Part 1 of the solution representation). Then, we use the SA algorithm shown in Algorithm 2, to improve the generated solution. The SA starts with an initial temperature, T_0, which decreases in each iteration according to a cooling schedule, $T_{new} = \beta_{SA} * T_{old}$ [61], and terminates once the temperature reaches the final temperature, T_f. For our numerical analyses, we choose $\beta_{SA} = 0.99$. The temperature is used in the *Boltzmann probability function*, $e^{\frac{-|f(s)-f(s')|}{T}}$, to accept non-improving solutions probabilistically, where T is the current temperature. Starting from an initial solution, the SA iteratively generates $I_{\max}$ new solutions at each temperature and updates the best found solution in each iteration. The function *neighbor*(s) finds a neighboring solution, s', to the current solution, s, using the well-known 2-Opt and 3-Opt operators [see, e.g., 62]. Fig. 3 exemplifies an output of the SA algorithm.

The output of Algorithm 2 is then used in the *Elliptical Customer Assignment (ECA)* heuristic to transform the obtained TRP solution to an initial STRPD solution. The ECA heuristic is used to remove some of the customers from the pure truck route obtained from Algorithm 2 and assign them to the UAVs. In this heuristic, as shown in Algorithm 3 , we first use the Least-Squares criterion to estimate the best ellipsoid fit to the given set of customers and depot locations [63]. Then, we find the shortest distance from each customer to the fitted ellipse, as illustrated in Fig. 4. To assign customers to the UAVs, we first make a *List* of all customers in ascending order of their perpendicular distance to the fitted ellipse. Then, we attempt to remove the customers from the truck route and assign them to the UAV routes. To do so, we select customers one by one from top of the created *List* and check all potential positions to which the customer can be assigned. These positions include the positions in the truck route, as well as the positions in all potential UAV routes, considering all possible combinations of launch and reconvention locations. The removed customer is assigned to the feasible position that is the most beneficial

Algorithm 2: SA heuristic to generate an initial solution to the TRP.

Inputs: $d_{ij}, C, T_0, T_f, \beta_{SA}, I_{max}$, Cooling_Schedule
Generate a random TRP solution s
$s_{best} \leftarrow s, f(s_{best}) \leftarrow f(s), T \leftarrow T_0$
while $T > T_f$ **do**
  $s' \leftarrow s$
  for $i = 1$ *to* I_{max} **do**
    $s' = neighbor(s)$
    if $f(s') < f(s_{best})$ **then**
      $s_{best} \leftarrow s'$
      $f(s_{best}) \leftarrow f(s')$
    end
    if s' *is accepted as the new solution based on Boltzmann probability function with temperature* T **then**
      $s \leftarrow s'$
    end
  end
  $T \leftarrow \beta_{SA} T$ (cooling schedule)
end
return s_{best}

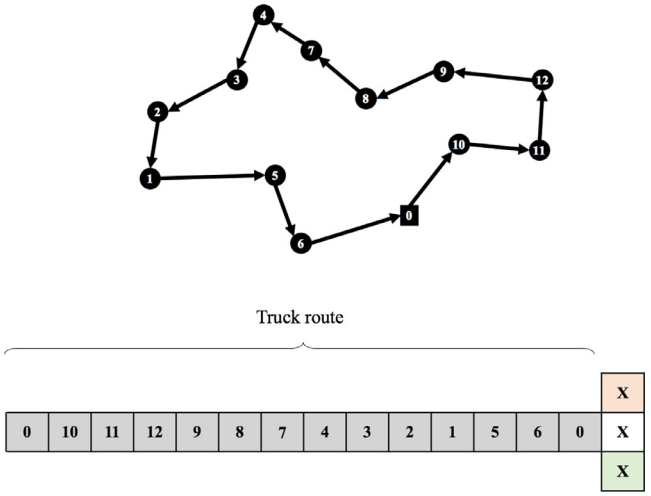

Fig. 3. A sample output of the Simulated Annealing algorithm.

in terms of reducing customer waiting time. This new solution is saved as the best found solution. Note that to ensure the feasibility of the new assignment, we need to check two constraints. First, the assignment needs to satisfy the flight range limit of the UAV. Second, the origin and destination of a new UAV trip need to be selected without any conflicts with already scheduled UAV trips. That is, a UAV can be scheduled for a flight if it is available in the truck and it has to rejoin the truck before an already scheduled UAV flight from the truck.

If a customer cannot be re-inserted at any of the potential positions because it would only create infeasible or non-improving feasible solutions, the customer remains at its original position. For customers that can not be assigned, we repeat these evaluation and insertion procedures iteratively as new positions may open up after each round of assignment. This heuristic terminates when there is no position that customers can be assigned to, and that improves the best found solution while maintaining feasibility. A sample output of the ECA heuristic is shown in Fig. 5.

4.3. Modification heuristics

In this section, we describe the various heuristics that we developed to use as part of the TDRA to solve the STRPD.

4.3.1. 2-Opt Heuristic

This heuristic targets the first two parts of the solution representation by considering them as a single vector and implementing the well-known 2-Opt operator on this vector [62]. The 2-Opt heuristic swaps two values in the vector. In our case,

Algorithm 3: The Elliptical Customer Assignment (ECA) heuristic.

Inputs: $s_{in} \leftarrow$ TRP solution by Algorithm 2, C_0
$s_{out} \leftarrow s_{in}, f(s_{out}) \leftarrow f(s_{in})$
Fit an ellipse to C_0 according to [63]
Make a *List* of all customers, sorted in descending order of their distance from the ellipse
repeat
  for $i = 1$ *to* $|List|$ **do**
    $j \leftarrow$ index of i^{th} customer in the *List*
    Select the customer c_j in the *List* and remove it from the truck route
    for *all potential positions in all UAV and truck routes in* s_{out} *to insert customer* c_j **do**
      Re-insert customer c_j in the current position and save the new solution as s'
      if (s' *is feasible) and* ($f(s') < f(s_{out})$) **then**
        $s_{out} \leftarrow s'$
        $f(s_{out}) \leftarrow f(s')$
      end
    end
  end
  Remove the assigned customers from the *List*
until *no feasible position is available for customers in the List for re-insertion in UAV routes*;
return s_{out}

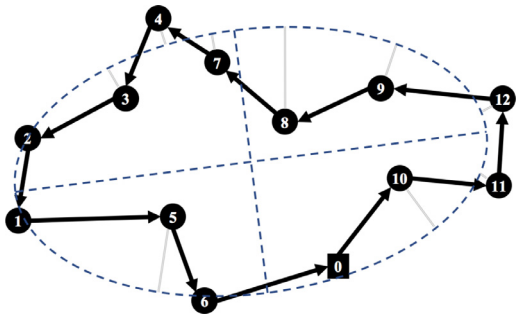

Fig. 4. Fitting an ellipse to the customers and calculating the distance of customers to the fitted ellipse.

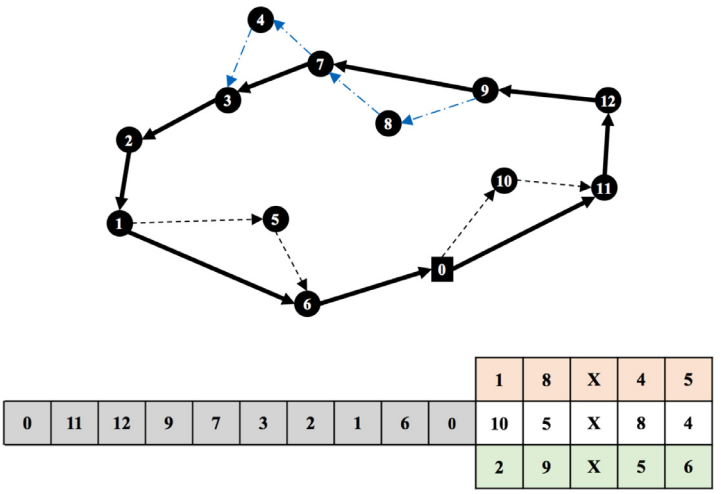

Fig. 5. A sample output of the Elliptical Customer Assignment heuristic.

we switch two customers within the vector of Parts 1 and 2. The customers are selected randomly for the 2-Opt method. Changing the values in Parts 1 and 2 while keeping the values in Parts 3 and 4 unchanged may lead to an infeasible solution due to the flight range constraint. In case of infeasibility, we use the Drone Planner heuristic (Algorithm 7 in Section 4.3.6) to find a potentially better and feasible assignment of customers to UAVs. If the 2-Opt heuristic does not find a feasible solution, it returns the input solution unchanged.

4.3.2. 3-Opt Heuristic

This heuristic is similar to the 2-Opt heuristic with the difference that instead of two customers, three customers are randomly selected from Parts 1 and 2 and exchanged [62]. This heuristic also uses the Drone Planner heuristic (Algorithm 7 in Section 4.3.6) to find a potentially better and feasible assignment of customers to UAVs. If the heuristic cannot find an improving and feasible solution considering the flight range constraint and the sequence of all UAV launch and reconvention locations, it returns the input solution unchanged.

4.3.3. Greedy assignment heuristic

This heuristic, i.e., Algorithm 4 targets Parts 1 and 2 of the solution representation to exchange customers between the truck and UAV routes. However, it excludes customers that are used as rendezvous locations since we consider other operators which particularly target them in the improvement procedure. The Greedy Assignment heuristic randomly selects one of the aforementioned customers from the truck or UAV routes and re-assigns it to its most beneficial position, which can be either in the truck or a UAV route. To obtain a list of all potential positions, we consider all the positions in the truck route, and all the positions in all the UAV routes considering all the feasible combinations of UAV launch and reconvention locations. The feasibility of a position takes into account the flight range limit of the UAVs and the correct sequence of origin and destination of each UAV flight. If the heuristic cannot find a new feasible improving solution after re-inserting the selected customer into all potential positions, it returns the input solution unchanged. The pseudo-code of the Greedy Assignment heuristic is shown in Algorithm 4 .

Algorithm 4: Greedy Assignment Heuristic.

Inputs: $s_{in}, f(s_{in})$
$s_{best} \leftarrow s_{in}, f_(s_{best}) \leftarrow f_(s_{in})$
Create a *List* of all customers served by truck or UAV, but not served at rendezvous locations
Select a customer i randomly from the *List* and remove it from its current route
for *all potential positions in all UAV and truck routes to insert customer* i **do**
    Re-insert customer i in the position and set the new solution as s'
    if (s' *is feasible) and* ($f(s') < f(s_{best})$) **then**
        $s_{best} \leftarrow s'$
        $f(s_{best}) \leftarrow f(s')$
    end
end
return $s_{best}, f(s_{best})$

4.3.4. Origin-destination relocation heuristic

This heuristic complements the Greedy Assignment heuristic by considering customers that are either an origin and/or destination of at least one UAV trip. First, we create a *List* of these customers. Then, we randomly select one customer from the *List* and attempt to relocate it to another position in the truck route. To relocate it, we check the re-insertion of the customer to all the possible positions in Part 1 of the solution and choose the most beneficial, feasible position. Therefore, unlike the Greedy Assignment heuristic, there is no customer exchange between the truck and UAVs. To ensure feasibility, since the UAV flights may be affected by this relocation, we remove all the UAV flights from the solution and use the Drone Planner heuristic (see Section 4.3.6) to find a feasible flight plan. If the heuristic cannot find a feasible solution in any of the available re-insertion positions, it returns the input solution of this heuristic unchanged. The pseudo-code of this heuristic is shown in Algorithm 5.

4.3.5. General assignment heuristic

This heuristic operates similar to the Greedy Assignment heuristic and targets the customers in Parts 1 and 2 of the solution representation. However, instead of removing a single customer, it randomly selects and removes at most five customers at the same time and re-inserts them one by one to the truck or UAV routes. Therefore, starting from the first customer, we insert it to the position that yields the best objective value. This procedure continues until all of the removed customers are re-inserted. If the heuristic cannot find a feasible solution that contains all the removed customers, it returns the input solution to the heuristic unchanged. The pseudo-code of this heuristic is shown in Algorithm 6.

Algorithm 5: Origin-Destination Relocation Heuristic.

Inputs: $s_{in}, f_{(s_{in})}$
$s_{best} \leftarrow s_{in}, f_{(s_{best})} \leftarrow f_{(s_{in})}$
Create a *List* of all customers served at rendezvous locations between truck and UAV
Select a customer i randomly from the *List* and remove it from its current route
for *all potential positions in truck route to insert customer* i **do**
  Insert customer i to the position and save the new solution as s'
  Apply *Drone Planner* heuristic (Section 4.3.6)
  if *(*s' *is feasible) and (*$f(s') < f(s_{best})$*)* **then**
    $s_{best} \leftarrow s'$
    $f(s_{best}) \leftarrow f(s')$
  end
end
return $s_{best}, f(s_{best})$

Algorithm 6: General Assignment Heuristic.

Inputs: $s_{in}, f(s_{in})$
$s_{best} \leftarrow s_{in}, f(s_{best}) \leftarrow f(s_{in})$
Create a *List* of all customers served by truck or UAV, but not served at rendezvous locations
Select a random integer number, $N \in [2, 5]$
Randomly select N customers from the *List* and remove them from their current routes
for *selected customer* $i = 1$ *to* N **do**
  for *all potential positions in all UAV and truck routes to insert customer* i **do**
    Insert customer i to the position and save the new solution as s'
    if *(*s' *is feasible) and (*$f(s') < f_{(s_{best})}$*)* **then**
      $s_{best} \leftarrow s'$
      $f(s_{best}) \leftarrow f(s')$
    end
  end
end
return $s_{best}, f_{(s_{best})}$

4.3.6. Drone planner heuristic

This heuristic re-assigns UAV customers to other UAVs by targeting Parts 3 and 4 of the solution representation to find a feasible solution with potentially lower customer waiting time. Note that since we do not alter Part 2 of the solution representation, the assignment of customers to UAVs remains unchanged. First, we remove all the UAV trips from the solution. Then, to select the best launch and reconvention locations for the UAV trips to serve each customer j, we make a list, P_j, of all potential combinations of these locations, (i, s), that satisfy the UAV flight range constraint. Then, we remove all combinations that contradict the sequence of customers in the truck route. Once we have all the potential combinations, the heuristic chooses the first customer j in the first UAV route to reschedule. To this end, it randomly picks one combination of launch and reconvention locations from P_j. Once these locations are assigned to customer j, list P_j is updated by removing infeasible combinations from the list. Then, the second customer served by the first UAV is chosen and the same process applies. Once all the trips by the first UAV are scheduled, we repeat the same process for all UAVs. Since we may obtain infeasible solutions in this process because combinations in P_j are limited at the end of each iteration, the heuristic repeats this process ν times to obtain a feasible solution. This number of iterations was chosen based on preliminary experiments. Once ν runs are complete, we choose the feasible solution that yields the best objective function value. If the heuristic does not obtain a feasible solution after 10 iterations, it returns the input solution unchanged. The pseudo-code of this heuristic is shown in Algorithm 7.

4.3.7. Wild change heuristic

This heuristic changes the assignments of customers to the truck and UAVs and also the sequence of deliveries by them. To do this, we randomly select at most five customers from Parts 1 and 2 of the solution representation and re-insert them concurrently in N randomly selected positions in Parts 1 and 2. Due to the flight range constraint, the new solution may be infeasible. Thus, we apply the Drone Planner heuristic (see Section 4.3.6) to find a feasible set of flights. If the heuristic cannot find a feasible solution, it returns the input solution unchanged. The pseudo-code of this heuristic is shown in Algorithm 8.

Algorithm 7: Drone Planner Heuristic.

Inputs: s_{in}, C_0, U, L, $\hat{d}_{ij} s_{best} \leftarrow s_{in}$, $f(s_{best}) \leftarrow f(s_{in})$
$O_u \leftarrow$ the number of customers served by UAV u
Remove all UAV flights from s_{in}
$P_j \leftarrow \left\{ (i,s) \mid (i,s \in V) \wedge (\hat{d}_{ij} + \hat{d}_{js} \leq L) \wedge (i \text{ and } s \text{ are served by truck}) \wedge (s \text{ is served after } i) \right\} \forall s \in C$
for *iter = 1 to 10* **do**
  for *u=1 to |U|* **do**
    for *ind=1 to* O_u **do**
      Select the customer with index *ind* in the route of UAV u, (C_{ind})
      Select a flight combination (i,s) randomly from $P_{C_{ind}}$ and schedule the trip of UAV u
      Update P_j according to the previously assigned flights to UAV_u
    end
  end
  Save the new solution as s'
  if *(s' is feasible) and* $(f(s') < f(s_{best})$ **then**
    $s_{best} \leftarrow s'$
    $f(s_{best}) \leftarrow f(s')$
  end
end
return $s_{best}, f(s_{best})$

Algorithm 8: Wild Change Heuristic.

Inputs: s_{in}
$s_{out} \leftarrow s_{in}$
Generate a random integer number $N \in [2, 5]$
Randomly select N customers and remove them from their current routes
Re-insert the selected customers to random positions in the routes of truck and UAV
Save the new solution as s'
Apply *Drone Planner* heuristic (see Algorithm 7)
if *s' is feasible* **then**
  $s_{out} \leftarrow s'$
end
return s_{out}

4.4. Adaptive weight adjustment and heuristic selection

According to Algorithm 1, in each iteration of the TDRA improvement phase, a heuristic is selected from the described heuristics in Section 4.3 and applied to the current solution. Each of the described heuristics has a weight which serves as the indicator of the performance of the heuristic. All the heuristics are given equal weights in the first iteration, but these weights are periodically adjusted during the run of the algorithm. The idea of adaptive weight adjustment, inspired by Ropke and Pisinger [57], is to adjust the weight of each heuristic in an iterative process based on statistics collected from earlier iterations. To this end, every κ iterations, so-called a segment of size κ, we collect the performance scores of each heuristic. At the end of each segment, we update the weights of the heuristics based on their performance scores. The scores are defined based on the quality of the new solution obtained by the heuristics. If a heuristic finds a solution with less total waiting time compared to the current solution, it receives one score. If the algorithm improves the best found solution, it receives two scores.

To update the weights of the heuristics at the end of each segment, we first normalize the heuristic scores, (π_q), by dividing the scores by the number of times each heuristic was used, (θ_q). The new weights are then calculated as

$$w_{q,l+1} = w_{q,l}(1-\gamma) + \gamma \frac{\pi_q}{\theta_q}, \tag{31}$$

where $w_{q,\,l}$ is the weight of heuristic q used in segment l and $\gamma \in [0, 1]$ is a coefficient used to balance between the value of earlier weights and the new normalized scores.

The TDRA selects a heuristic from the list of designed heuristics using the *roulette wheel selection* principle. Once each of the heuristics received an updated weight, $w_{q,\,l}$, in the beginning of segment l, the roulette wheel method selects a heuristic

with probability $w_{q,l}/(\Sigma_q w_{q,l})$. Therefore, those heuristics that obtain higher quality solutions, receive a higher probability to be selected again in the next segment.

4.5. Shake heuristic

The Shake heuristic is a strategy to change the search neighborhood to escape from local optima. After $c\xi$ non-improving iterations, we run an extra set of η iterations, where c is the number of customers and ξ is a coefficient. We accept the generated solutions from each of these η iterations, even if they are worse than the previous best found solution. The Shake heuristic is not a complete restart since the new solution is η steps (iterations) away from the current solution. The Shake heuristic uses the Wild Change and General Assignment heuristics since they can contribute the most diversification using their operators inside them, compared to other heuristics.

5. Computational results

This section presents the results of the application of the proposed mathematical model and algorithm to a set of generated problem instances in addition to several problem instances adopted from the literature, as well as real-world instances. All computations are performed on a 64-bit Windows operating system with dual processor, 2.8 GHz CPU, 20 GB RAM. We implemented the MILP model in Python 2.7 [64] using Gurobi 7.5 as a solver [65]. The TDRA algorithm was implemented in C++.

5.1. Problem sets

We use four sets of problem instances to evaluate the performance of the proposed mathematical model and algorithm:

Set 1 includes 70 randomly generated problem instances consisting of 4 to 10 customers in a 100×100 area with the depot located at (0,0) and 10 instances for each problem size. The purpose of this set is to examine the computation time of our MILP and to find the largest problem size that can be solved to optimality using Gurobi Optimizer. The Baseline parameter values are set to $K = 2$ UAVs, a UAV to truck speed ratio of $\alpha = 1.5$, a flight range limit given by a coverage percentage of $\mathcal{P} = 25\%$ (cf., Section 5.2), and truck and UAV service times of $S_{D_j} = S_{T_j} = 0$. Both the truck path and the UAVs flight path satisfy Euclidean geometry. We examine the sensitivity of computation time to changes in the number of UAVs, flight range limit, and speed ratio. Including the sensitivity analysis, a total of 700 experiments are run on this set with the mathematical model.

Set 2 consists of 30 problem instances with 10, 15, 20 customers in a 100x100 area with the depot located at (0,0) and 10 instances for each problem size. We use these instances to compare the solution quality and computation time performance of the proposed TDRA with the MILP on small and medium-sized problem instances. The baseline parameter values are chosen identical to Set 1. We analyze the sensitivity of the results to changes in the number of UAVs, flight range limit, and speed ratio, given the Min-Sum objective function. Including the sensitivity analysis, a total of 180 experiments are run on this set with the mathematical model. The TDRA is run 10 times on each problem instance, resulting in a total of 1800 experiments on this set.

Set 3 includes 20 adopted problem instances from Salehipour et al. [66] consisting of 10, 20, 50, and 100 customers in a 100x100 area with random depot location and 5 instances for each problem size. The purpose of this set is to compare the STRPD with the TRP to examine the potential savings in customer waiting time that can be obtained by the proposed system design. We chose Salehipour et al. [66] problem set because the TRP solutions are available. We base our analyses on the best-known TRP solutions reported in the literature by Silva et al. [67]. It is important to note that Silva et al. [67] round down all travel times to the nearest smaller integer value for their analyses. To enable a fair comparison of our results with the benchmark instances of Silva et al. [67], we apply our proposed algorithm on the same rounded down travel times. The baseline parameter values are chosen identical to Set 1. We analyze the sensitivity of our results to changes in the number of UAVs, flight range limit, and speed ratio, given a Min-Sum objective function. The TDRA is run 10 times on each problem instance, resulting in a total of 1200 experiments on this set.

Set 4 is used to illustrate the application of the proposed multi-modal UAV-based delivery models to real-world problem instances from the last-mile delivery operations of a major e-commerce platform from a case study in São Paulo, Brazil. This set consists of 12 real-world problem instances with the number of customers ranging from 28 to 101 in two 1-square kilometer areas (called $S1$ and $S2$) in downtown São Paulo. Area $S1$ includes instances from 28 to 69 customers while area $S2$ contains 56 to 101 customers. Packages are delivered from a distribution center located approximately 5.3 km and 6.3 km away from $S1$ and $S2$, respectively. While the UAVs travel in Euclidean space, the truck uses the real-world road network. Real distances are calculated using Google Distance Matrix (GDM) service [68]. For baseline parameter values, we use $K = 2$ UAVs, a 25 min flight time limit for UAVs, a truck speed of 10 km/h (based on real data), a UAV to truck speed ratio of $\alpha = 1.5$, $S_{T_j} = 180$ s of service time for the truck (based on real data), and $S_{D_j} = 60$ s of service time for UAVs. The service time of the truck accounts for parking, unloading the package from the truck, and delivery times, whereas the service time of the UAVs accounts for landing, unloading, and take-off. We examine the effect of changes in the number of UAVs, speed ratio, and service time of UAVs on the objective function value. Finally, we examine the effect of confining UAVs to traveling

on top of the real road network on the objective function value. The TDRA is run 10 times on each problem instance, resulting in a total of 960 experiments on this set.

It is worth mentioning that in our sensitivity analyses, we change one parameter value at a time while keeping the rest of the parameters at their baseline values. In Sets 1–4, we consider sensitivity analyses based on the number of UAVs ($K \in \{1, 2, 3\}$), speed ratio ($\alpha \in \{1.5, 1.0, 0.75\}$), and coverage percentage ($\mathcal{P} \in \{25\%, 75\%\}$). In Set 4, also, we further analyze the effect of changes in UAV service time ($S_{D_j} \in \{60, 90, 120\}$), and of assuming real road network distances for UAVs rather than the Euclidean distance metric.

5.2. Calculation of flight range limit

Since the problems in sets 1–3 are generated in a generic 100x100 area, we need to define the flight range (time) limit in these problems relative to the size of this area. To this end, we define a so-called coverage percentage, $\mathcal{P} \in [0\%, 100\%]$, to determine the flight time limit, L. The procedure outlined in Algorithm 9 can be used to determine the flight time limit. In

Algorithm 9: Calculation of flight range limit.

1 **Inputs:** Coverage percentage ($\mathcal{P}$), $L = \epsilon$, V
2 **repeat**
3 Calculate the total number of feasible flights, $\sum_{j\in C}\sum_{\substack{i\in V\\ i\neq j}}\sum_{\substack{s\in V\\ s\neq j}}\psi_{ijs}$.Set $L = L + \epsilon$.
4 **until** *percentage of feasible flights,* $\frac{\sum_{j\in C}\sum_{\substack{i\in V\\ i\neq j}}\sum_{\substack{s\in V\\ s\neq j}}\psi_{ijs}}{c^2(c+1)} \geq \mathcal{P}$;

this algorithm, ϵ is a small increment. In our numerical analyses, we use $\epsilon = 1$. The feasibility of a flight (ψ_{ijs}) is given by

$$\psi_{ijs} = \begin{cases} 1 & \text{if } \hat{d}_{ij} + \hat{d}_{js} \leq L, \\ 0 & \text{otherwise,} \end{cases} \tag{32}$$

where a value of 1 indicates feasibility and a value of 0 indicates infeasibility. Given a network of customer locations, L, and $\mathcal{P}$, we start from a low value for $L = \epsilon$, and calculate the percentage of feasible trips. If this value is less than $\mathcal{P}$, we increase L and recalculate the percentage of feasible flights. We follow this procedure until the percentage of feasible flights is greater than $\mathcal{P}$.

5.3. Parameter tuning

To calculate the parameter values of the TDRA, we conduct a set of full-factorial experiments. To select the parameters in the full-factorial design experiment, we conducted some preliminary experiments and chose the parameters with the highest effect on the performance of the algorithm. Among all the parameters, four of them have been selected as the most sensitive parameters to be tuned: the initial temperature of the SA procedure described in Section 4.2, T_0; the balancing parameter γ in Section 4.4 used to adjust the weights of the heuristics; the parameter ξ of the Shake heuristic discussed in Section 4.5, and the parameter η in the Shake heuristic. The selected levels for each parameter are $T_0 \in \{3000, 4000, 5000\}$, $\gamma \in \{0.1, 0.2, 0.3\}$, $\xi \in \{20, 25, 30\}$, and $\eta \in \{30, 40, 50\}$. We use a problem with 20 customers and 2 UAVs for our full-factorial design experiments. The TDRA is run 10 times for each set of parameter values, and based on a total number of 810 runs, the parameter values are set to $T_0 = 5000, \gamma = 0.2, \xi = 20, \eta = 50$. The rest of the parameter values are selected through some brief experiments due to minor sensitiviety of the algorithm to these parameters: $\nu = 10$, the number of times that the Drone-Planner heuristic is applied to find a feasible solution in Section 4.3.6, and $\kappa = 50$, the number of iterations in each segment of the weight adjustment procedure in Section 4.4.

We also conduct a full-factorial design of experiments to evaluate the performance of the heuristics described in Section 4.3. Since the Drone Planner heuristic is used inside other heuristics, this heuristic is always included in the combinations, resulting in 2^6 combinations of heuristics in our full-factorial design. We run the TDRA on 30 problem instances of sizes 10, 20, and 50, with 10 instances for each size. After running the TDRA 10 times on each problem instance, our results show that all of the heuristics positively contribute to the TDRA for finding better quality solutions. That is, removing either heuristic from the TDRA aggravates the performance of the TDRA. To examine the effect of the designed components, we removed the proposed heuristics and strategies from the TDRA and compared the performance of the modified algorithm with our proposed TDRA (Table 2). The considered components are (a) Remove Greedy Assignment heuristic, (b) Remove Origin-Destination Relocation heuristic, (c) Remove General Assignment heuristic, (d) Remove Wild change heuristic, (e) Consider (a-d) together, and (f) Remove Adaptive Weight Adjustment. The results show that some of the heuristic are more effective in finding better solutions, including (c) General Assignment heuristic, and (e) combination of operators. It is also noticed that the effect of these components is minor in smaller problems compared to the larger problems. The effects can be as high as 16.57% in case (e). Nonetheless, the Adaptive Weight Adjustment and heuristic selection (see Section 4.4) periodically updates the weights of the heuristics and selects the most productive heuristics to use in the TDRA based on their performances.

Table 2
Average percentage decrease in the obtained customer waiting time savings after removing the TDRA components. The algorithm is tested to solve problem instances in Set 3 for the Baseline scenario.

Problem size	(a)	(b)	(c)	(d)	(e)	(f)
10	0.09	0.27	0.33	0.34	0.35	0.26
20	5.51	4.47	8.04	4.21	9.99	4.67
50	6.1	5.74	12.17	5.16	16.57	5.27
100	1.36	2.51	12.44	2.04	13.72	3.03

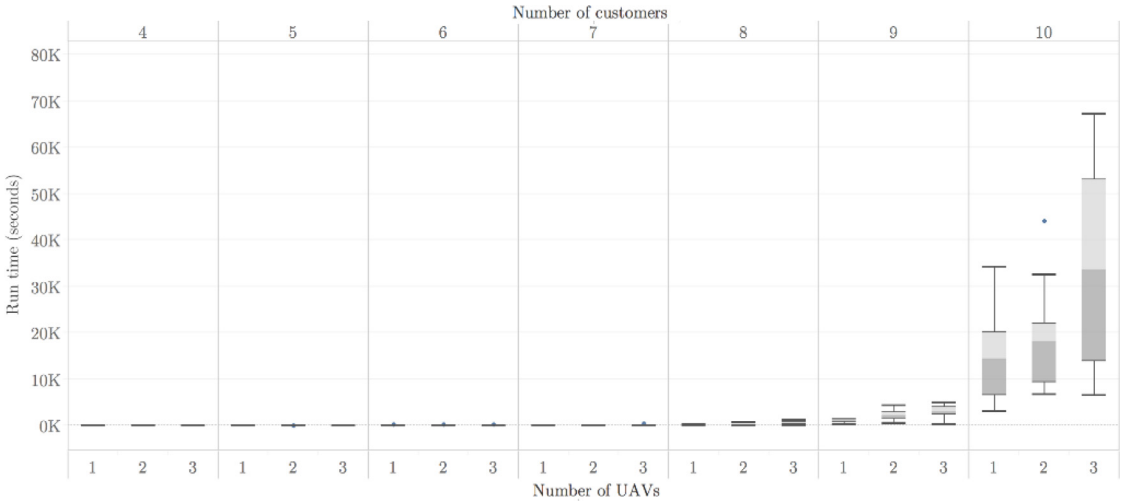

Fig. 6. Effect of the number of UAVs on MILP CPU time using Gurobi solver. Parameter values are $\alpha = 1.5$, $\mathcal{P} = 25\%$, and $K \in \{1, 2, 3\}$. Each box-plot is obtained based on 10 values.

5.4. Numerical analyses on set 1: MILP computation time

We use box and whisker plot in Fig. 6 to illustrate the computation time of the MILP with respect to the number of UAVs. The mathematical model is used to solve the 70 generated problems in Set 1 with baseline parameter values to optimality. As expected, the computation time increases in the number of UAVs. The increase in computation time from 2 to 3 UAVs is particularly pronounced. It is also noteworthy that while the computation time of problem instances with up to 8 customers is negligible, it increases drastically for problem instances with 9 or more customers. This result demonstrates that the mathematical model cannot be used to solve problems with more than 8 customers in reasonable time.

We also used the TDRA algorithm to solve the Set 1 problem instances. The algorithm could obtain the optimal solutions for all problem instances in this set. Therefore, we are confident that the proposed algorithm can obtain optimal solutions in small-scale problems.

5.5. Numerical analyses on set 2: Solution quality of the TDRA

This set is used to compare the performances of the TDRA algorithm and the MILP. The maximum run-time for the MILP is set to 10,000 s. The summary of results is illustrated in Fig. 7 with detailed results given in Appendix B where we report the best found solutions along with run-time of the MILP. For the TDRA, we report the absolute best and the average of the best found solutions over 10 runs, as well as the average run time. The relative percentage gap between the TDRA solution and MILP solution is denoted by $\Delta\% = \frac{(MS(MILP)) - MS(TDRA)}{MS(MILP)} * 100$.

The results in Fig. 7 show that $\Delta\%$ is 0 in 10-customer problems for all scenarios, indicating that the algorithm is able to find optimal or near-optimal solutions for problems with 10 customers. As the number of customers increases, savings in customer waiting times, $\Delta\%$, also increases due to higher computational complexity of the problem and the enforced 10,000 second run-time limit for the MILP. These gaps indicate the superiority of the TDRA compared to the MILP for solving problems with more than 10 customers with respect to quality and run-time. For the scenario with a coverage percentage of $\mathcal{P} = 75\%$, the MILP could not find any feasible solution in 10,000 s for problems with 15 or 20 customers, while the TDRA solves them in approximately 41 and 62 s, respectively, implying a high efficiency of the TDRA compared to the MILP. Across all experiments, for only two of the problem instances the TDRA obtains a worse solution than the MILP. However, for these instances, $\Delta\%$ is less than 1%, indicating a negligible gap and promising performance of the TDRA.

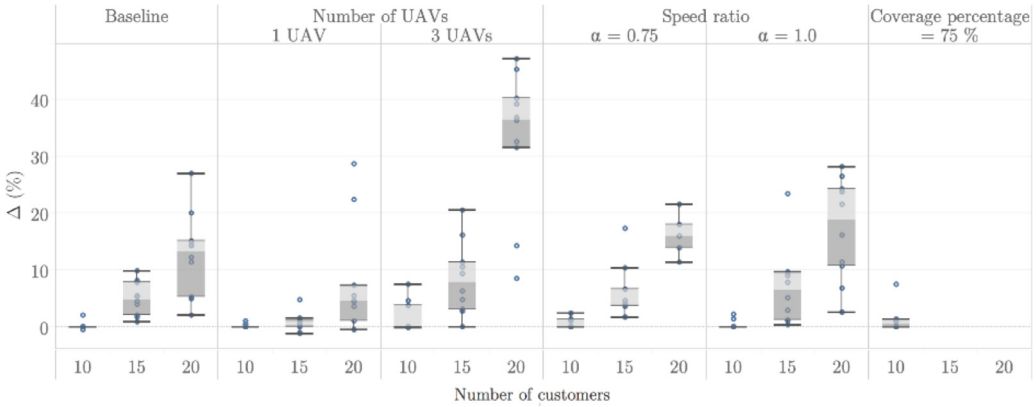

Fig. 7. Summary of comparison of objective values obtained from TDRA and MILP model on Set 2. Baseline parameter values are $K = 2$, $\alpha = 1.5$, $\mathcal{P} = 25\%$. For each scenario, the chosen parameter value is changed. Each box-plot is obtained based on 10 values.

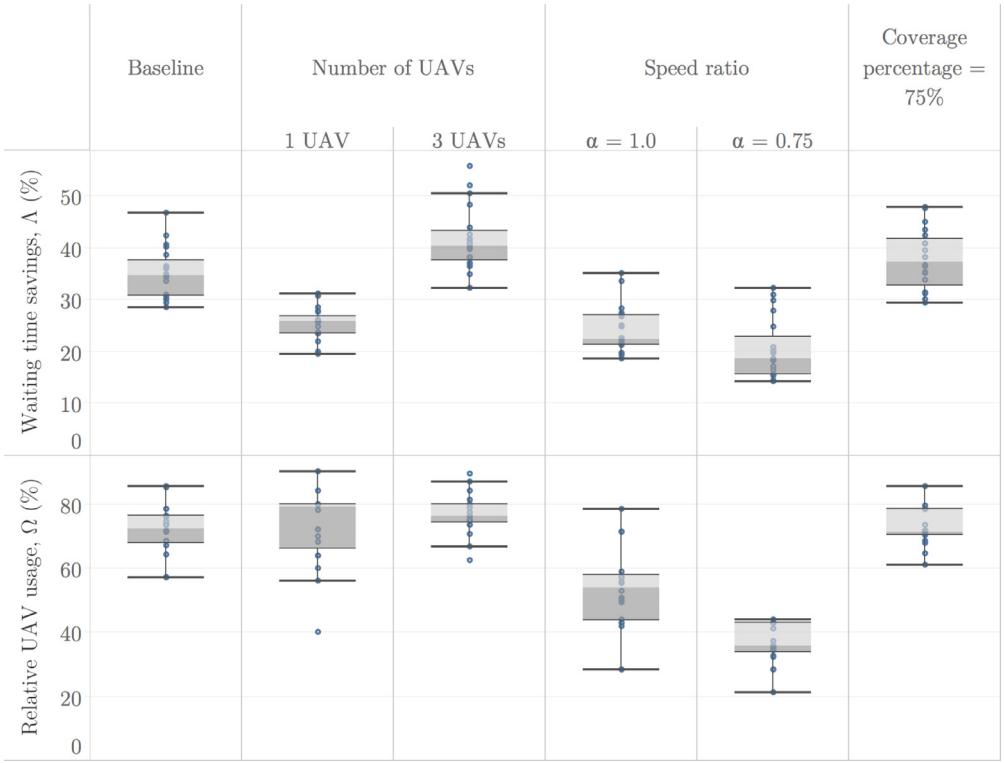

Fig. 8. Summary of comparison of delivery performance between STRPD and TRP on Set 3. Baseline parameter values are $K = 2$, $\alpha = 1.5$, $\mathcal{P} = 25\%$. For each scenario, the chosen parameter value is changed. Each box-plot is obtained based on 20 values.

5.6. Numerical analyses on set 3: Savings of the STRPD over the TRP

In this section, we compare the TDRA with the TRP on several problem instances adopted from Salehipour et al. [66]. The summary of results is illustrated in Fig. 8 with detailed results given in Appendix C where we report the Min-Sum objective function value of the best found TRP solution by Silva et al. [67]. We also report the average and best found solutions, and the average run-time over 10 runs of the TDRA for the baseline scenario. $\Omega\% = \frac{c_S}{c_{\max}} * 100$ denotes the fraction of UAV usage potential (cf., Proposition 1) exploited by a given solution, where c_S is the number of customers served by UAVs in the best found TDRA solution and $c_{\max}$ is the maximum number of customers that could be potentially served by UAVs, as we proved in Proposition 1. The waiting time savings percentage is defined as the relative percentage gap between the STRPD solution and TRP solution, and is calculated by $\Lambda\% = \frac{(MS(TRP)) - MS(STRPD)}{MS(TRP)} * 100$. The instance name *TRP-S10-R1* refers to the problem

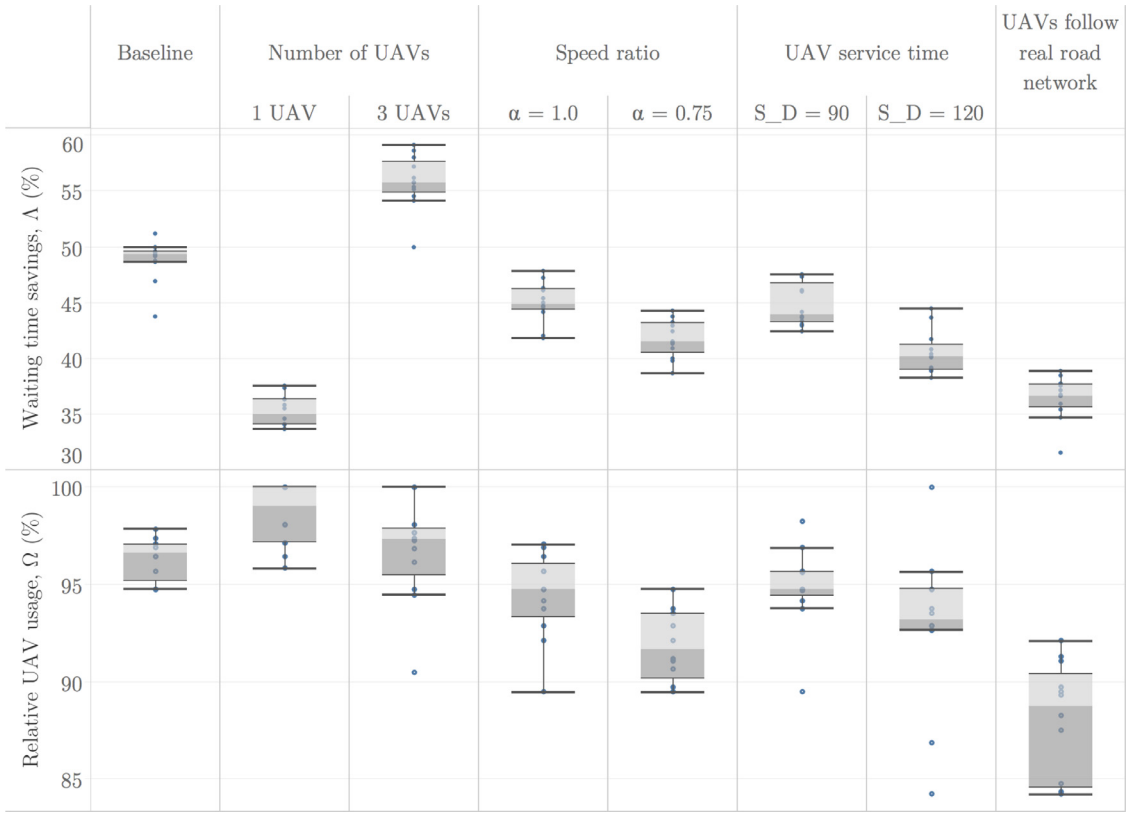

Fig. 9. Summary of comparison of delivery performance between STRPD and TRP on Set 4. Baseline parameter values are $K = 2$, $\alpha = 1.5$, truck speed = 10 km/h, $L = 25$ min, $S_{T_j} = 180$, $S_{D_j} = 60$, real road network for the truck, and Euclidean travel network for the UAVs. For each scenario, the chosen parameter value is changed. Each box-plot is obtained based on 12 values.

instance #1 in the problem set with 10 customers adapted from Salehipour et al. [66]. The other instance names can be read accordingly.

The results in Fig. 8 show up to $\Lambda\% = 35.2\%$ of waiting time reduction with an average of 35.2% savings in customer waiting times in the Baseline scenario. The maximum attained saving is 55.7% in a scenario with $K = 3$ UAVs. The average waiting time savings significantly change for the 1 UAV and 3 UAV scenarios, respectively, indicating the essential role of the number of UAVs for the waiting time performance of the STRPD. The average waiting time saving decreased to 24.1% and 20.4% in scenarios with speed ratios of $\alpha = 1.0$ and $\alpha = 0.75$, respectively, showing the importance of the speed related advantages of UAVs. This becomes even more apparent with respect to Ω%, which decreases from 71.5% to 42.1% and 32.8%, respectively, for $\alpha = 1.0$ and $\alpha = 0.75$. That is, as UAVs become slower, they are used to serve fewer customers compared to the Baseline scenario. However, it is interesting to note that even with UAVs being slower than the truck ($\alpha = 0.75$), we still achieve an average waiting time reduction of 20.4% which is significant. We also see that in the scenario with $\mathcal{P} = 75\%$, the flight time limit increases compared to the Baseline scenario, which uses $\mathcal{P} = 25\%$, therefore, making more customers potentially available to be served by UAVs, and raising the average savings in waiting time to 37.8%.

It is important to notice that in these analyses, Ω is never 100% due to two reasons. First, not all of the customers are located in the flight time limit of the UAVs, forcing the truck to serve such customers. Second, there can be long travel times between some of the customers although they are in the flight time limit of the UAVs. In such cases, the UAVs do not serve these customers because the truck would have to wait for a long time for the UAVs to return at the next customer location. Also, since $c_{\max}$ is affected by the number of UAVs, there is no general valid rule for the value of Ω% with respect to the number of UAVs. Speed ratio, however, clearly affects Ω%, that is, the slower the UAVs, the lower the relative UAV usage.

5.7. Numerical analyses on set 4: Real-world case study in São Paulo, Brazil

In this section, we compare the TDRA with the TRP on real-world problem instances in São Paulo, Brazil. The parameter values are selected as described in Section 5.1. The summary of results is illustrated in Fig. 9 with detailed results given in Appendix D. Tables D.6–D.13 in Appendix D report the Min-Sum objective function values of the best found TRP solutions obtained by the algorithm in [67]. The rest of the values are defined and reported as for Set 3.

Table 3
Comparison of cost objective and customer waiting time objective on Set 3 for the Baseline scenario.

Problem	Optimized for minimum customer waiting times			Optimized for minimum costs		
	Customer waiting times*	Costs	Savings in customer waiting times	Customer waiting times	Costs*	Savings in costs
TRP-S10-R1	905	110.69	39.66	1264	102.8	7.67
TRP-S10-R2	874	121.75	16.13	1015	98.44	23.67
TRP-S10-R3	810	116.51	10.74	897	115.11	1.22
TRP-S10-R4	738	140.65	6.63	787	94.735	48.47
TRP-S10-R5	585	135.69	38.63	811	70.235	93.19
TRP-S20-R1	2,194	163.83	31.72	2,890	125.08	30.98
TRP-S20-R2	2,061	196.53	30.47	2,689	108.02	81.94
TRP-S20-R3	2,116	184.63	10.86	2,346	140.7	31.22
TRP-S20-R4	1,950	166.38	40.35	2,737	102.83	61.8
TRP-S20-R5	2,131	186.21	54.66	3,296	131.16	41.97
TRP-S50-R1	7,756	246.9	45.41	11,278	221.91	11.26
TRP-S50-R2	7,437	256.6	42.32	10,585	176.12	45.69
TRP-S50-R3	7,903	235.18	38.89	10,977	212.79	10.52
TRP-S50-R4	8,360	229.4	33.49	11,160	209.85	9.31
TRP-S50-R5	7,438	239.09	50.69	11,209	188.84	26.61
TRP-S100-R1	23,154	294.1	24.03	28,719	260.3	12.98
TRP-S100-R2	22,224	348.13	48.22	32,942	301.51	15.46
TRP-S100-R3	23,123	331.72	27.31	29,439	299.46	10.77
TRP-S100-R4	24,316	318.72	25.24	30,455	295.25	7.95
TRP-S100-R5	22,801	363.82	35.72	30,947	316.81	14.83
Average			32.56			29.38

The results of the Baseline scenario illustrate average savings in customer waiting time of 48.8% and maximum savings of up to 51.2% compared to the TRP. From our case study analyses, we can confirm that the number of UAVs appears to have the most prominent effect on the waiting time performance of the delivery systems. By changing the number of UAVs to 1 and 3, $\Lambda\%$ changes to 35.2% and 55.8% on average, respectively. This is a particularly relevant insight as the optimal number of UAV per truck plays a key role in the development of future delivery vehicle generations by vehicle manufacturers.

Changing the speed ratio to 1.0 and 0.75 also reduces the average $\Lambda\%$ to 45.0% and 41.7%, respectively, similar to our observations for Set 3. As the service time of UAVs, S_{D_j}, increases to 90 and 120 s, UAVs become less efficient and consequently serve fewer customers. Therefore, the average $\Omega\%$ and, consequently, $\Lambda\%$ reduces.

In the last scenario, in which UAVs are assumed to follow the real road network, $\Lambda\%$ reduces considerably to an average of 36.4%. This is a particularly relevant result as part of the public discussion around the use of UAVs for urban deliveries revolves around the question of whether forcing drones to follow the existing road network due to privacy and safety concerns.

For Set 4, it is important to note that average $\Omega\%$ values are greater than 90% in almost all scenarios. This is primarily due to the high density of demand in the considered 1 km^2 area. Once the truck reaches the demand area, all customers are covered by the 25 min flight time limit of the UAVs and therefore, they all can be reached by UAVs. However, it should be noted that a $\Omega = 0.95\%$, for example, does not mean that 95% of all customers are served by drones. Rather it follows the definition of Ω in Section 5.6. Fig. D.11 in Appendix D illustrates a solution obtained by the TDRA on problem instance S1-3 with 3 UAVs.

5.8. Comparison of cost objective vs. customer waiting time objective

In this section, we aim to compare the performance of the proposed delivery model under two objective functions. The first objective function is minimizing the sum of customer waiting times, while the second objective function minimizes the total transportation costs, formulated as $\sum_{i\in C_0}\sum_{j\in C_E} c_T d_{ij} x_{ij} + \sum_{i\in C_0}\sum_{j\in C}\sum_{s\in C_E}\sum_{u\in U} c_U (d_{ij}+d_{js}) y^u_{ijs}$, where c_T and c_U are the cost coefficients of the truck and UAVs per mile, respectively. We choose $c_T = \$0.485$ and $c_U = \$0.02$, which are calculated based on driver pay rate, energy consumption rate, and energy costs according to [69,70]. The problems in Set 3 are solved for the Baseline scenario with both objective functions and the results are shown in Table 3. When we solve the problem with each of the objective functions, we also calculate the value of the other objective function and compare with the optimal solution we obtain to calculate the savings. The average savings is approximately 30% for both of the objectives. This result indicates that for a delivery company focused on customer waiting times, if the delivery cost is minimized, the waiting time of the customers may increase by an average of 30%. Similarly, the cost objective should be minimized if the focus of the company is on delivery costs rather than customer waiting time.

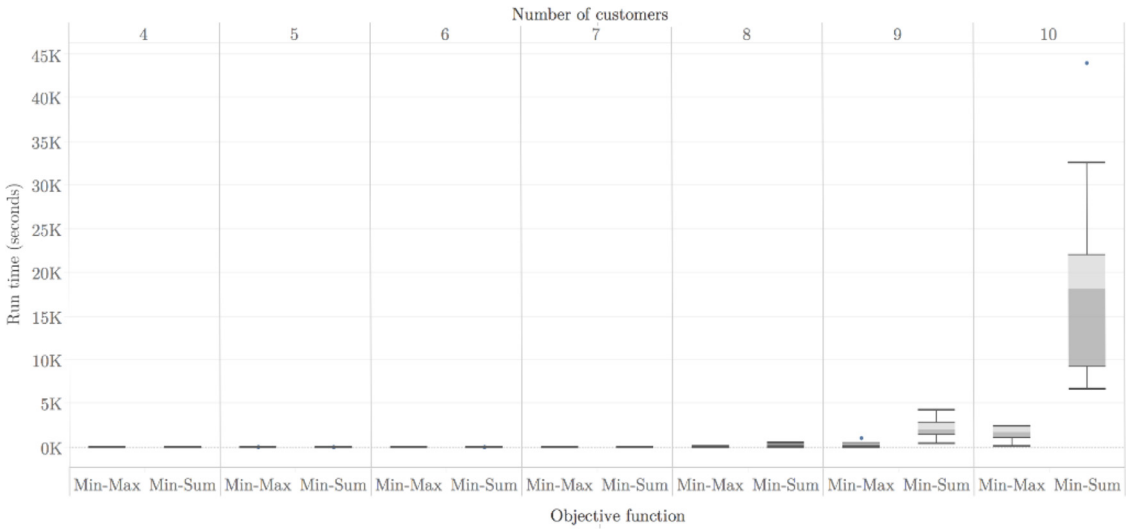

Fig. A10. Effect of objective function on MILP model run-time. Parameter values are $K = 2$, $\alpha = 1.5$, $\mathcal{P} = 25\%$. Each problem is solved with both the *Min-Sum* and *Min-Max* objectives. Each box-plot is obtained based on 10 values.

6. Conclusions and future research

Given the speed of urbanization, increasing traffic loads and congestion levels in cities, and a rapid growth in e-commerce and customer-centric delivery services, fast, efficient, and reliable urban delivery systems are indispensable. In this paper, we introduce a multi-modal, UAV-based delivery problem, so-called the Simultaneous Traveling Repairman Problem with Drones (STRPD), aiming at minimizing customer waiting times for deliveries. This problem considers a single truck that is able to launch multiple UAVs to deliver packages to customers. The movements of truck and UAVs are synchronized to allow for the UAVs to return to the truck at customer locations within their allowable flight time constraint. The objective of the problem is to allocate customers to UAVs and the truck, and to optimize the sequence of deliveries in the routes of the truck and the UAVs to minimize customer waiting time in the system.

The problem is formulated as a Mixed-Integer Linear Programming (MILP), with a bound analysis to examine the maximum attainable reduction in the customer waiting time in the system, when using the proposed system design as opposed to a purely truck-based system. Due to the computational complexity of the problem, a so-called Truck and Drone Routing Algorithm (TDRA) is developed to be able to solve real-world problem instances efficiently. The mathematical model and TDRA are used to solve several problem instances that are either generated, adopted from the literature, or from a real-world case study of e-commerce deliveries in São Paulo, Brazil.

A comparison of the TDRA and the MILP model demonstrates that the proposed algorithm is able to obtain optimal or near optimal solutions with negligible optimality gaps in small-scale problems. Applying the TDRA to adopted problem instances from the literature and comparing it with the TRP reveals that the proposed system can reduce customer waiting time in the system by up to 46.8% when using $K = 2$ UAVs (i.e., for the baseline parameter values of the analysis). We also observe that in high-density demand areas, the use and utilization of UAVs increase as there are more customers to potentially be served by UAVs. According to the results of a real-world case study analysis in downtown São Paulo, Brazil, we demonstrate that we can reap more than 95% of the usage potential of UAVs for parcel delivery in highly dense demand zones.

There are several possible directions for future research. In this paper, we assume that the truck cannot wait for the return of UAVs at the location they were launched from. Future research should consider a more flexible system which allows the truck to decide whether to wait for the return of a UAV at the launch location or to move to the next customer and retrieve the UAV later. The performance of such a system could be compared to the performance of the STRPD presented in this paper. Further, we only consider a single truck in this work, whereas most urban delivery systems would use multiple trucks to serve all demands in a city. Therefore, allowing for multiple trucks that may even share a common UAV fleet would constitute a practical and highly relevant direction for future research.

Acknowledgement

We would like to thank Dr. Marcos Melo Silva for his gracious support by using their algorithm proposed in [67] for solving the Traveling Repairman Problem (TRP) on the problem instances in Set 4. Also, we would like to thank the anonymous reviewers for their constructive comments.

Appendix A. Alternative objective function

As stated in Section 1, a common alternative to minimizing the average waiting time of customers (*Min-Sum* objective) is the objective to minimize the maximum waiting time across customers (*Min-Max*). This objective function is equivalent to minimizing the waiting time of the last visited customer, either by the truck or a UAV. To implement this alternative *Min-Max* objective function in our analysis, we would need to re-define the following objective function of the mathematical model outlined in Eq. (1) though (28) as

$$Minimize \max_{i \in C, u \in U} \left\{ t_i, \hat{t}_{iu} \right\}. \tag{1'}$$

However, to keep the optimization model linear, we need to replace the objective function stated in Eq. (1') by

$$Minimize : W, \tag{1''}$$

where the continuous auxiliary variable W needs to satisfy the additional constraints

$$W \geq t_i, \quad W \geq \hat{t}_{iu}, \quad \forall i \in C, \forall u \in U. \tag{A.1}$$

To illustrate the effects of this alternative optimization objective on the computational performance of our proposed model, the modified mathematical model is used to solve the 70 generated problems in Set 1 to optimality with coverage percentage $\mathcal{P} = 25\%$ and $K = 2$ UAVs. The results are shown in Fig. A.10, grouped by networks with 4 to 10 customers, for both the *Min-Sum* and *Min-Max* objectives. Fig. A.10 shows that the computation time under the *Min-Sum* objective is considerably higher than under the *Min-Max* objective. This confirms the findings in the past researches on the TRP, stating that the *Min-Sum* objective is computationally more expensive than the *Min-Max* alternative due to some particular properties [21,22]. For instance, a small change in the tour, especially, in sequence of visits to the first customers on the route can potentially affect the waiting times of all subsequent customers on the route.

Appendix B. Detailed numerical analyses on set 2

Tables B.4 presents the detailed results for problem set 2 for the Baseline scenario given the Min-Sum objective. The * in column *Mathematical Model Best Solution* indicates that the solution is proven optimal. The tables of the results of the remaining scenarios are not reported for the sake of brevity.

Table B.4
Comparison of TDRA with MILP on Set 2 with baseline parameter values.

Instance	c	MILP		TDRA			
		Best Solution	Run Time (S)	Average Solution	Best Solution	Run Time (S)	Δ (%)
Gen-10-1	10	941.1	10,000.0	948.8	941.0	28.2	0.0
Gen-10-2	10	*765.2	9,235.0	765.2	765.0	28.2	0.0
Gen-10-3	10	692.8	10,000.0	695.3	692.6	25.6	0.0
Gen-10-4	10	1,187.9	10,000.0	1,187.7	1,187.7	25.7	0.0
Gen-10-5	10	1,091.5	10,000.0	1,091.4	1,091.4	28.4	0.0
Gen-10-6	10	792.8	10,000.0	798.9	796.8	28.2	-0.5
Gen-10-7	10	1,134.4	10,000.0	1,134.2	1,134.2	28.8	0.0
Gen-10-8	10	928.1	10,000.0	908.6	908.6	26.0	2.1
Gen-10-9	10	744.8	10,000.0	744.6	744.5	26.5	0.0
Gen-10-10	10	808.4	10,000.0	808.3	808.3	25.5	0.0
Gen-15-1	15	1,911.0	10,000.0	1,916.2	1,826.6	45.3	4.4
Gen-15-2	15	1,582.9	10,000.0	1,634.3	1,554.9	42.7	1.8
Gen-15-3	15	1,605.9	10,000.0	1,641.5	1,522.5	42.3	5.2
Gen-15-4	15	1,359.1	10,000.0	1,406.6	1,307.0	42.7	3.8
Gen-15-5	15	1,628.2	10,000.0	1,790.8	1,613.3	44.1	0.9
Gen-15-6	15	1,531.6	10,000.0	1,659.0	1,499.7	40.3	2.1
Gen-15-7	15	1,559.5	10,000.0	1,627.1	1,437.3	42.2	7.8
Gen-15-8	15	2,013.1	10,000.0	2,130.1	1,849.0	43.8	8.2
Gen-15-9	15	1,930.0	10,000.0	1,976.5	1,825.0	46.2	5.4
Gen-15-10	15	1,836.6	10,000.0	1,678.0	1,657.0	46.7	9.8
Gen-20-1	20	3,272.1	10,000.0	2,667.3	2,391.2	68.8	26.9
Gen-20-2	20	2,317.4	10,000.0	2,550.0	2,204.8	65.3	4.9
Gen-20-3	20	2,321.6	10,000.0	2,809.2	2,274.9	61.2	2.0
Gen-20-4	20	2,111.3	10,000.0	2,359.0	2,001.5	60.8	5.2
Gen-20-5	20	2,841.3	10,000.0	2,807.2	2,492.4	64.3	12.3
Gen-20-6	20	2,488.1	10,000.0	2,526.0	2,128.0	66.7	14.5
Gen-20-7	20	2,911.9	10,000.0	2,890.0	2,497.1	73.5	14.2
Gen-20-8	20	2,762.8	10,000.0	2,637.9	2,209.8	68.8	20.0
Gen-20-9	20	2,619.4	10,000.0	2,568.1	2,224.8	62.4	15.1
Gen-20-10	20	2,676.9	10,000.0	2,810.0	2373.8	64.7	11.3

Appendix C. Detailed numerical analyses on set 3

Tables C.5 present the detailed results for problem set 3 for the baseline scenario given the Min-Sum objective. The tables of the results of the remaining scenarios are not reported for the sake of brevity.

Table C.5
Comparison of STRPD with TRP on Set 3 with baseline parameter values.

Instance	c	TRP	TDRA				
			Average Solution	Best Solution	Run Time (S)	Ω (%)	Λ (%)
TRP-S10-R1	10	1,303.0	906.0	905.0	26.9	75.0	30.5
TRP-S10-R2	10	1,517.0	875.8	874.0	22.3	62.5	42.4
TRP-S10-R3	10	1,233.0	810.0	810.0	24.9	62.5	34.3
TRP-S10-R4	10	1,386.0	738.0	738.0	21.1	75.0	46.8
TRP-S10-R5	10	978.0	585.0	585.0	22.6	62.5	40.2
TRP-S20-R1	20	3,175.0	2,279.6	2,194.0	56.0	85.7	30.9
TRP-S20-R2	20	3,248.0	2,126.3	2,061.0	64.3	64.3	36.5
TRP-S20-R3	20	3,570.0	2,238.0	2,116.0	55.5	92.9	40.7
TRP-S20-R4	20	2,983.0	2,011.7	1,950.0	59.6	78.6	34.6
TRP-S20-R5	20	3,248.0	2,292.9	2,131.0	61.5	78.6	34.4
TRP-S50-R1	50	12,198.0	8,043.7	7,756.0	216.6	76.5	36.4
TRP-S50-R2	50	11,621.0	7,572.4	7,437.0	218.6	82.4	36.0
TRP-S50-R3	50	12,139.0	8,353.8	7,903.0	213.8	70.6	34.9
TRP-S50-R4	50	13,071.0	8,886.0	8,360.0	211.1	73.5	36.0
TRP-S50-R5	50	12,126.0	7,912.1	7,438.0	215.5	64.7	38.7
TRP-S100-R1	100	32,779.0	23,678.0	23,154.0	1,640.9	64.7	29.4
TRP-S100-R2	100	33,435.0	23,037.0	22,224.0	1,610.5	70.6	33.5
TRP-S100-R3	100	32,390.0	23,528.0	23,123.0	1,660.4	70.6	28.6
TRP-S100-R4	100	34,733.0	24,879.0	24,316.0	1,625.5	67.6	30.0
TRP-S100-R5	100	32,598.0	23,510.0	22,801.0	1,726.5	66.2	30.1
Average						72.2	35.2

Appendix D. Detailed numerical analyses on set 4

Tables D.6 –D.13 present the detailed results on problem set 4 given the Min-Sum objective.

Table D.6
Comparison of STRPD with TRP on Set 4 with baseline parameter values.

Instance	c	TRP	TDRA				
			Average Solution	Best Solution	Run Time (S)	Ω (%)	Λ (%)
S1-1	48	386,066.0	208,420.0	194,640.0	212.8	96.9	49.6
S1-2	48	398,249.1	207,290.0	204,370.0	209.2	96.9	48.7
S1-3	28	179,491.8	93,157.0	89,819.0	85.2	94.7	50.0
S1-4	28	166,068.1	95,717.0	93,497.0	91.4	94.7	43.7
S1-5	69	720,066.7	384,580.0	363,430.0	396.5	97.8	49.5
S1-6	69	715,450.4	376,640.0	349,490.0	386.3	95.7	51.2
S2-1	84	985,450.2	536,590.0	496,760.0	705.7	96.4	49.6
S2-2	84	984,780.2	551,720.0	522,990.0	656.2	96.4	46.9
S2-3	56	503,982.8	274,570.0	258,830.0	256.9	94.7	48.6
S2-4	56	503,711.7	278,960.0	256,290.0	257.2	97.4	49.1
S2-5	101	1,349,830.9	744,480.0	685,240.0	1,171.6	97.1	49.2
S2-6	101	1,366,058.3	735,950.0	691,320.0	1,210.5	97.1	49.4
Average						96.3	48.8

Table D.7
Comparison of STRPD with TRP on Set 4 with 1 UAV.

Instance	c	TRP	TDRA				
			Average Solution	Best Solution	Run Time (S)	Ω (%)	Λ (%)
S1-1	48	386,066.0	263,000.0	252,790.0	234.6	100.0	34.5
S1-2	48	398,249.1	254,210.0	248,840.0	240.0	95.8	37.5
S1-3	28	179,491.8	114,250.0	112,460.0	76.6	100.0	37.3
S1-4	28	166,068.1	111,340.0	106,670.0	74.3	100.0	35.8
S1-5	69	720,066.7	505,330.0	474,820.0	468.3	97.1	34.1
S1-6	69	715,450.4	510,070.0	461,400.0	476.5	97.1	35.5
S2-1	84	985,450.2	685,930.0	653,520.0	1,350.7	100.0	33.7
S2-2	84	984,780.2	684,310.0	649,630.0	1,431.8	100.0	34.0
S2-3	56	503,982.8	334,630.0	321,090.0	343.8	96.4	36.3
S2-4	56	503,711.7	343,610.0	320,610.0	368.8	100.0	36.4
S2-5	101	1,349,830.9	983,320.0	894,880.0	1,885.1	98.0	33.7
S2-6	101	1,366,058.3	979,360.0	901,190.0	1,940.7	98.0	34.0
Average						98.6	35.2

Table D.8
Comparison of STRPD with TRP on Set 4 with 3 UAVs.

Instance	c	TRP	TDRA				
			Average Solution	Best Solution	Run Time (S)	Ω (%)	Λ (%)
S1-1	48	386,066.0	179,050.0	170,930.0	210.5	94.4	55.7
S1-2	48	398,249.1	178,630.0	174,690.0	236.6	97.2	56.1
S1-3	28	179,491.8	82,022.0	80,072.0	96.6	100.0	55.4
S1-4	28	166,068.1	84,807.0	83,177.0	101.5	90.5	49.9
S1-5	69	720,066.7	322,600.0	298,170.0	390.7	98.1	58.6
S1-6	69	715,450.4	320,980.0	300,430.0	404.0	96.2	58.0
S2-1	84	985,450.2	459,890.0	441,990.0	579.2	100.0	55.1
S2-2	84	984,780.2	455,760.0	436,060.0	603.8	96.8	55.7
S2-3	56	503,982.8	237,290.0	230,950.0	267.6	97.6	54.2
S2-4	56	503,711.7	237,020.0	229,230.0	243.1	97.6	54.5
S2-5	101	1,349,830.9	605,920.0	578,410.0	1,032.0	97.4	57.1
S2-6	101	1,366,058.3	611,560.0	557,830.0	1,073.3	94.7	59.2
Average						96.7	55.8

Table D.9
Comparison of STRPD with TRP on Set 4 with speed ratio 1.0.

Instance	c	TRP	TDRA				
			Average Solution	Best Solution	Run Time (S)	Ω (%)	Λ (%)
S1-1	48	386,066.0	227,720.0	213,810.0	230.9	93.8	44.6
S1-2	48	398,249.1	221,870.0	214,590.0	228.8	96.9	46.1
S1-3	28	179,491.8	99,088.0	93,587.0	85.7	94.7	47.9
S1-4	28	166,068.1	100,390.0	96,583.0	85.7	89.5	41.8
S1-5	69	720,066.7	426,770.0	393,070.0	506.3	95.7	45.4
S1-6	69	715,450.4	420,620.0	399,510.0	530.5	95.7	44.2
S2-1	84	985,450.2	583,740.0	571,890.0	975.7	92.9	42.0
S2-2	84	984,780.2	582,750.0	541,920.0	1,001.3	96.4	45.0
S2-3	56	503,982.8	292,720.0	279,750.0	340.3	94.7	44.5
S2-4	56	503,711.7	298,870.0	278,540.0	341.4	92.1	44.7
S2-5	101	1,349,830.9	788,170.0	725,510.0	1,834.9	94.1	46.3
S2-6	101	1,366,058.3	777,030.0	721,360.0	1,818.1	97.1	47.2
Average						94.5	45.0

Table D.10
Comparison of STRPD with TRP on Set 4 with speed ratio 0.75.

Instance	c	TRP	TDRA				
			Average Solution	Best Solution	Run Time (S)	Ω (%)	Λ (%)
S1-1	48	386,066.0	249,490.0	236,900.0	234.5	93.8	38.6
S1-2	48	398,249.1	244,290.0	227,170.0	236.8	90.6	43.0
S1-3	28	179,491.8	110,270.0	104,980.0	86.8	89.5	41.5
S1-4	28	166,068.1	108,890.0	99,617.0	91.2	89.5	40.0
S1-5	69	720,066.7	454,780.0	421,900.0	578.3	93.5	41.4
S1-6	69	715,450.4	447,420.0	402,440.0	571.1	93.5	43.8
S2-1	84	985,450.2	614,200.0	549,520.0	1,044.7	91.1	44.2
S2-2	84	984,780.2	620,910.0	581,850.0	1,066.6	92.9	40.9
S2-3	56	503,982.8	313,730.0	303,510.0	357.7	94.7	39.8
S2-4	56	503,711.7	317,160.0	290,290.0	357.0	92.1	42.4
S2-5	101	1,349,830.9	843,510.0	792,600.0	2,129.0	91.2	41.3
S2-6	101	1,366,058.3	836,780.0	774,870.0	2,179.0	89.7	43.3
Average						91.8	41.7

Table D.11
Comparison of STRPD with TRP on Set 4 with $S_{D_j} = 90$ s.

Instance	c	TRP	TDRA				
			Average Solution	Best Solution	Run Time (S)	Ω (%)	Λ (%)
S1-1	48	386,066.0	225,640.0	217,210.0	220.8	93.8	43.7
S1-2	48	398,249.1	218,950.0	210,030.0	193.4	96.9	47.3
S1-3	28	179,491.8	97,723.0	94,228.0	88.6	89.5	47.5
S1-4	28	166,068.1	97,857.0	95,632.0	86.7	94.7	42.4
S1-5	69	720,066.7	416,220.0	388,980.0	410.5	95.7	46.0
S1-6	69	715,450.4	401,860.0	376,410.0	407.0	95.7	47.4
S2-1	84	985,450.2	592,040.0	561,070.0	700.7	98.2	43.1
S2-2	84	984,780.2	581,910.0	555,120.0	746.9	94.6	43.6
S2-3	56	503,982.8	295,350.0	285,460.0	276.3	94.7	43.4
S2-4	56	503,711.7	295,110.0	287,300.0	270.4	94.7	43.0
S2-5	101	1,349,830.9	799,140.0	754,550.0	1,336.2	94.1	44.1
S2-6	101	1,366,058.3	780,300.0	736,590.0	1,336.0	95.6	46.1
Average						94.8	44.8

Table D.12
Comparison of STRPD with TRP on Set 4 with $S_{D_j} = 120$ s.

Instance	c	TRP	TDRA				
			Average Solution	Best Solution	Run Time (S)	Ω (%)	Λ (%)
S1-1	48	386,066.0	241,370.0	230,150.0	213.6	93.8	40.4
S1-2	48	398,249.1	235,970.0	224,450.0	211.2	100.0	43.6
S1-3	28	179,491.8	102,190.0	99,650.0	89.4	94.7	44.5
S1-4	28	166,068.1	102,400.0	99,447.0	93.5	84.2	40.1
S1-5	69	720,066.7	462,550.0	438,740.0	454.7	93.5	39.1
S1-6	69	715,450.4	449,300.0	416,840.0	431.2	95.7	41.7
S2-1	84	985,450.2	641,120.0	608,450.0	784.9	92.9	38.3
S2-2	84	984,780.2	635,140.0	598,760.0	818.9	92.9	39.2
S2-3	56	503,982.8	317,110.0	308,000.0	300.7	94.7	38.9
S2-4	56	503,711.7	317,450.0	298,500.0	287.6	86.8	40.7
S2-5	101	1,349,830.9	863,530.0	824,260.0	1,953.5	92.6	38.9
S2-6	101	1,366,058.3	859,100.0	816,700.0	1,646.0	92.6	40.2
Average						92.9	40.5

Table D.13
Comparison of STRPD with TRP on Set 4 with real travel network for UAVs.

Instance	c	TRP	TDRA				
			Average Solution	Best Solution	Run Time (S)	Ω (%)	Λ (%)
S1-1	48	386,066.0	265,950.0	247,640.0	255.5	84.4	35.9
S1-2	48	398,249.1	260,400.0	248,120.0	262.6	87.5	37.7
S1-3	28	179,491.8	119,330.0	113,870.0	95.6	84.2	36.6
S1-4	28	166,068.1	116,850.0	113,800.0	91.1	89.5	31.5
S1-5	69	720,066.7	491,920.0	470,290.0	652.8	84.8	34.7
S1-6	69	715,450.4	493,030.0	447,210.0	659.7	91.3	37.5
S2-1	84	985,450.2	659,720.0	602,720.0	1,159.9	91.1	38.8
S2-2	84	984,780.2	655,210.0	636,060.0	1,107.5	89.3	35.4
S2-3	56	503,982.8	332,590.0	319,390.0	374.2	92.1	36.6
S2-4	56	503,711.7	337,430.0	319,000.0	377.5	84.2	36.7
S2-5	101	1,349,830.9	884,300.0	848,280.0	2,394.6	89.7	37.2
S2-6	101	1,366,058.3	882,290.0	841,560.0	2,369.5	88.2	38.4
Average						88.0	36.4

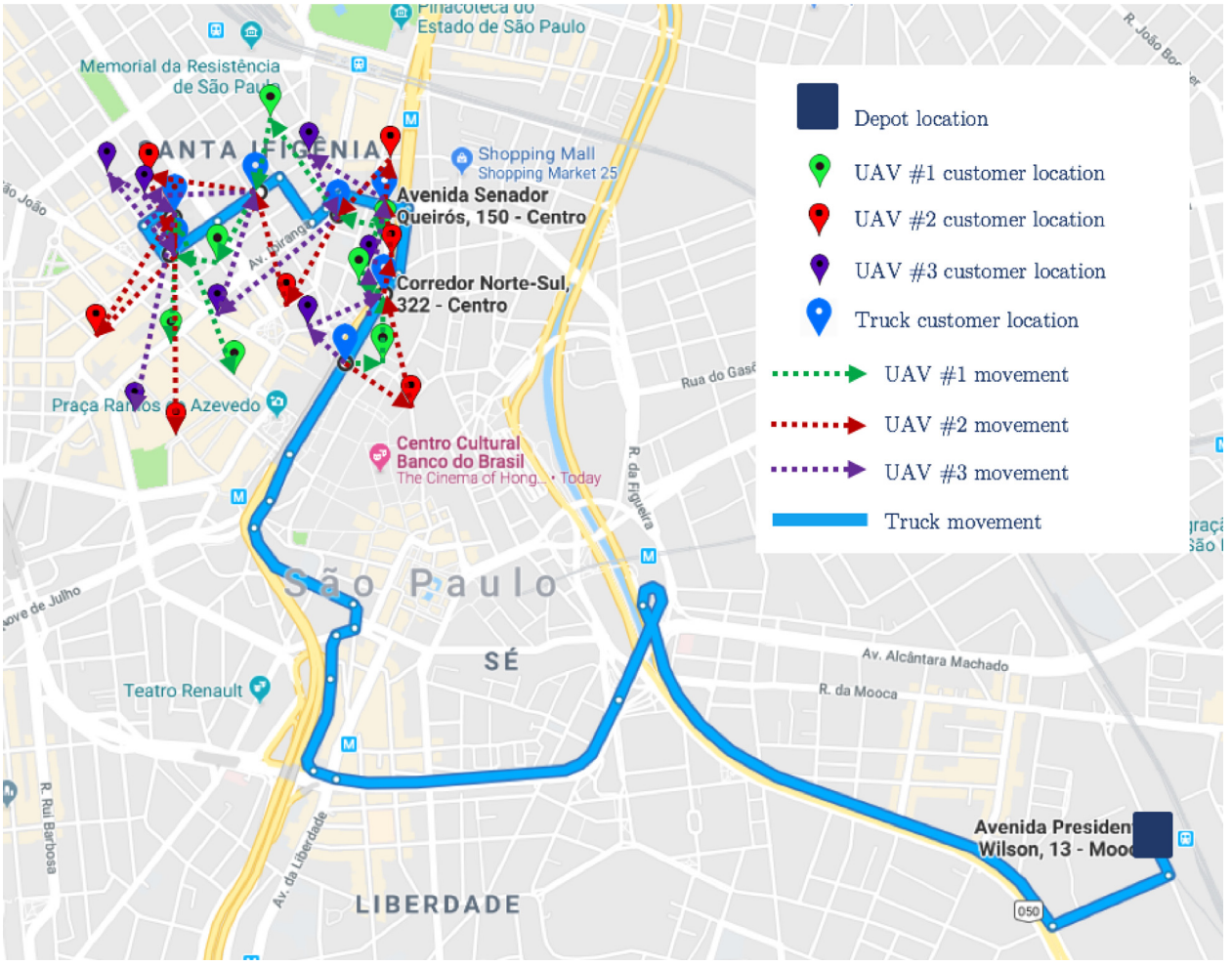

Fig. D.11. The obtained solution to the STRPD by the TDRA on problem instance S1-3 with 3 UAVs.

References

[1] R. Goodman, Whatever you call it, just don't think of last-mile logistics, last, Global Logist. Supp. Chain Strat. 9 (12) (2005) 46–51.
[2] K. Jacobs, S. Warner, M. Rietra, L. Mazza, J. Buvat, A. Khadikar, S. Cherian, Y. Khemka, The last-mile delivery challenge, Digital Report, Capgemini Research Institute, 2019.
[3] S. Rosenbush, L. Stevens, At UPS, the Algorithm Is the Driver, 2015, https://www.wsj.com/articles/at-ups-the-algorithm-is-the-driver-1424136536.
[4] M. Winkenbach, P.R. Kleindorfer, S. Spinler, Enabling urban logistics services at La Poste through multi-echelon location-routing, Transp. Sci. 50 (2) (2015) 520–540.
[5] M. Winkenbach, A. Roset, S. Spinler, Strategic redesign of urban mail and parcel networks at la Poste, Interfaces (Providence) 46 (5) (2016) 445–458.
[6] T.G. Crainic, N. Ricciardi, G. Storchi, Advanced freight transportation systems for congested urban areas, Transp. Res. Part C Emerg. Technol. 12 (2) (2004) 119–137.
[7] World Bank. 2009. World Development Report 2009 : Reshaping Economic Geography. World Bank. © World Bank. https://openknowledge.worldbank.org/handle/10986/5991 License: CC BY 3.0 IGO. http://hdl.handle.net/10986/5991.
[8] Q.M. Ha, Y. Deville, Q.D. Pham, M.H. Hà, On the min-cost traveling salesman problem with drone, Transp. Res. Part C Emerg. Technol. 86 (2018) 597–621.
[9] M. Franco, DHL uses completely autonomous system to deliver consumer goods by drone, 2016, http://newatlas.com/dhl-drone-delivery/43248/.
[10] S. Perez, L. Kolodny, UPS tests show delivery drones still need work, 2017, https://techcrunch.com/2017/02/21/ups-tests-show-delivery-drones-still-need-work/.
[11] Amazon.com, Amazon Prime Air, 2013, https://www.amazon.com/Amazon-Prime-Air/b?node=8037720011.
[12] A. Levin, Alphabet and Chipotle Are Bringing Burrito Delivery Drones to Campus, 2016, https://www.bloomberg.com/news/articles/2016-09-08/burrito-by-drone-coming-to-campus-in-test-of-alphabet-s-delivery.
[13] L. Kolodny, Zipline raises $25 million to deliver medical supplies by drone, 2016, https://techcrunch.com/2016/11/09/zipline-raises-25-million-to-deliver-medical-supplies-by-drone.
[14] A. Levin, Alphabet's Project Wing Delivery Drones to Be Tested in U.S, 2016, https://www.bloomberg.com/news/articles/2016-08-02/google-s-project-wing-delivery-drones-to-be-tested-at-u-s-site.
[15] F. Lardinois, Amazon starts Prime Air drone delivery trial in the UK but only with two beta users, 2016, https://techcrunch.com/2016/12/14/amazons-prime-air-delivery-uk/.
[16] A. Zaleski, Cities Seek Deliverance From the E-Commerce Boom, 2017, https://www.citylab.com/transportation/2017/04/cities-seek-deliverance-from-the-e-commerce-boom/523671/.
[17] L. Kolodny, Mercedes-Benz and Matternet unveil vans that launch delivery drones, 2016, https://techcrunch.com/2016/09/07/mercedes-benz-and-matternet-unveil-vans-that-launch-delivery-drones/.
[18] c. Koç, T. Bektaş, O. Jabali, G. Laporte, Thirty years of heterogeneous vehicle routing, Eur. J. Oper. Res. 249 (1) (2016) 1–21.
[19] M. Moshref-Javadi, S. Lee, The customer-centric, multi-commodity vehicle routing problem with split delivery, Expert Syst. Appl. 56 (2016) 335–348.
[20] M. Joerss, J. Schröder, F. Neuhaus, C. Klink, F. Mann, Parcel delivery: the future of last mile, Digital Report, McKinsey & Company, 2016.
[21] S.U. Ngueveu, C. Prins, R.W. Calvo, An effective memetic algorithm for the cumulative capacitated vehicle routing problem, Comput. Oper. Res. 37 (11) (2010) 1877–1885.
[22] A. Blum, P. Chalasani, D. Coppersmith, B. Pulleyblank, P. Raghavan, M. Sudan, The minimum latency problem, in: Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, ACM, 1994, pp. 163–171.
[23] M.M. Flood, The traveling-salesman problem, Oper. Res. 4 (1) (1956) 61–75.
[24] K.L. Hoffman, M. Padberg, G. Rinaldi, Traveling Salesman Problem, in: Encyclopedia of Operations Research and Management Science, Springer, 2013, pp. 1573–1578.
[25] N. Christofides, The vehicle routing problem, Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle 10 (V1) (1976) 55–70.
[26] M.M. Solomon, Algorithms for the vehicle routing and scheduling problems with time window constraints, Oper. Res. 35 (2) (1987) 254–265.
[27] G. Laporte, M. Desrochers, Y. Nobert, Two exact algorithms for the distance-constrained vehicle routing problem, Networks 14 (1) (1984) 161–172.
[28] F. Afrati, S. Cosmadakis, C.H. Papadimitriou, G. Papageorgiou, N. Papakostantinou, The complexity of the travelling repairman problem, RAIRO Informatique théorique 20 (1) (1986) 79–87.
[29] J.N. Tsitsiklis, Special cases of traveling salesman and repairman problems with time windows, Networks 22 (3) (1992) 263–282.
[30] M. Goemans, J. Kleinberg, An improved approximation ratio for the minimum latency problem, Math. Prog. 82 (1–2) (1998) 111–124.
[31] N. Mladenović, D. Urošević, S. Hanafi, Variable neighborhood search for the travelling Deliveryman problem, 4OR 11 (1) (2013) 57–73.
[32] F. Angel-Bello, A. Alvarez, I. García, Two improved formulations for the minimum latency problem, Appl. Math. Model. 37 (4) (2013) 2257–2266.
[33] M. Moshref-Javadi, S. Lee, A taxonomy to the class of Minimum Latency Problems, in: Proceedings of the IIE Annual Conference, 2013, pp. 3896–3905.
[34] M. Moshref-Javadi, S. Lee, Using drones to minimize latency in distribution systems, in: Proceedings of the IIE Annual Conference. Proceedings, Institute of Industrial and Systems Engineers (IISE), 2017, pp. 235–240.
[35] N. Bjelić, M. Vidović, D. Popović, Variable neighborhood search algorithm for heterogeneous traveling repairmen problem with time windows, Expert Syst. Appl. 40 (15) (2013) 5997–6006.
[36] M. Moshref-Javadi, S. Lee, The latency location-routing problem, Eur. J. Oper. Res. 255 (2) (2016) 604–619.
[37] J.C. Rivera, H.M. Afsar, C. Prins, Mathematical formulations and exact algorithm for the multitrip cumulative capacitated single-vehicle routing problem, Eur. J. Oper. Res. 249 (1) (2016) 93–104.
[38] J.C. Gerdessen, Vehicle routing problem with trailers, Eur. J. Oper. Res. 93 (1) (1996) 135–147.
[39] C.C. Murray, A.G. Chu, The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery, Transp. Res. Part C Emerg. Technol. 54 (2015) 86–109.
[40] N. Agatz, P. Bouman, M. Schmidt, Optimization approaches for the traveling salesman problem with drone, Transp. Sci. 52 (4) (2018) 965–981, doi:10.1287/trsc.2017.0791.
[41] X. Wang, S. Poikonen, B. Golden, The vehicle routing problem with drones: several worst-case results, Opt. Lett. 11 (4) (2017) 679–697.
[42] H. Savuran, M. Karakaya, Efficient route planning for an unmanned air vehicle deployed on a moving carrier, Soft Comput. 20 (7) (2016) 2905–2920.
[43] Y.S. Chang, H.J. Lee, Optimal delivery routing with wider drone-delivery areas along a shorter truck-route, Expert Syst. Appl. 104 (2018) 307–317.
[44] M.S. Othman, A. Shurbevski, Y. Karuno, H. Nagamochi, Routing of carrier-vehicle systems with dedicated last-stretch delivery vehicle and fixed carrier route, J. Inf. Process. 25 (2017) 655–666.
[45] J.F. Campbell, D.C. Sweeney Ii, J. Zhang, Strategic design for delivery with trucks and drones, Technical Report, University of Missouri St. Louis, 2017.
[46] J.G. Carlsson, S. Song, Coordinated logistics with a truck and a drone, Manag. Sci. 64 (9) (2018) 4052–4069, doi:10.1287/mnsc.2017.2824.
[47] A. Otto, N. Agatz, J. Campbell, B. Golden, E. Pesch, Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: a survey, Networks 72 (4) (2018) 411–458.
[48] M. Tavana, K. Khalili-Damghani, F.J. Santos-Arteaga, M.-H. Zandi, Drone shipping versus truck delivery in a cross-docking system with multiple fleets and products, Expert Syst. Appl. 72 (2017) 93–107.
[49] P. Bouman, N. Agatz, M. Schmidt, Dynamic programming approaches for the traveling salesman problem with drone, Networks 72 (4) (2018) 528–542.
[50] A.M. Ham, Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming, Transp. Res. Part C Emerg. Technol. 91 (2018) 1–14.

[51] P. Kitjacharoenchai, M. Ventresca, M. Moshref-Javadi, S. Lee, J.M.A. Tanchoco, P.A. Brunese, Multiple traveling salesman problem with drones: mathematical model and heuristic approach, Comput. Ind. Eng. 129 (2019) 14–30.
[52] E.E. Yurek, H.C. Ozmutlu, A decomposition-based iterative optimization algorithm for traveling salesman problem with drone, Transp. Res. Part C Emerg. Technol. 91 (2018) 249–262.
[53] N. Boysen, D. Briskorn, S. Fedtke, S. Schwerdfeger, Drone delivery from trucks: drone scheduling for given truck routes, Networks 72 (4) (2018) 506–527.
[54] S. Chowdhury, A. Emelogu, M. Marufuzzaman, S.G. Nurre, L. Bian, Drones for disaster response and relief operations: a continuous approximation model, Int. J. Prod. Econ. 188 (2017) 167–184.
[55] D. Wang, The economics of drone delivery, 2016, http://spectrum.ieee.org/automaton/robotics/drones/the-economics-of-drone-delivery.
[56] A.M. Campbell, D. Vandenbussche, W. Hermann, Routing for relief efforts, Transp. Sci. 42 (2) (2008) 127–145.
[57] S. Ropke, D. Pisinger, An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows, Transp. Sci. 40 (4) (2006) 455–472.
[58] P. Shaw, A New Local Search Algorithzm Providing High Quality Solutions to Vehicle Routing Problems, APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK, 1997.
[59] A. Hemmati, L.M. Hvattum, K. Fagerholt, I. Norstad, Benchmark suite for industrial and tramp ship routing and scheduling problems, INFOR: Inf. Syst. Oper. Res. 52 (1) (2014) 28–38.
[60] A. Megahed, M. Goetschalckx, A modeling framework and local search solution methodology for a production-distribution problem with supplier selection and time-aggregated quantity discounts, Appl. Math. Model. 68 (2019) 198–218.
[61] Y. Crama, M. Schyns, Simulated annealing for complex portfolio selection problems, Eur. J. Oper. Res. 150 (3) (2003) 546–571.
[62] C. Voudouris, E. Tsang, Guided local search and its application to the traveling salesman problem, Eur. J. Oper. Res. 113 (2) (1999) 469–499.
[63] W. Gander, G.H. Golub, R. Strebel, Least-squares fitting of circles and ellipses, BIT Numer. Math. 34 (4) (1994) 558–578.
[64] Python, Python Programming Language, 2019, https://www.python.org/.
[65] Gurobi, Gurobi Optimization Solver, 2019, https://www.gurobi.com/.
[66] A. Salehipour, K. Sörensen, P. Goos, O. Bräysy, Efficient GRASP+ VND and GRASP+ VNS metaheuristics for the traveling repairman problem, 4OR 9 (2) (2011) 189–209.
[67] M.M. Silva, A. Subramanian, T. Vidal, L.S. Ochi, A simple and effective metaheuristic for the minimum latency problem, Eur. J. Oper. Res. 221 (3) (2012) 513–520.
[68] Google-GDM, Google Distance Matrix (GDM) API, 2018, https://developers.google.com/maps/documentation/distance-matrix/.
[69] A. Kharpal, Amazon wins patent for a flying warehouse that will deploy drones to deliver parcels in minutes, 2016, https://www.cnbc.com/2016/12/29/amazon-flying-warehouse-deploy-delivery-drones-patent.html.
[70] J.J. Yoon, The traveling salesman problem with multiple drones: an optimization model for last-mile delivery, Massachusetts Institute of Technology, 2018 Ph.D. thesis.