A-VRPD: Automating Drone-Based Last-Mile Delivery Using Self-Driving Cars

Navid Mohammad Imran, Sabyasachee Mishra, and Myounggyu Won, *Member, IEEE*

***Abstract*— Drone-based last-mile delivery is an emerging technology that uses drones loaded onto a truck to deliver parcels to customers. In this paper, we introduce a fully automated system for drone-based last-mile delivery through incorporation of autonomous vehicles (AVs). A novel problem called the autonomous vehicle routing problem with drones (A-VRPD) is defined. A-VRPD is to select AVs from a pool of available AVs based on crowd sourcing, assign selected AVs to customer groups, and schedule routes for selected AVs to optimize the total operational cost. We formulate A-VRPD as a Mixed Integer Linear Program (MILP) and propose an optimization framework to solve the problem. A greedy algorithm is also developed to significantly improve the running time for large-scale delivery scenarios. Extensive simulations were conducted taking into account real-world operational costs for different types of AVs, traveled distances calculated considering the real-time traffic conditions using Google Map API, and varying load capacities of AVs. We evaluated the performance in comparison with two different state-of-the-art solutions: an algorithm designed to address the traditional vehicle routing problem with drones (VRP-D), which involves human-operated trucks working in tandem with drones to deliver parcels, and an algorithm for the two echelon vehicle routing problem (2E-VRP), wherein parcels are first transported to satellite locations and subsequently delivered from those satellites to the customers. The results indicate a substantial increase in profits for both the delivery company and vehicle owners compared with the state-of-the-art algorithms.**

***Index Terms*— Drone-based last-mile delivery, vehicle routing problem with drones, traveling salesman problem with drones.**

I. Introduction

THE drone-based last-mile delivery [1] is to utilize drones in delivering parcels. There are different methods of utilizing drones for last-mile delivery, *e.g.,* drone-only delivery, drone-truck delivery as separate entities, and collaborative drone-truck delivery. In this paper, drone-based last-mile delivery refers specifically to the collaborative approach in which one or more drones are loaded onto a truck, and both the drones and the truck work together to deliver parcels to customers. Due to their ability to fly, drones can take a more direct route to customers and are less affected by ground obstacles [2]. Additionally, the speed of drones is faster than conventional trucks, allowing for much faster access to customers [3]. These unique characteristics of drones enable significantly reduced operational cost and increased delivery speed for last-mile logistics, potentially revolutionizing traditional parcel delivery systems [4]. Numerous logistics companies such as Amazon [5], Google [6], DHL [7], and Alibaba [8] have been increasingly adopting drone-based delivery solutions.

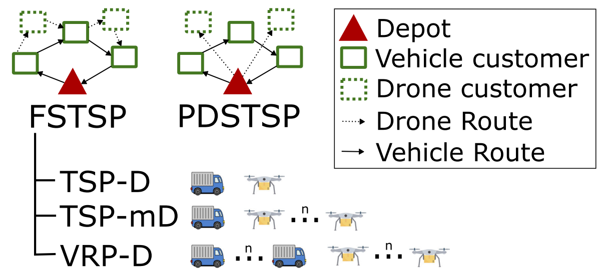


Fig. 1. An illustration of major drone-based last-mile delivery problems.

A significant amount of research has been devoted to developing collaborative parcel delivery systems that involve both trucks and drones. Murray and Chu were the first who formally defined the problem of combining drones with traditional trucks to deliver parcels more effectively [9]. In their pioneering work, two different problems were introduced: the Flying Sidekick Traveling Salesman Problem (FSTSP) and the Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP). As shown in Fig. 1, in FSTSP, a single vehicle works in tandem with a drone to deliver parcels (*i.e.,* a drone loaded on a truck, traveling together, can fly to serve a customer on its own), and in PDSTSP, a fleet of drones and vehicles deliver parcels separately from the depot.

There exist numerous variants of FSTSP [10]. One of the most actively researched one is the traveling salesman problem with drone (TSP-D) [11], [12] where a single drone and a single truck is assumed. This problem has been addressed in many papers [13], [14], [15], [16], [17], [18], [19]. There is another well-researched variant known as TSP-mD [20], [21], [22]. TSP-mD is an extension of TSP-D where a truck is loaded with m drones which are launched from the truck to serve customers and rejoin the truck later at a different location. Other works investigated more general delivery scenarios with multiple drones and multiple trucks, which is called the vehicle routing problem with drones

Manuscript received 8 May 2022; revised 19 December 2022 and 26 March 2023; accepted 4 April 2023. Date of publication 18 April 2023; date of current version 30 August 2023. The Associate Editor for this article was H. Jula. *(Corresponding author: Myounggyu Won.)*

Navid Mohammad Imran and Myounggyu Won are with the Department of Computer Science, The University of Memphis, Memphis, TN 35152 USA (e-mail: nimran@memphis.edu; mwon@memphis.edu).

Sabyasachee Mishra is with the Department of Civil Engineering, The University of Memphis, Memphis, TN 35152 USA (e-mail: smishra3@memphis.edu).

Digital Object Identifier 10.1109/TITS.2023.3266460

(VRP-D) [23], [24], [25], [26], [27]. Our work is closely related to VRP-D; as such, a specific focus is given to the VRP-D problem. We present the detailed review of numerous solutions for TSP-D, TSP-mD, and VRP-D in Section II.

In contrast to existing VRP-D solutions, we study the next-generation drone-based last-mile delivery based on autonomous vehicles (AVs). Due to the significant advances of AV technologies [28], innovative mobility services leveraging a large batch of AVs [29], [30] are expected to emerge such as automated taxi [31] and fully automated parcel delivery systems [32]. In light of these technological advancements and evolving societal needs, we introduce a new variant of VRP-D in this article, termed the Autonomous Vehicle Routing Problem with Drones (A-VRPD). Our aim is to simultaneously optimize the scheduling and routes of AVs in tandem with drones to minimize overall operational costs. Through this article, we investigate a solution for A-VRPD that enables fully automated drone-based last-mile delivery.

A-VRPD has a number of differences compared to the traditional VRP-D and presents several novel challenges. First, in A-VRPD, there is no driver; in contrast, in VRP-D, when drones are delivering, trucks are also used to perform delivery. This makes direct application of an existing solution for VRP-D for solving A-VRPD difficult. Second, the problem complexity for enabling rendezvous motion of trucks and drones is relaxed in A-VRPD, thereby requiring a completely new mathematical model. Third, since A-VRPD involves participation of individual AVs based on crowdsourcing, the number of AVs is significantly larger than that for traditional trucks for VRP-D. Therefore, the scalability becomes a crucial issue for A-VRPD. Furthermore, each AV's unique properties such as varying load capacities, fuel levels, fuel consumption rates, vehicle types, initial locations, and available time frames must be taken into account in designing a solution for A-VRPD. Fourth, another notable difference is that traditional VRPD formulations assume a single customer per drone trip. On the other hand, A-VRPD allows drones to serve multiple customers at a single trip.

Considering the unique aspects of A-VRPD, in this article, we present a novel solution to solve A-VRPD. The solution aims to minimize the total operational cost including the vehicle and drone costs to provide maximum profits to both the logistics company and individual AV owners. Considering a large number of AVs with heterogeneous characteristics such as the available time frame, loading capacity, and fuel efficiency, the proposed solution simultaneously optimizes the scheduling and routes of AVs to serve customers in collaboration with drones by taking into account the real-time traffic conditions. We formulate A-VRPD as a mixed integer linear program (MILP) and propose an optimization framework to effectively solve the problem. To enhance the scalability for large-scale delivery scenarios, a greedy algorithm is also proposed. A novel tree-based cost-computation algorithm is designed to maximize profits based on parameters such as the real-world operational costs for different types of AVs, expected traveling distances and times calculated using Google Map API, varying load capacities of AVs, and available operation times of AVs.

Extensive simulations are performed to evaluate the effectiveness of the proposed solution. Numerous random delivery scenarios and varying numbers of available AVs are considered to validate the performance, in comparison with two state-of-the-art VRP-D heuristic solutions [33], [34] and a 2E-VRP algorithm [35]. The results demonstrate that the proposed solution significantly reduces the running time at the cost of relatively small performance degradation compared to the state-of-the-art algorithms. We also demonstrate that the zero fixed cost (wages) for A-VRPD leads to a significant amount of profits that can be shared between the delivery company and individual AV owners. The contributions of this article are summarized as follows.

- To the best of our knowledge, we are the first to explore a next-generation vehicle routing problem that integrates AVs to entirely automate the drone-based last-mile delivery process.
- We formulate the A-VRPD as a MILP, thoroughly accounting for its differences compared to the traditional VRP-D, with the goal of minimizing operational costs through the simultaneous optimization of AV scheduling and routing.
- A novel greedy heuristic solution is proposed to solve A-VRPD targeting large-scale delivery scenarios.
- Extensive simulations are conducted under various random delivery scenarios to demonstrate that the proposed solution significantly reduces the running time while producing large profits for both the logistics company and AV owners compared with state-of-the-art VRP-D and 2E-VRP solutions.

In Section II, we review the related work concentrating on drone-based last-mile delivery. The system model and notations used in this article are presented in Section III. We then formulate the A-VRPD problem as an ILP in Section IV and present the details of the greedy algorithm in Section V. The simulation results are analyzed in Section VI, followed by the conclusion of this work in Section VII.

II. Related Work

While the vehicle routing problem (VRP) [44] has been extensively researched due to its relevance across various domains, including berth allocation [45] and machine scheduling [46], this section primarily offers a comprehensive review of diverse strategies employed in drone-based last-mile delivery, particularly highlighting the most recent solutions for VRP-D. Murray and Chu was the first to formally define the problem of drone-truck collaboration for parcel delivery [9]. In their pioneering work, two different problems were introduced: the Flying Sidekick Traveling Salesman Problem (FSTSP) where a single vehicle delivers parcels in collaboration with a single drone, and the Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP) where a vehicle and a fleet of drones deliver parcels separately from the depot.

We first review numerous variants of FSTSP. One of the widely researched one is the traveling salesman problem with drones (TSP-D). Various heuristics have been proposed to solve TSP-D [13], [14], [15], [16], [17], [18], [19].

Schermer et al. focused on the scalability of the problem and proposed two heuristic solutions, *i.e.,* the two-phase heuristic (TPH) and single-phase heuristic (SPH) [17]. Bouman et al. proposed a dynamic programming approach to solve TSP-D [18]. Tang et al. created a constraint programming approach to solve the problem [19]. Poikonen et al. developed four branch-and-bound-based heuristics [47].

Variants of TSP-D have also been investigated. Ha et al. addressed TSP-D focusing on the operational cost [15]. Jeong et al. took into account the power consumption of drones and the restricted flying areas of the drones [48]. Dukkanci et al., similar to [15], aimed to minimize the operational cost considering the energy consumption of drones [49]. Wang et al. developed a multi-objective version of TSP-D to optimize both the operational cost and the time required to serve all customers [50]. Nonetheless, dependence on a single drone and a truck for delivery can lead to challenges such as limited delivery capacity and coverage as well as extended delivery time [24], which can result in higher costs per delivery location [25].

Wang et al. generalized TSP-D and introduced the vehicle routing problem with drones (VRP-D) [23]. Compared to other solutions, multiple trucks and multiple drones are considered assuming that drones can be launched from a truck at any of the customer locations and the base station. Poikonen et al. improved Wang's work by considering the limited battery of drones, varying distance metrics, operational cost in the objective function [24]. Sacramento et al. incorporated the time-limit constraint in the objective function and proposed an adaptive large neighborhood search metaheuristic [26]. Schermer et al. adopted sets of valid inequalities to improve the performance [25]. Kitjacharoenchai et al. developed a solution for the problem to address the limitations of the drone-launch and delivery time [27]. Murray et al. proposed a solution for an arbitrary number of heterogeneous drones for a truck with specific emphasis on real-world issues [51].

Since our work is directly related to VRP-D, we give a special emphasis on reviewing the details of latest solutions for VRP-D (published since 2022). Table I summarizes characteristics of those solutions in comparison with our work. Kuo et al. [34] study an extension of VRP-D by taking into account the constraint of customer time windows. More specifically, each customer is associated with a time window and should be visited by either a vehicle or a drone within the time window. A variable neighborhood search heuristic is proposed to solve the problem. Wang et al. [36] investigate VRP-D considering road traffic conditions, which is called the truck–drone hybrid routing problem with time-dependent road travel time drones (TDHRP-TDRTT). An iterative local search algorithm is developed to minimize the total distribution cost. Gu et al. [37] focus on the limitation of VRP-D that a drone visits only a single customer per trip. They aim to improve the practicality of the problem by allowing drones to visit multiple customers per trip before the drone returns to the truck where it was launched. Huang et al. [38] develop an ant colony optimization (ACO) algorithm to solve VRP-D while no significant modification is made to traditional VRP-D formulation. Nguyen et al. [39] extend the problem by adding two additional constraints. First, the total weight of parcels is considered in determining whether the capacity of a vehicle is exceeded. Second, the total working time of a vehicle and a drone is considered to ensure that each vehicle (both trucks and drones) do not operate longer than a pre-defined value. Rave et al. [40] incorporate specific locations called micro depots where drones can be launched. An adaptive large neighborhood search algorithm is designed to solve the extended version of VRP-D. Montaña et al. [41] note that there does not exist an analysis of the impact of parcel delivery using drones on sustainability and carbon emission reduction and analyze the efficiency of drone-based delivery for reducing carbon emissions. Sitek et al. [42] consider an extended VRP-D, called the extended vehicle routing problem with drones (EVRP-D) where mobile points (mobile hubs) are deployed for drone take-offs. A genetic algorithm is designed to optimize the cost as well as selection of mobile hubs. Wu et al. [43] develop an improved variable neighborhood decent algorithm to solve VRP-D in consideration of the impact of the payload and flight time of a drone on energy consumption.

While recent VRP-D solutions demonstrate remarkable performance in determining optimal delivery routes for trucks and drones, there are some limitations in their application towards AV routing with drones. One primary limitation of these solutions is their underlying assumption that trucks are involved in delivery, which consequently leads to increased costs due to the need for employing drivers. Moreover, existing solutions face challenges in terms of scalability due to their limitations in the number of trucks and customers that can be accommodated. The scalability issue is expected to become more pronounced as the solution incorporates the current traffic conditions. Additionally, current solutions do not fully take into account the individual properties of trucks, such as varying load capacities, fuel levels, consumption rates, vehicle types, and initial locations, which is crucial for the crowd-sourcing-based approach that involves heterogeneous AVs working in collaboration with drones.

III. Preliminaries

In this section, we present basic definitions, notations and assumptions related to AVs, customers, and drones. Fig. 2 can be referred to throughout this section to get a better grasp of the overall operation of the proposed approach.

A basic mechanism is that an AV loaded with drones traveling to a designated location, referred to as the "waiting location," where it can park and launch its drones to deliver parcels to customers. The proposed approach is a crowd sourcing-based delivery system where any AV owners can participate to make profits. When participating for the first time, an AV is directed to the nearest depot, where a readily deployable automatic battery replacement system is installed such as [52], [53], [54], and [55] (*e.g.,* Fig. 3). The AV owner is also expected to conveniently install the battery replacement system on their vehicle's roof-mounted carrier or hitch-mounted cargo carrier (*e.g.,* Fig. 3), as it operates as a separate system powered by its own battery, eliminating the

TABLE I
THE PUBLICATIONS RELATED TO VRP-D (PUBLISHED SINCE 2022)

References	Driver	Avail time	Parcel capacity	Type	Begin	End	Max Customers	Heuristic Methods	Opt. Objective	Characteristics	Opt. method
Kuo *et al.* [34]	yes	no	yes	no	Depot	Depot	50	Variable neighborhood search	Fuel cost	Customer time windows	MILP
Wang *et al.* [36]	yes	no	yes	no	Depot	Depot	200	Iterated local search	Fixed cost, drone cost, truck-drone coordination cost, truck cost	Time-dependent travel times	Not specified.
Gu *et al.* [37]	yes	yes	yes	no	Depot	Depot	200	Iterative local search	Fixed cost, total operation duration	Deployed drones capable of visiting multiple customers	MILP
Huang *et al.* [38]	yes	no	yes	no	Depot	Depot	200	Ant colony optimization	Fixed cost, travel cost for trucks and drones	Similar to traditional VRP-D formulation	MILP
Nguyen *et al.* [39]	yes	yes	yes	no	Depot	Depot	400	Slack induction by sweep removals	Total operational cost incurred by trucks and drones	Available time and capacity of trucks	MILP
Rave *et al.* [40]	yes	yes	yes	no	Depot	Depot	200	Adaptive large neighborhood search	Fixed cost, traveling cost, waiting cost, service cost	Dedicated drone stations (microdepots)	MILP
Montana *et al.* [41]	yes	no	yes	no	Depot	Depot	25	No heuristic	Total emissions generated by both trucks and drones	Focused on reduction of carbon emissions	MILP
Sitek *et al.* [42]	yes	no	no	no	Depot	Depot	100	Dedicated genetic algorithm	Fixed cost, operational cost, number of mobile points	Mobile points (mobile hubs) used for drone take-offs	Constraint satisfaction problem (CSP)
Wu *et al.* [43]	yes	no	no	no	Depot	Depot	199	Improved variable neighborhood descent	Delivery time	The drone payload and flight time	MILP
This paper	no	yes	yes	yes	Any	Any	500	Greedy	Total operational cost for AVs and drones	Autonomous vehicles	MILP

need for intricate connections to the AV. A logistics company offers the minimum amount of compensation m to attract participants and runs the proposed solution to select AVs to use for delivery. Assume that there are N_v AVs who wish to participate denoted by a set $V = \{v_1, v_2, \ldots, v_{N_v}\}$. Types of AVs differ, *i.e.,* sport utility vehicles (SUVs), trucks, and passenger vehicles, which can be easily extended to include more AV types. To calculate the total operational cost accurately, we separately define the cost for using AV v when it is mobile, denoted by c_v^M, and when it is stationary, denoted by c_v^S. These mobile and stationary costs vary depending on AV types. The load capacity of AV v denoted by q_v represents the number of parcels that can be loaded on the AV, which is also different depending on AV types. Another important property of AV v is the remaining fuel (or electricity for electric vehicles), which is denoted by f_v^A. We note that different types of AVs have different fuel consumption rates. The fuel consumption rate for AV v for the mobile and stationary modes are denoted by f_v^M and f_v^S, respectively. AVs can only be utilized when they are available. Therefore, the available time frame for AV v is denoted by τ_v^A (usually provided by the AV owner). We also assume that each AV has the maximum number of battery swap operations denoted by P due to the limited number of batteries it can load.

Next let us present notations and assumptions related to customers. There are N_l customer locations denoted by $L = \{l_1, l_2, \ldots, l_{N_l}\}$. Parcels are delivered to these customer locations from a distribution center (also called as the depot). These customers are organized into N_g groups denoted by $G = \{g_1, g_2, \ldots, g_{N_g}\}$. The group membership of each customer is represented by a $N_l \times N_g$ matrix $\mathbf{M}^{LG}$, where $\mathbf{M}^{LG}_{i,j} = 1$ means that customer $i \in L$ belongs to group $j \in G$. We assume

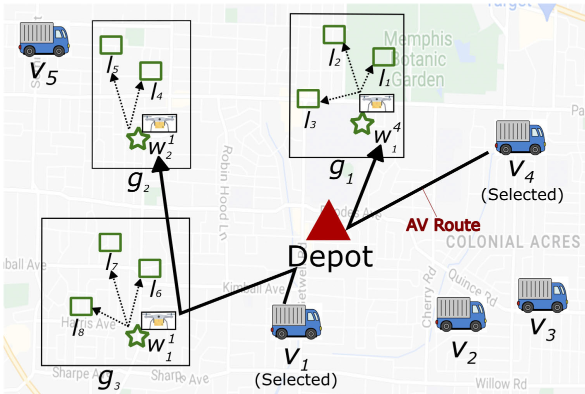


Fig. 2. An example operation of the proposed solution for A-VRPD.

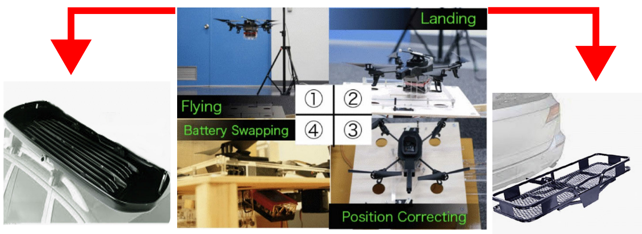


Fig. 3. An example of an automated drone battery replacement platform [52] and potential methods for mounting the system onto a vehicle.

that the delivery area is divided into zones as practiced by many logistics companies [56]. A similar zoning approach is used in creating the customer groups. More specifically, customers are first organized into different groups based on the company's zoning policy. Subsequently, each group is further divided by applying a maximum limit of battery swap operations per group, denoted by p. If the customers in a group cannot be serviced within p battery swap operations, the group is divided into smaller ones, each of which can be served within p battery swap operations. This parameter p is used by the company to control the group size. Every group has a designated "waiting location" where an AV is parked and deploys its drone to deliver parcels to the customers in that specific group. In particular, w_j^i denotes the waiting location for the j-th group in the sequence of groups served by AV $i \in V$.

Now we explain notations and assumptions pertaining to drones. Note that AVs do not deliver parcels, and only drones perform delivery. The proposed solution schedules AVs to serve a set of customer groups. An AV $i \in V$ drives to a waiting location w_j^i for serving its j-th group $i \in G$, parks there, and launches its drone. The drone is supposed to cover all customers in that specific group. A drone can deliver multiple parcels at a single trip. However, for simplicity, we assume that a drone delivers one parcel per trip. When the battery of the drone is depleted, it is automatically swapped. The drone operation cost denoted by c_d (dependent on the traveled distance of a drone) is defined as a factor of the mobile cost for AV. Following the method used in [34], in our work, we set $c_d = 1/10c_v^M$. Since the waiting location and customer locations are all known for each group, the total

TABLE II
THE LIST OF NOTATIONS

Symbol	Description
$V = \{v_1, v_2, ..., v_{N_v}\}$	The set of available AVs
c_v^M	The operational cost of AV $v \in V$ in the mobile mode (per distance)
c_v^S	The operational cost of AV $v \in V$ in the stationary mode (per time)
q_v	The load capacity of AV $v \in V$
τ_v^A	The available time for $v \in V$
f_v^M	The fuel consumption rate of AV $v \in V$ (mobile)
f_v^S	The fuel consumption rate of AV $v \in V$ (stationary)
m	The minimum amount of compensation for each AV
P	The maximum number of drone battery swap operations for each AV
$L = \{l_1, l_2, ..., l_{N_l}\}$	The set of customers
$G = \{g_1, g_2, ..., g_{N_g}\}$	The set of customer groups
$\mathbf{M}^{LG}$	The matrix that defines the group membership of each customer
w_j^i	The waiting location for the j-th customer group in the sequence of groups served by AV $i \in V$
p	The maximum number of battery swap operations allowed per group
$\mathbf{M}^{VG}$	The matrix that defines a mapping between AVs and customer groups
c_d	The drone operational cost (per distance)
c_d^T	The total drone operational cost
τ_g^D	The expected delivery finish time for drones(s) to cover customer group $g \in G$
b	The available budget

operational cost for drones c_d^T can be easily pre-computed, which allows for significant reduction of running time (up to 350X faster running time is possible in some instances, as it will be demonstrated in Section VI-E), making it extremely extensible for a large number of customers. Similarly, the expected time for a drone to finish delivery for all customers in a particular group $g \in G$ denoted by τ_g^D can also be easily pre-computed. Table II summarizes all notations introduced in this section.

IV. Last-Mile Delivery Using Autonomous Vehicles With Drones

This section presents an overview of the proposed approach, followed by descriptions of the pre-computation phase and mathematical model.

A. Overview

Our objective is to choose a subset of participating AVs, represented by $\overline{V}$, from the available AVs V ($\overline{V} \subseteq V$). Each vehicle $v \in \overline{V}$ is assigned to cater to one or more groups in G following a designated order (indicating the AV's route). The aim is to minimize the overall operational cost of delivering parcels to all customers, adhering to a given budget constraint b. We propose a two-phase solution that concurrently optimizes the allocation of AVs to customer groups and the routes of chosen AVs in order to minimize the overall operational cost. More specifically, a pre-computation phase is introduced in Section IV-B to address the scalability issue for existing VRP-D solutions, especially considering a large number of AVs and customer locations. The results of the pre-computation phase are provided as input to our mathematical model to perform the AV-to-group mapping and computation of the optimal AV routes. The details of the mathematical model are presented in Section IV-C.

B. Pre-Computation Phase

A significant challenge for VRP-D is its limited scalability, stemming from the intricate nature of the problem space. Many heuristic solutions for VRP-D support a limited number of customers due to their extensive running time. Reducing the running time is especially crucial for A-VRPD, as it involves a greater number of AVs and requires rapid updates to accommodate real-time traffic conditions while AVs and drones are serving customers. For example, based on current traffic conditions and the inclusion of newly added AVs, a vehicle may be quickly reassigned to cater to a customer group that differs from its original schedule.

To tackle the scalability issue, we introduce a pre-computation phase. In particular, this phase exploits the distinct features of A-VRPD to considerably decrease running time, *i.e.,* AVs do not engage in parcel delivery but simply wait as drones carry out the deliveries to customers. Therefore, the route of an AV can be simplified into a sequence of the waiting locations that the AV visits. Additionally, in A-VRPD, the drone path can be pre-determined because (1) customer groups are determined by the company's zoning policy and the maximum allowable battery swap operations for each group, and (2) drones are tasked with delivering parcels exclusively to customers within a specific group. As we will show more details in Section VI-E, the pre-computation phase enables by up to 350X faster running time in some instances compared to state-of-the-art VRP-D solutions, making frequent solution update possible to account for real-time traffic conditions.

We now present the details of the pre-computation phase. The total operational cost consists of the cumulative distance d_v^T traveled by each selected AV v and the total time τ_v^T required for the journey, which includes the waiting time while the AV's drone serves customers. The pre-computation phase is designed to reduce the computational delay for calculating d_v^T and τ_v^T by pre-calculating distance and time segments constituting d_v^T and τ_v^T.

Here we present the details on how distance segments constituting d_v^T are pre-computed. Let us denote the length of a route between two locations i and j by $d_{i,j}$. The length of a route is obtained using the Google MAP API to take into account the real-time traffic conditions. We compute all distance segments between two "essential" locations: the distance (1) between the depot and a waiting location, (2) between two waiting locations, and (3) between the depot and the AV's original location. The total distance d_v^T for AV v that is scheduled to visit a sequence of waiting locations $\{w_1, \ldots, w_n\}$ can then be computed using the pre-computed distance segments as follows.

$$d_v^T = d_{v,f} + d_{f,w_1} + \sum_{k=1}^{n-1} d_{w_k,w_{k+1}} + d_{w_n,f} + d_{f,v},$$

This implies that an AV v initially proceeds to depot f for loading parcels and a drone, after which it visits a series of waiting locations $w_1, \ldots, w_n$ to cater to customers. Upon completion of service to all customers, the AV returns to the depot and then proceeds to its original location.

Next, we explain how the time segments constituting the total amount time for AV ($\tau_{i,j}$) are pre-computed. Let us denote the travel time between two locations i and j by $\tau_{i,j}$. We calculate the travel time using the Google MAP API. The total travel time for AV (τ_v^T) is then calculated based on the pre-computed time segments as the following.

$$\tau_v^T = \tau_{v,f} + \tau_v^L + \tau_{f,w_1} + \sum_{k=1}^{n-1} \tau_{w_k,w_{k+1}} + \tau_{w_n,f} + \tau_{f,v} + \tau_v^W.$$

Note that for a more precise representation of total time, we consider the time needed to load parcels and drones onto the AV, denoted by τ_v^L. Additionally, we account for the cumulative waiting time of an AV while its drones complete deliveries, represented by τ_v^W.

C. Mathematical Model

$$\begin{aligned}
\underset{\mathbf{M}'^{VG}}{\arg\min} \quad & \sum_{v\in\overline{V}} (c_v^M \cdot d_v^T + c_v^S \cdot \tau_v^W) & \\
\text{s.t.} \quad & \sum_{i\in V} \mathbf{M}_{i,j}^{VG} = 1, \quad \forall j \in G & (1)\\
& \sum_{i\in\overline{V}} (c_i^M \cdot d_i^T + c_i^S \cdot \tau_i^W) + c_d^T \le b & (2)\\
& \sum_{j\in G} (\mathbf{M}_{i,j}^{VG} \cdot \sum_{k\in L} \mathbf{M}_{j,k}^{LG}) \le q_i, \quad \forall i \in V & (3)\\
& (f_v^M \cdot d_i^T + f_v^S \cdot \tau_i^W) \le f_v^A, \quad \forall i \in V & (4)\\
& \tau_i^T \le \tau_i^A, \quad \forall i \in V & (5)\\
& \tau_i^W = \sum_{j\in G} \mathbf{M}_{ij}^{VG} \cdot \tau_j^D, \quad \forall i \in V & (6)\\
& [b - \sum_{i\in\overline{V}} (c_i^M \cdot d_i^T + c_i^S \cdot \tau_i^W)]/2 > m \cdot |\overline{V}| & (7)\\
& p \cdot \sum_{j\in G} \mathbf{M}_{i,j}^{VG} \le P, \quad \forall i \in V & (8)\\
& w_k^i = j, \mathbf{M}'^{VG}_{i,j} = k, \quad \forall i \in V, \ \forall j \in G & (9)\\
& 1 \le k \le n_i, \quad \forall i \in V & (10)\\
& n_i = \sum_{j\in G} \mathbf{M}_{ij}^{VG}, \quad \forall i \in V & (11)\\
& d_i^T = d_{i,f} + d_{f,w_1^i} + \sum_{k=1}^{n_i-1} d_{w_k,w_{k+1}^i} & \\
& \qquad + d_{w_{n_i}^i,f} + d_{f,i}, \quad \forall i \in V & (12)\\
& \tau_i^T = \tau_{i,f} + \tau_i^L + \tau_{f,w_1^i} + \sum_{k=1}^{n_i-1} \tau_{w_k,w_{k+1}^i} & \\
& \qquad + \tau_{w_{n_i}^i,f} + \tau_{f,i} + \tau_i^W, \quad \forall i \in V & (13)\\
& \mathbf{M}_{i,j}^{VG} = 1, \mathbf{M}'^{VG}_{i,j} > 0, \quad \forall i \in V, \ \forall j \in G & (14)
\end{aligned}$$

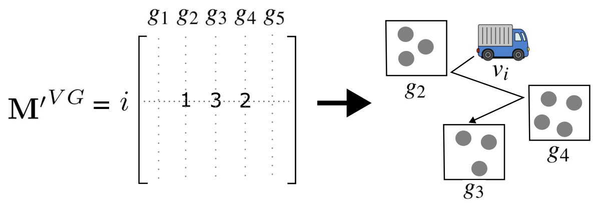


Fig. 4. An example solution for AV v_i scheduled to cover three groups in the order of g_2, g_4, and then g_3.

In this section, we present a mathematical model to solve A-VRPD based on the distance and time segments derived from the pre-computation phase. In this model, we aim to minimize the total operational cost that is the sum of individual vehicle costs for the selected AVs. This per-vehicle cost for AV v is computed based on the total distance it has traveled (d_v^T) and the accumulated waiting time while its drone serves customers (τ_v^W); Thus, the per-vehicle cost is $c_v^M \cdot d_v^T + c_v^S \cdot \tau_v^W$. Note that τ_v^W is included in the per-vehicle cost because it reflects the AV's energy consumption in the idle mode to support the operation of its drone, *i.e.,* to run the system to communicate with the drone, check the battery level of the drone, and run the automated battery replacement system. Therefore, the total cost for all participating AVs is $\sum_{v \in \overline{V}} (c_v^M \cdot d_v^T + c_v^S \cdot \tau_v^W)$. It should also be noted that the drone cost is pre-computed in the pre-computation phase and is added to calculate the final total cost.

We formulate our mathematical model as a MILP. The MILP formulation is depicted above. As shown, the objective function is to find matrix $\mathbf{M}'^{VG}$, which defines the mapping between AVs and customer groups as well as a sequence of groups to be covered by each AV (*i.e.,* representing the route of each AV), to minimize the total cost. Consider an example solution shown in Fig. 4 for AV v_i to better explain $\mathbf{M}'^{VG}$. The solution indicates that v_i is scheduled to visit three groups in the order of g_2, g_4, and then g_3.

Constraint (1) ensures that each group is covered by only one AV, and all groups are covered. Constraint (2) specifies that the total cost, which includes the mobile and stationary expenses for AVs as well as the drone-related costs, must not exceed the available budget b. Constraint (3) dictates that the number of customer locations covered by an AV should be smaller than the AV's capacity. Since $\mathbf{M}_{i,j}^{VG} = 1$ when group $j \in G$ is covered by AV $i \in V$, if we multiply $\mathbf{M}_{i,j}^{VG}$ by $\sum_{k \in L} \mathbf{M}_{j,k}^{LG}$ (*i.e.,* the total number of customer locations of group j), we obtain the total number of customer locations of the group covered by an AV $i \in V$. If we repeat this computation for all groups, we get the total number of customer locations covered by an AV v, which should be smaller than its capacity. Constraint (4) is used to ensure that each AV has enough gas to serve all customer locations assigned to it. Constraint (5) enforces that the total operation time of an AV does not exceed the available time of the AV. This means that an AV should return to its original location before its available time is expired. Constraint (6) is an equality constraint that defines the total amount of time that an AV has waited for its drones to complete delivery. Constraint (7) ensures that each AV should receive a compensation greater than or equal to the minimum amount of compensation m. Constraint (8) ensures that each AV can perform at most P battery swap operations for its drone. Here, p is the maximum number of battery swap operations defined for each group. Constraints (9) and (10) are used to compute the k-th customer group to visit ($1 \leq k \leq n_i$), where n_i is the total number of customer groups that AV v_i is scheduled to visit. More precisely, $\mathbf{M}'^{VG}_{i,j} = k$ implies that group j is the k-th group visited by AV i. Thus, constraint (9) indicates that, if $\mathbf{M}'^{VG}_{i,j} = k$, the waiting location for the k-th group served by AV i should be the one assigned to group j. Constraint (11) defines an equality constraint used to compute n_i. Constraints (12) and (13) compute the total cost d_i^T and the total time τ_i^T for each AV $i \in V$, respectively. Constraint (14) is used to derive $\mathbf{M}^{VG}$ from $\mathbf{M}'^{VG}$ where each element of $\mathbf{M}^{VG}$ represents whether the corresponding customer group has been served or not, *i.e.,* $\mathbf{M}_{i,j}^{VG} = 1$ means that group $j \in G$ is covered by AV $i \in V$.

V. Greedy Approach

As A-VRPD is an NP-hard problem, we propose a greedy algorithm to substantially reduce running time for large-scale delivery scenarios. The fundamental principle of the greedy algorithm is to iteratively select a participating AV that serves the maximum number of customer locations at the lowest possible cost, until all customer locations are covered or the minimum compensation constraint is breached. The greedy algorithm comprises two phases: tree-based computation of the per-AV cost, and the selection of an AV associated with the minimum cost. These two phases are iterated until a set of AVs serving all customers is identified or the minimum compensation constraint is violated. The specifics of these two phases are presented below.

A. Tree-Based Computation of Per-AV Cost

The first phase of the greedy algorithm involves calculating the per-AV cost for each vehicle by constructing a tree data structure. To illustrate how the per-AV cost is computed, consider the example in Fig.5, which depicts a tree built for AV v. In this example, the tree's root represents the distribution center (also known as the depot). Each node in the tree's first level signifies a customer group that can be served by the AV directly from the distribution center. Specifically, a customer group is deemed serviceable by an AV if the number of customer locations within the group is less than or equal to the current number of parcels loaded onto the AV (assuming one parcel delivery per customer), and the maximum number of battery swap operations is not surpassed (i.e., $p \cdot \sum_{j \in \widehat{G}} \mathbf{M}_{i,j}^{VG} \leq P$ for vehicle v_i, where $\widehat{G} \subseteq G$ is the set of customer groups covered so far). The greedy algorithm permits partial coverage of a customer group by defining a coverage ratio, with a default of 100%. As illustrated in Fig.5, AV v has a load capacity of 6, and customer group g_1 can be served by AV v as it consists of only 2 customers.

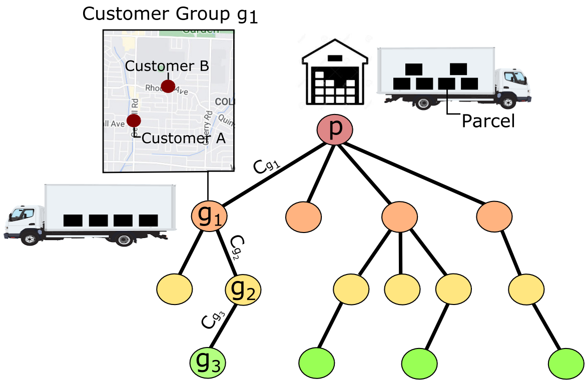


Fig. 5. An illustration of a tree structure constructed by the greedy algorithm for each AV.

The procedure for constructing nodes in the first level is replicated for subsequent levels of the tree. Specifically, we identify customer groups that can be served by AV v from each node in the first level. It is important to note that the number of parcels loaded on AV v decreases as the AV has already covered a customer group in the previous level. Additionally, fully covered customer groups are excluded. For instance, as demonstrated in Fig. 5, when building nodes for the second level from node g_1, the number of parcels loaded on AV v is reduced to 4, since the AV has served the two customers in g_1. Furthermore, we ensure that g_1 is not used to construct nodes in the second level, as it has already been completely covered. The greedy algorithm persists in building nodes in subsequent levels until all parcels loaded on the AV have been delivered to customers.

Upon constructing the tree for AV v, we can compute the per-AV cost for v. Specifically, each edge in the tree is associated with the cost of covering a customer group. For instance, as depicted in Fig. 5, the edge connecting nodes p and g_1 is associated with the cost c_{g_1} to cover customer group g_1. More precisely, the cost to cover customer group j from the current location i is defined as $(d_{i,j} c_v^M + \tau_j^D c_v^S)/|j|$, where $|j|$ represents the number of customers belonging to customer group j (i.e., the group's size). The greedy algorithm prioritizes customer groups with more customers when the traveling distance is equal, by dividing the group's coverage cost by its size. As a result, we can calculate the sum of edge weights along a simple path between the root and a terminal node. The per-AV cost corresponds to the smallest cost of simple paths to all terminal nodes. Notably, the simple path with the lowest cost represents the sequence of customer groups visited by the AV.

B. Selection of AV

Given the per-AV costs of all AVs, the next step of the greedy algorithm is to select an AV with the minimum per-AV cost. Algorithm 1 summarizes how an AV is selected in a greedy fashion. Lines 3-7 indicate the process of building the tree and computing the per-AV cost for each AV. Line 9 shows that the greedy algorithm finds an AV with the minimum per-AV cost. Once such an AV with the minimum per-AV cost is found, the number of participating AVs n_v is incremented by one (Line 10), and the customer groups covered by the AV are excluded (Line 12). Also, the total cost c_f is updated by adding the per-AV cost of the selected AV (Line 13). The above-mentioned process is repeated until all customer groups are covered or the minimum compensation constraint (i.e., $[b - c_f]/2 > m \cdot n_v$) is violated. (Line 2).

Algorithm 1 The Greedy Algorithm

1 $c^f \leftarrow 0$
2 **while** $|G| > 0$ *and* $[b - c_f]/2 > m \cdot n_v$ **do**
3   **for** *each AV* $v \in V$ **do**
4     $\mathbb{T}_v \leftarrow \text{BuildTree}(v)$
5     $c_v \leftarrow \text{ComputePerAVCost}(\mathbb{T}_v, v)$
6   // C_V is a set of per-AV costs
7     $C_V \leftarrow C_V \cup c_v$
8   // Find $v' \in V$ with the minimum cost using C_V
9   $v' \leftarrow \text{FindAVMinCost}(C_V)$
10   $n_v = n_v + 1$
11   // $G_{v'}$ is a set of customer groups covered by AV v'
12   $G \leftarrow G \setminus G_{v'}$
13   $c^f \leftarrow c^f + c_{v'}$
14   $C_V \leftarrow \emptyset$

There are noteworthy aspects of the greedy algorithm worth mentioning. The algorithm permits the same AV to be selected multiple times. We observe that the greedy algorithm tends to choose the same vehicle when there are insufficient available AVs. One important consideration is that when the same AV is selected, we ensure that the additional cost for the AV to return to the facility center to reload parcels is incorporated into the per-AV cost. Additionally, the greedy algorithm accounts for the traveling costs of moving from an AV's original location to the distribution center, as well as the cost of traveling from the final customer group back to the original location. This ensures a more accurate calculation of the per-AV cost.

VI. Computational Results

In this section, we evaluate the performance of the proposed approach. The MATLAB optimization toolbox [57] is used to implement the optimization framework for A-VRPD and the proposed greedy algorithm. In particular, the mixed-integer linear programming (MILP) solver is used. For performance comparison, we implement two state-of-the-art VRP-D algorithms denoted by SoA1 [33] and SoA2 [34], respectively. Specifically, SoA1 allows a drone to return to a vehicle, which is different from the one where it was launched, to swap its battery and/or to pick up parcels. We implement their adaptive multi-start simulated annealing (AMS-SA) metaheuristic algorithm. SoA2 extends a traditional VRP-D formulation by considering the customer time window. We implement their variable neighborhood search (VNS) heuristic algorithm while relaxing the time window constraint to ensure fairness in performance comparison.

We are also aware of a closely-related problem called the Two Echelon Vehicle Routing Problem (2E-VRP). In 2E-VRP, the logistics network consists of two echelons. More

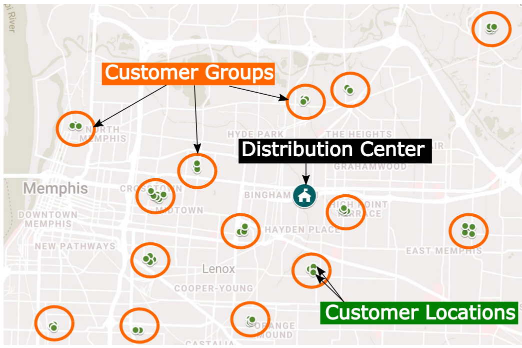


Fig. 6. An area in the city of Memphis where the simulation study is performed. The Google Map API is used to calculate the trajectories of vehicles to serve the customer locations.

specifically, parcels are delivered from the depot to the satellites in the first echelon. And then, the parcels are delivered from the satellites to customers in the second echelon. The objective of 2E-VRP is to optimize the total transportation cost. Although A-VRPD has some common aspects compared to 2E-VRP, there are notable differences between the two: (1) In A-VRPD, AVs are selected based on heterogeneous characteristics of each AV such as varying capacities, initial locations, fuel level, available time. Such a vehicle selection mechanism does not exist in 2E-VRP; (2) In A-VRPD, a drone is used to serve customers, and therefore, the routing path in the second echelon for VRP-D can be significantly simplified. On the other hand, in 2E-VRP, the 2nd-level trucks are used to serve customers; (3) In 2E-VRP, the start and end locations for trucks are fixed. The 1st-level trucks start and end at the depot. The 2nd-level trucks start and end at a satellite. On the other hand, in A-VRPD, each AV has individual start and end locations; (4) More importantly, in A-VRPD, vehicles do not deliver since they are autonomous vehicles. Therefore, A-VRPD does not incur the fixed cost, *i.e.,* wages for drivers, thereby creating extra profits that can be shared with AV owners. 2E-VRP does not have such a mechanism. With those key differences in mind, we implement a state-of-the-art heuristic algorithm for 2E-VRP [35]. In particular, we exclude the pickup demands for fair performance comparison. The proposed optimization framework, greedy algorithm, and the state-of-the-art heuristic algorithms are executed on a PC equipped with Intel Core i7-9750H and 16GB RAM.

Different delivery scenarios are used in this simulation study. More specifically, 10 random delivery scenarios are created. In each scenario, 500 customer locations are randomly selected. These customer locations are organized into 80 groups based on a grid zoning method and the minimum number of battery swap operations (*i.e.,* $p = 2$). It is noted that a different zoning method can be easily adopted such as [56]. Fig. 6 shows a sample delivery scenario with a depot, customer locations, and customer groups. Additionally, we consider different AV types including sedans (type 1), SUVs (type 2), and pickup trucks (type 3). We assume that AVs are uniformly distributed in the area. The default number of available AVs is set to 50.

Since the objective of the proposed approach is to minimize the total operational cost, the key metric for this simulation study is the amount of profits generated for both the delivery company and AV owners. Specifically, profits for AV owners are produced based on the 50:50 profit model that allows the delivery company and participating AV owners to equally share profits. Note, however, that any profit model can be easily adopted. In addition to measuring the cost, another main metric that we use in this simulation study is the running time. We measure these two metrics by varying delivery scenarios and the number of available AVs. Table III summarizes the default parameters used in this simulations study. More precisely, the fuel costs for different types of vehicles are determined based on real-world fuel consumption data [58]. The total budget b is determined based on a real-world salary for truck drivers [59]. More specifically, considering the fact that a truck driver makes approximately \$24 per hour and covers about 100 customers a day [60], [61], the per-customer cost for the delivery company is roughly \$2.4; Since there are 500 customers to serve, the budget b is set to \$1,200.

TABLE III
THE DEFAULT INPUT PARAMETERS USED FOR SIMULATION

Parameters	Values (type 1, type 2, type 3)
b	\$1,200
c_v^M	\$0.1, \$0.12, \$0.15
c_v^S	\$0.00013, \$0.00033, \$0.00071
f_v^A	13 gal, 15 gal, 23 gal
q_v	7, 9, 14
m	\$5
p	2
P	20
c_d	$0.1 \times c_v^M$

A. Cost Analysis

A notable advantage of A-VRPD is that it does not involve the fixed cost in terms of wages for drivers since AVs do not deliver parcels. On the other hand, VRP-D solutions produce more effective routes for both vehicles and drones since vehicles deliver parcels in tandem with drones. Therefore, performance degradation for A-VRPD in terms of effectiveness of vehicle route is inevitable. In this section, we evaluate the performance degradation compared with state-of-the-art VRP-D and 2E-VRP algorithms. More specifically, we measure the total cost based on the total traveled distance (*i.e.,* excluding the fixed cost) for our solution, SoA1, SoA2, and 2E-VRP under 10 random delivery scenarios.

Results are depicted in Fig. 7. As shown, the total cost for A-VRPD is higher than that for VRP-D algorithms. The reason is that the VRP-D algorithms produce more efficient routes for trucks and drones because trucks and drones can visit any customer location. In contrast, in A-VRPD, AVs are only allowed to visit the waiting locations of customer groups. In addition, our solution for A-VRPD includes the stationary cost which is induced while the AV is waiting for its drone to deliver (See Fig. 8 demonstrating that the stationary cost takes 21.1% of the total cost on average). On average, the total cost for A-VRPD (Greedy) is higher than that for SoA1 and SoA2 by 12.5% and 22.1%, respectively.

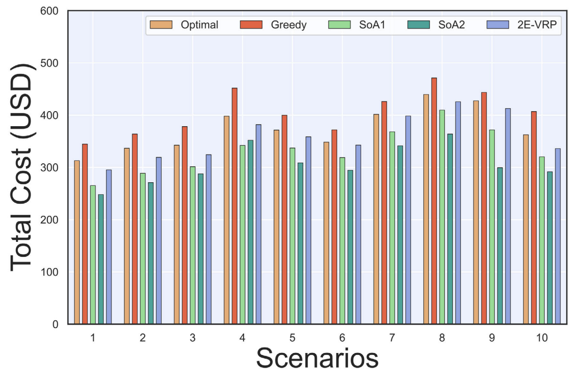


Fig. 7. The total operational cost for vehicles and drones excluding the fixed cost (wages).

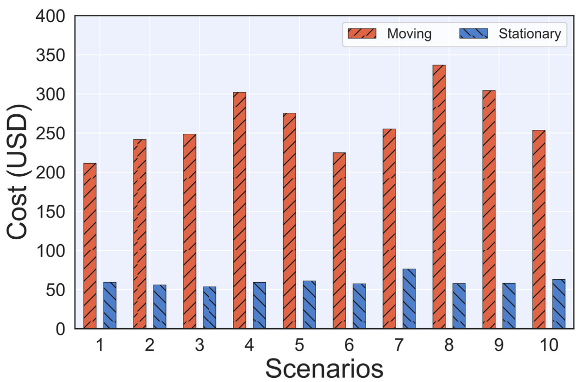


Fig. 8. The stationary cost for A-VRPD.

An interesting observation is the performance gap between the VRP-D algorithms and the 2E-VRP algorithm. Specifically, the total cost for the 2E-VRP algorithm is higher than that for SoA1 and SoA2 by 8.1% and 17.5%, respectively. A possible reason is that the 2E-VRP algorithm produces routes separately for the level-1 vehicles and level-2 vehicles, resulting in suboptimal routing paths compared to the VRP-D algorithms. Another reason is that the VRP-D algorithms use drones that have a smaller operational cost than trucks.

We then compare the total cost for our solution with the 2E-VRP algorithm. We observe that our solution generates a higher cost than that for the 2E-VRP algorithm despite the fact that out solution uses drones with a lower operational cost to serve customers, while, in the 2E-VRP algorithm, the 2nd-level trucks deliver parcels to customers. The reason for the degraded cost for A-VRPD is that AVs have a smaller load capacity depending on vehicle types; as such, they have to return to the depot to reload especially when there are not enough available AVs. Another reason is because the A-VRP solution requires AVs to move from their original locations to the depot, and then back to their original locations once delivery is completed, thereby increasing the cost.

Although our solution produces less efficient routes for vehicles and drones, it should be noted that the results in Fig. 7 do not account for the fixed cost (wages) which comprises a huge portion of the total operational cost. In the next section, we analyze the extra profits generated from the fact that the A-VRP algorithm does not incur the fixed cost.

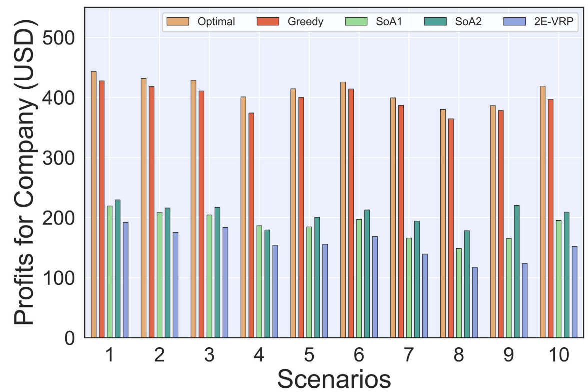


Fig. 9. The profits generated for the delivery company in 10 random delivery scenarios.

B. Profit Analysis

In this section, we analyze profits generated by the proposed optimal and greedy algorithms and compare with SoA1, SoA2, and 2E-VRP algorithms. The profit here is defined as the budget b subtracted by the total cost to serve all customers. In this experiment, the total cost for SoA1, SoA2, and 2E-VRP includes the fixed cost while our A-VRPD algorithms do not incur the fixed cost.

The profits for the delivery company under varying delivery scenarios are depicted in Fig. 9. The average profits for the greedy and optimal algorithms are $412.96 and $397.14, respectively. The optimal algorithm achieves 3.9% higher profits on average compared with the greedy algorithm. Such a small difference in earned profits between the two algorithms demonstrates the effectiveness of the greedy algorithm, especially considering the significantly faster running time of the greedy algorithm. More detailed experimental results on the running time are presented in Section VI-E.

We also measure the profits produced by SoA1, SoA2, and 2E-VRP algorithms and compare with that for our greedy algorithm. The results indicate that, on average, the greedy algorithm achieves higher profits by 111.7%, 93%, and 154.3% compared with that for SoA1, SoA2, and 2E-VRP algorithms, respectively. Overall, despite the performance degradation for A-VRPD in terms of the effectiveness of routes for vehicles and drones (as presented in Section VI-A), the proposed algorithms allow for significantly higher profits compared with state-of-the-art algorithms.

The profits for company accrued with the proposed approach can be used to provide compensations to AV owners. A logistics company can use these profits to attract more AVs to participate, potentially leading to higher profits. Fig. 10 depicts the results that the average profit for each AV owner is $8.26 and $7.94 per day when the optimal and greedy algorithms are used, respectively. A daily profit of $8.26 for AV owners translates into a monthly profit of about $247. It is worth to note that this is a net profit excluding all other costs for AV owners such as the fuel/mileage cost. Additionally, this profit is generated without involving the human labor of the AV owner at all! Another interesting aspect is that depending on the number of available AVs, this monthly profit

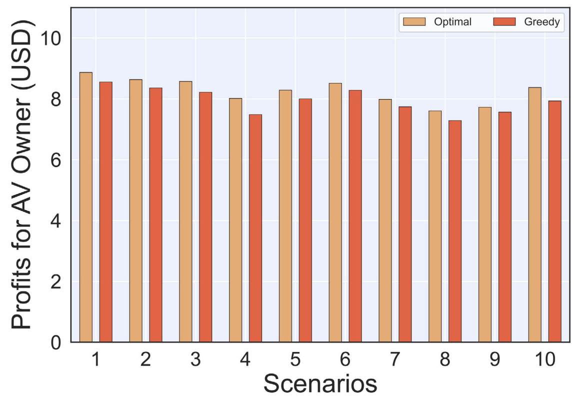


Fig. 10. The profits generated for AV owners under 10 different delivery scenarios.

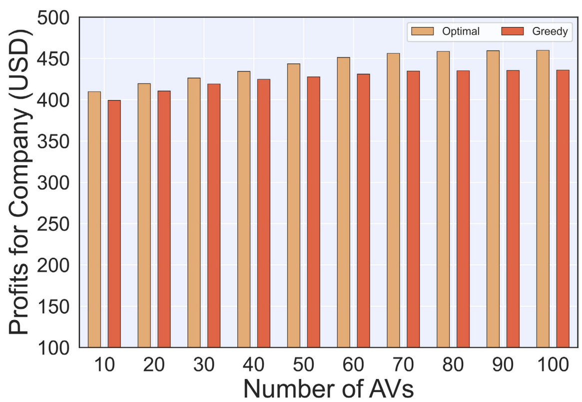


Fig. 11. The profits for the delivery company with varying numbers of available AVs under 10 different delivery scenarios.

can increase up to $1,229 per month under our scenarios. In the next section, we present an in-depth analysis of the effect of the number of AVs.

C. Number of Available AVs

The profits for the delivery company and AV owners depend on the number of available AVs. We evaluate the effect of the number of available AVs on the profits for the company and AV owners. The number of available AVs is varied from 10 to 100 with an interval of 10 to measure the profits.

Fig 11 depicts the profits for the delivery company. It is observed that the profits for company increase as the number of available AVs increases regardless of the algorithms. The reason for the lower profit with a smaller number of available AVs is because of the additional cost for AVs to return to the distribution center for reloading parcels. In contrast, such an additional cost can be saved when there is sufficiently large number of AVs available, thereby increasing the profits.

We then evaluate the effect of the number of available AVs on the profits for AV owners. Fig. 12 depicts the results. In contrast to the profits for the delivery company which increase as the number of AVs increases, the profits for AV owners become higher with a smaller number of AVs. These results demonstrate that serving more customers is more profitable for AVs despite the additional cost incurred for them to return to the distribution center for reloading parcels. Although AV owners receive higher profits when

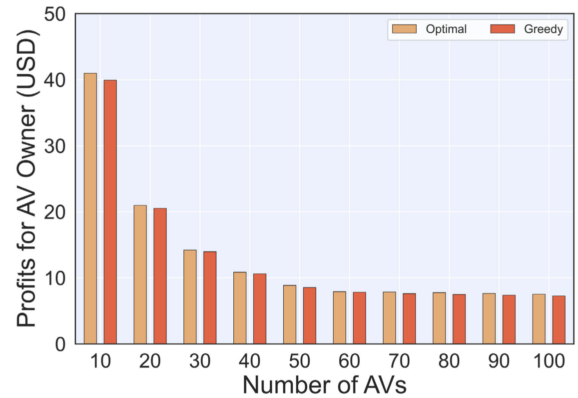


Fig. 12. The profits for participating AV owners with varying numbers of AVs under 10 different delivery scenarios.

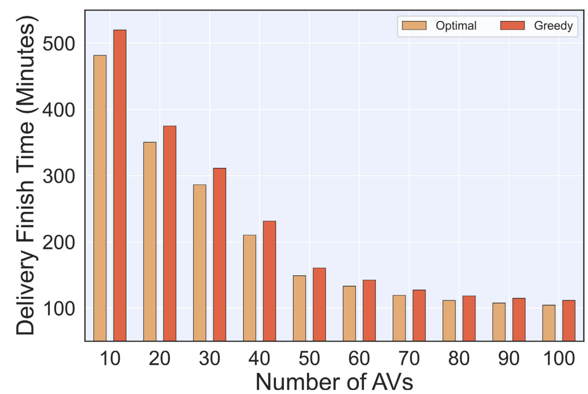


Fig. 13. The delivery finish time measured with varying numbers of AVs.

there the number of available AVs is smaller, a small number of AVs leads to increased delivery finish time. More details on simulation results in terms of the delivery finish time are presented in the following section. We also observe that the decreasing trend of profits flattens out as the number of available AVs increases. The reason for this observation is that AVs are recruited only within the available budget b; As such, not all AVs may be used even though they are available.

D. Delivery Finish Time

We evaluate the effect of the number of available AVs on the delivery finish time. Fig. 13 depicts the results, which demonstrate that the delivery finish time substantially increases as the number of AVs decreases. More specifically, the delivery finish time for 10 available AVs is increased by 360% on average, when the optimal algorithm is used, compared with that for 100 AVs. Similarly, the delivery time for 10 available AVs increases by 365% higher on average when the greedy algorithm is used, compared with that for 100 available AVs. The reason for the higher delivery finish time is attributed to the additional delay for AVs to return to the distribution center to reload parcels.

E. Running Time

One of the key benefits of the proposed solution is that it is designed to significantly reduce the running time by

TABLE IV
RUNNING TIME IN SECONDS

Methods	100 customers					500 customers				
	Scn 1	Scn 2	Scn 3	Scn 4	Scn 5	Scn 1	Scn 2	Scn 3	Scn 4	Scn 5
OPT	16.3	14.8	19.38	17.16	13.4	252.52	235.46	271.15	243.65	237.64
Greedy	0.96	0.88	1.05	1.03	0.79	6.21	5.34	6.87	5.67	4.86
SoA1	81.55	84.24	86.73	83.47	87.12	739.77	722.02	761.69	742.28	717.43
SoA2	356.25	369.64	406.72	383.18	361.43	1649.44	1566.23	1762.24	1704.51	1643.57
2E-VRP	71.35	79.2	84.67	81.66	75.45	547.56	521.23	612.96	598.45	577.04

simplifying the routes of AVs leveraging the fact that AVs do not deliver parcels, and also by pre-computing the drone routes through organization of customer locations into groups. In this section, we evaluate the running time of our approach compared with the SoA1, SoA2, and 2E-VRP algorithms in 5 different random delivery scenarios by varying the number of customers.

Results are presented in Table IV. We observe that the running time of both the optimal and greedy algorithms is substantially smaller compared with that for VRP-D heuristic algorithms. Specifically, it is notable to observe that the running time for our optimal algorithm is 3X and 13X faster than that for SoA1 and SoA2, respectively. The running time for our greedy algorithm is 107X and 342X faster compared with that for SoA1 and SoA2, respectively. The results demonstrate the impact of the simple routes of AVs since they do not deliver parcels and the effect of the proposed pre-computation phase on significantly reducing the running time. In comparison with the 2E-VRP algorithm, the optimal and greedy algorithms improve the running time by 2X and 89X, respectively. The reason for the improved running time can be attributed to the fact that the 2E VRP algorithm consumes much more time to simultaneously optimize the path for both the 1st and 2nd-level, and the customer-to-satellite mapping.

F. Benchmark Dataset

For more effective performance evaluation, in this section, a benchmark dataset adopted from a state-of-the-art solution for VRP-D [33] is used for profit analysis for our solutions, SoA1, SoA2, and 2E-VRP algorithms. The dataset is the modified version of the instances created by [62]. Among the instances consisting of 6 categories *i.e.,* C1, C2, R1, R2, RC1 and RC2, we adopt the RC2 instances for our experiments since this scenario has the most resemblance with our own dataset. In the RC2 instances, the customer coordinates are randomly produced and clustered in a grid region of $[100, 100]^2$. The maximum waiting time for drones T is set at 10 min. Since there are only 100 customer locations, we used the budget of \$240 (*i.e.,* 2.4*100) and the number of vehicles was set to 10.

Figs. 14 and 15 display the profits for the delivery company and AV owners, respectively. Interestingly, the results are very similar to what we obtained from the experiments performed with our own dataset. More specifically, the profits for the company obtained with the optimal algorithm are slightly higher than that for the greedy algorithm, and the greedy algorithm achieves much higher profits for the delivery company

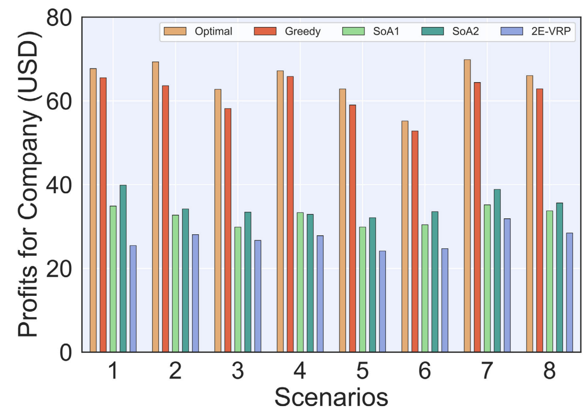


Fig. 14. The profits for the delivery company under 8 different delivery scenarios of the benchmark dataset.

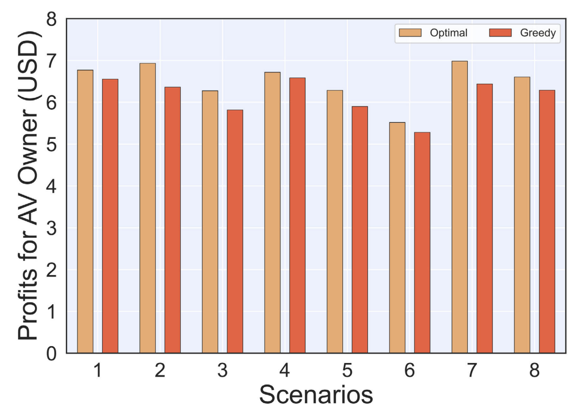


Fig. 15. The profits for participating AV owners under 8 different delivery scenarios of the benchmark dataset.

compared with that for SoA1, SoA2, and 2E-VRP algorithms mainly because our algorithms do not incur the fixed cost. The profits for the company obtained with the greedy algorithm are higher by 89%, 75%, and 126% compared with that for the SoA1, SoA2, and 2E-VRP algorithms, respectively. Regarding the profits for AV owners, it is observed that the average profits for each AV owner are \$6.5 and \$6.2 when the greedy and optimal algorithms are used, respectively.

VII. CONCLUSION

In this article, we have presented an Autonomous Vehicle Routing Problem with Drones (A-VRPD) that integrates AVs into a traditional vehicle routing problem to fully automate drone-based last-mile delivery. A Mixed Integer Linear Programming (MILP) based mathematical model is developed to simultaneously optimize the scheduling and routes of AVs, serving customers in a way that minimizes the total operational cost. The proposed two-phase approach significantly reduces the running time, addressing the scalability issue stemming from the large number of AVs and enabling frequent solution update to account for real-time traffic conditions. We have also presented a greedy algorithm to solve A-VRPD, especially for large-scale delivery scenarios with a large number of customers. Extensive simulations were performed in various delivery scenarios with varying fuel costs and salaries for truck drivers. The results demonstrate that the proposed solution produces a significant amount of profits both for the delivery company and AV owners at a much faster running speed in comparison with state-of-the-art VRP-D and 2E-VRP algorithms.

The expected social impact of the proposed solution is significant as personal delivery is emerging as a new paradigm especially in recent years when many countries are suffering from driver shortage. There are numerous companies that have or are planning to adopt a new solution to utilize personal vehicles to deliver parcels based on a crowd-sourcing approach. Amazon introduced the Amazon Flex system that allows anyone to easily participate in parcel delivery with only minimum requirements. UPS also adopted the personal vehicle driver system, and FedEx is exploiting part-time drivers to perform delivery using their personal vehicles. Not to mention the major logistics companies, various local companies are currently running their delivery business relying on personal vehicles. Given the growing demand for personal delivery and significant advances in autonomous vehicles, we expect that the proposed work will contribute to opening the new door for research on fully automated crowd sourcing and drone-based last-mile delivery systems to help our society prepare for the future.

REFERENCES

[1] P. Kitjacharoenchai, B.-C. Min, and S. Lee, "Two echelon vehicle routing problem with drones in last mile delivery," *Int. J. Prod. Econ.*, vol. 225, Jul. 2020, Art. no. 107598.

[2] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, "Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey," *Networks*, vol. 72, no. 4, pp. 411–458, 2018.

[3] S. H. Chung, B. Sah, and J. Lee, "Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions," *Comput. Oper. Res.*, vol. 123, Nov. 2020, Art. no. 105004.

[4] I. Dayarian, M. Savelsbergh, and J. P. Clarke, "Same-day delivery with drone resupply," *Transp. Sci.*, vol. 54, no. 1, pp. 229–249, Jan. 2020.

[5] A. Meola. *Shop Online and Get Your Items Delivery by a Drone Delivery Service: The Future Amazon and Domino's Have Envisioned For Us*. Accessed: Jan. 13, 2021. [Online]. Available: https://www.businessinsider.com/delivery-drones-market-service-2017-7

[6] E. Volkman. *8 Things That Can be Delivered by Drone*. Accessed: Nov. 11, 2020. [Online]. Available: https://tech.co/8-things-can-delivered-drone-2017-12

[7] DHL. *DHL Express Launches Its First Regular Fully-Automated and Intelligent Urban Drone Delivery Service*. Accessed: Jan. 15, 2021. [Online]. Available: https://www.dhl.com/global-en/home/press/press-archive/2019

[8] K. Leo. *Alibaba Begins Drone Delivery Trials in China*. Accessed: Feb. 11, 2021. [Online]. Available: https://www.bbc.com/news/technology-31129804

[9] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 86–109, May 2015.

[10] P. L. Gonzalez-R, D. Canca, J. L. Andrade-Pineda, M. Calle, and J. M. Leon-Blanco, "Truck-drone team logistics: A heuristic approach to multi-drop route planning," *Transp. Res. C, Emerg. Technol.*, vol. 114, pp. 657–680, May 2020.

[11] S. A. Vásquez, G. Angulo, and M. A. Klapp, "An exact solution method for the TSP with drone based on decomposition," *Comput. Oper. Res.*, vol. 127, Mar. 2021, Art. no. 105127.

[12] R. Roberti and M. Ruthmair, "Exact methods for the traveling salesman problem with drone," *Transp. Sci.*, vol. 55, no. 2, pp. 315–335, Mar. 2021.

[13] N. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman problem with drone," *Transp. Sci.*, vol. 52, no. 4, pp. 965–981, Apr. 2015.

[14] M. Dell'Amico, R. Montemanni, and S. Novellani, "Drone-assisted deliveries: New formulations for the flying sidekick traveling salesman problem," *Optim. Lett.*, vol. 15, pp. 1617–1648, Jul. 2021.

[15] Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Hà, "On the min-cost traveling salesman problem with drone," *Transp. Res. C, Emerg. Technol.*, vol. 86, pp. 597–621, Jan. 2018.

[16] R. Daknama and E. Kraus, "Vehicle routing with drones," 2017, *arXiv:1705.06431*.

[17] D. Schermer, M. Moeini, and O. Wendt, "Algorithms for solving the vehicle routing problem with drones," in *Proc. Asian Conf. Intell. Inf. Database Syst.* Dong Hoi City, Vietnam: Springer, 2018, pp. 352–361.

[18] P. Bouman, N. Agatz, and M. Schmidt, "Dynamic programming approaches for the traveling salesman problem with drone," *Networks*, vol. 72, no. 4, pp. 528–542, Dec. 2018.

[19] Z. Tang, W.-J. van Hoeve, and P. Shaw, "A study on the traveling salesman problem with a drone," in *Proc. Int. Conf. Integr. Constraint Program., Artif. Intell., Oper. Res.* Thessaloniki, Greece: Springer, 2019, pp. 557–564.

[20] Y. S. Chang and H. J. Lee, "Optimal delivery routing with wider drone-delivery areas along a shorter truck-route," *Expert Syst. Appl.*, vol. 104, pp. 307–317, Aug. 2018.

[21] S. M. Ferrandez, T. Harbison, T. Weber, R. Sturges, and R. Rich, "Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm," *J. Ind. Eng. Manage.*, vol. 9, no. 2, pp. 374–388, 2016.

[22] P. A. Tu, N. T. Dat, and P. Q. Dung, "Traveling salesman problem with multiple drones," in *Proc. 9th Int. Symp. Inf. Commun. Technol.*, 2018, pp. 46–53.

[23] X. Wang, S. Poikonen, and B. Golden, "The vehicle routing problem with drones: Several worst-case results," *Optim. Lett.*, vol. 11, no. 4, pp. 679–697, 2017.

[24] S. Poikonen, X. Wang, and B. Golden, "The vehicle routing problem with drones: Extended models and connections," *Networks*, vol. 70, no. 1, pp. 34–43, Aug. 2017.

[25] D. Schermer, M. Moeini, and O. Wendt, "A matheuristic for the vehicle routing problem with drones and its variants," *Transp. Res. C, Emerg. Technol.*, vol. 106, pp. 166–204, Sep. 2019.

[26] D. Sacramento, D. Pisinger, and S. Ropke, "An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones," *Transp. Res. C, Emerg. Technol.*, vol. 102, pp. 289–315, May 2019.

[27] P. Kitjacharoenchai, M. Ventresca, M. Moshref-Javadi, S. Lee, J. M. A. Tanchoco, and P. A. Brunese, "Multiple traveling salesman problem with drones: Mathematical model and heuristic approach," *Comput. Ind. Eng.*, vol. 129, pp. 14–30, Mar. 2019.

[28] F. Duarte and C. Ratti, "The impact of autonomous vehicles on cities: A review," *J. Urban Technol.*, vol. 25, no. 4, pp. 3–18, Oct. 2018.

[29] M. Thomas and T. Deepti, "Reinventing carsharing as a modern and profitable service," in *Proc. Intell. Transp. Soc. Amer. Annu. Meeting*, 2018, pp. 4–7.

[30] A. Stocker and S. Shaheen, "Shared automated vehicle (SAV) pilots and automated vehicle policy in the U.S.: Current and future developments," in *Road Vehicle Automation 5*. Orlando, FL, USA: Springer, 2019, pp. 131–147.

[31] S. Narayanan, E. Chaniotakis, and C. Antoniou, "Shared autonomous vehicle services: A comprehensive review," *Transp. Res. C, Emerg. Technol.*, vol. 111, pp. 255–293, Feb. 2020.

[32] T. Schlenther, K. Martins-Turner, J. F. Bischoff, and K. Nagel, "Potential of private autonomous vehicles for parcel delivery," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2674, no. 11, pp. 520–531, Nov. 2020.

[33] M. Amine Masmoudi, S. Mancini, R. Baldacci, and Y.-H. Kuo, "Vehicle routing problems with drones equipped with multi-package payload compartments," *Transp. Res. E, Logistics Transp. Rev.*, vol. 164, Aug. 2022, Art. no. 102757.

[34] R. J. Kuo, S.-H. Lu, P.-Y. Lai, and S. T. W. Mara, "Vehicle routing problem with drones considering time windows," *Expert Syst. Appl.*, vol. 191, Apr. 2022, Art. no. 116264.

[35] R. Liu and S. Jiang, "A variable neighborhood search algorithm with constraint relaxation for the two-echelon vehicle routing problem with simultaneous delivery and pickup demands," *Soft Comput.*, vol. 26, pp. 8879–8896, Jan. 2022.

[36] Y. Wang, Z. Wang, X. Hu, G. Xue, and X. Guan, "Truck–drone hybrid routing problem with time-dependent road travel time," *Transp. Res. C, Emerg. Technol.*, vol. 144, Nov. 2022, Art. no. 103901.

[37] R. Gu, M. Poon, Z. Luo, Y. Liu, and Z. Liu, "A hierarchical solution evaluation method and a hybrid algorithm for the vehicle routing problem with drones and multiple visits," *Transp. Res. C, Emerg. Technol.*, vol. 141, Aug. 2022, Art. no. 103733.

[38] S.-H. Huang, Y.-H. Huang, C. A. Blazquez, and C.-Y. Chen, "Solving the vehicle routing problem with drone for delivery services using an ant colony optimization algorithm," *Adv. Eng. Informat.*, vol. 51, Jan. 2022, Art. no. 101536.

[39] M. A. Nguyen, G. T.-H. Dang, M. H. Hà, and M.-T. Pham, "The min-cost parallel drone scheduling vehicle routing problem," *Eur. J. Oper. Res.*, vol. 299, no. 3, pp. 910–930, Jun. 2022.

[40] A. Rave, P. Fontaine, and H. Kuhn, "Drone location and vehicle fleet planning with trucks and aerial drones," *Eur. J. Oper. Res.*, vol. 308, no. 1, pp. 113–130, Jul. 2023.

[41] L. C. Montaña, L. Malagon-Alvarado, P. A. Miranda, M. M. Arboleda, E. L. Solano-Charris, and C. A. Vega-Mejía, "A novel mathematical approach for the Truck-and-Drone location-routing problem," *Proc. Comput. Sci.*, vol. 200, pp. 1378–1391, Jan. 2022.

[42] P. Sitek, J. Wikarek, and M. Jagodziński, "A proactive approach to extended vehicle routing problem with drones (EVRPD)," *Appl. Sci.*, vol. 12, no. 16, p. 8255, Aug. 2022.

[43] G. Wu, N. Mao, Q. Luo, B. Xu, J. Shi, and P. N. Suganthan, "Collaborative truck-drone routing for contactless parcel delivery during the epidemic," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25077–25091, Dec. 2022.

[44] A. Baniamerian, M. Bashiri, and R. Tavakkoli-Moghaddam, "Modified variable neighborhood search and genetic algorithm for profitable heterogeneous vehicle routing problem with cross-docking," *Appl. Soft Comput.*, vol. 75, pp. 441–460, Feb. 2019.

[45] M. Tang, B. Ji, X. Fang, and S. S. Yu, "Discretization-strategy-based solution for berth allocation and quay crane assignment problem," *J. Mar. Sci. Eng.*, vol. 10, no. 4, p. 495, Apr. 2022.

[46] L. Fanjul-Peyro, R. Ruiz, and F. Perea, "Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times," *Comput. Oper. Res.*, vol. 101, pp. 173–182, Jan. 2019.

[47] S. Poikonen, B. L. Golden, and E. Wasil, "A branch-and-bound approach to the traveling salesman problem with a drone," *Informs J. Comput.*, vol. 31, no. 2, pp. 335–346, 2019.

[48] H. Y. Jeong, B. D. Song, and S. Lee, "Truck-drone hybrid delivery routing: Payload-energy dependency and no-fly zones," *Int. J. Prod. Econ.*, vol. 214, pp. 220–233, Aug. 2019.

[49] O. Dukkanci, B. Kara, and T. Bektas, "The drone delivery problem," Tech. Rep., 2019, vol. 10. [Online]. Available: https://ssrn.com/abstract=3314556, doi: 10.2139/ssrn.3314556.

[50] K. Wang, B. Yuan, M. Zhao, and Y. Lu, "Cooperative route planning for the drone and truck in delivery services: A bi-objective optimisation approach," *J. Oper. Res. Soc.*, vol. 71, no. 10, pp. 1657–1674, 2020.

[51] C. C. Murray and R. Raj, "The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones," *Transp. Res. C, Emerg. Technol.*, vol. 110, pp. 368–398, Jan. 2020.

[52] K. Fujii, K. Higuchi, and J. Rekimoto, "Endless flyer: A continuous flying drone with automatic battery replacement," in *Proc. IEEE 10th Int. Conf. Ubiquitous Intell. Comput. IEEE 10th Int. Conf. Autonomic Trusted Comput.*, Dec. 2013, pp. 216–223.

[53] D. Lee, J. Zhou, and W. T. Lin, "Autonomous battery swapping system for quadcopter," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2015, pp. 118–124.

[54] K. A. O. Suzuki, P. K. Filho, and J. R. Morrison, "Automatic battery replacement system for UAVs: Analysis and design," *J. Intell. Robot. Syst.*, vol. 65, nos. 1–4, pp. 563–586, Jan. 2012.

[55] S. C. De Silva, M. Phlernjai, S. Rianmora, and P. Ratsamee, "Inverted docking station: A conceptual design for a battery-swapping platform for quadrotor UAVs," *Drones*, vol. 6, no. 3, p. 56, Feb. 2022.

[56] H. Jung, K. Lee, and W. Chun, "Integration of GIS, GPS, and optimization technologies for the effective control of parcel delivery service," *Comput. Ind. Eng.*, vol. 51, no. 1, pp. 154–162, Sep. 2006.

[57] MATLAB. *Optimization Toolbox*. Accessed: Oct. 11, 2020. [Online]. Available: https://www.mathworks.com/products/optimization.html

[58] *Fuel Consumption for Selected Gasoline and Diesel Vehicles*. Accessed: Feb. 3, 2021. [Online]. Available: https://www.energy.gov/eere/vehicles

[59] ZipRecruiter. *How Much Do Fedex Driver Jobs Pay Per Hour?* Accessed: Aug. 10, 2021. [Online]. Available: https://www.ziprecruiter.com/Salaries/Fedex-Driver-Salary-per-Hour

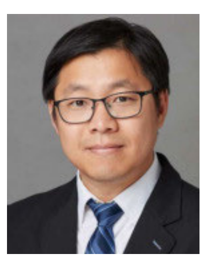
[60] J. Allen et al., "Understanding the impact of e-commerce on last-mile light goods vehicle activity in urban areas: The case of London," *Transp. Res. D, Transp. Environ.*, vol. 61, pp. 325–338, Jun. 2017.

[61] Wired. *The Astronomical Math Behind UPS' New Tool to Deliver Packages Faster*. Accessed: Aug. 10, 2021. [Online]. Available: https://www.wired.com/2013/06/ups-astronomical-math/

[62] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Oper. Res.*, vol. 35, no. 2, pp. 254–265, Apr. 1987.

Navid Mohammad Imran received the B.Sc. degree in electrical engineering from South Dakota State University in 2016. He is currently pursuing the Ph.D. degree with the Department of Computer Science, The University of Memphis. His current research interests include intelligent transportation systems, vehicle-to-everything (V2X) networks, and the application of machine learning for their optimization and security.

Sabyasachee Mishra received the M.S. degree from the Indian Institute of Technology (IIT) Bombay, India, and the Ph.D. degree from Wayne State University, Detroit, MI, USA. He is currently an Associate Professor with the Department of Civil Engineering, The University of Memphis, and the Director of C-TIER. He is involved with several national and state projects from the Federal Highway Administration, the U.S. Department of Transportation, and the Tennessee Department of Transportation. His research interests include travel demand modeling, transportation planning, policy, transportation economics, and smart cities.



Myounggyu Won (Member, IEEE) received the Ph.D. degree in computer science from Texas A&M University, College Station, TX, USA, in 2013. He was a Post-Doctoral Researcher with the Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), South Korea, from July 2013 to July 2014. He is currently an Assistant Professor with the Department of Computer Science, The University of Memphis, Memphis, TN, USA. Prior to joining The University of Memphis, he was an Assistant Professor with the Department of Electrical Engineering and Computer Science, South Dakota State University, Brookings, SD, USA, from August 2015 to August 2018. His research interests include smart sensor systems, connected vehicles, mobile computing, wireless sensor networks, and intelligent transportation systems. He received the Graduate Research Excellence Award from the Department of Computer Science and Engineering, Texas A&M University, in 2012.