Communication-aware Drone Delivery Problem

Cihan Tugrul Cicek, Çağrı Koç, Hakan Gultekin, Güneş Erdoğan

***Abstract*—The drone delivery problem (DDP) has been introduced to include aerial vehicles in last-mile delivery operations to increase efficiency. However, the existing studies have not incorporated the communication quality requirements of such a delivery operation. This study introduces the communication-aware DDP (C-DDP), which incorporates handover and outage constraints into the conventional multi-depot multi-trip green vehicle routing problem with time windows. In particular, any trip of a drone to deliver a customer package must require less than a certain number of handover operations and cannot exceed a predefined outage duration threshold. A mixed integer programming (MIP) model is developed to minimize the total flight distance while satisfying communication constraints. We present a genetic algorithm (GA) that can solve large instances and compare its performance with an off-the-shelf MIP solver. Computational study shows that the GA and MIP solver performances are equivalent to solving smaller instances. @R1.C1 We also compare the GA performance against another evolutionary algorithm, particle swarm optimization (PSO), for larger instances and find that the GA outperforms the PSO with slightly longer CPU times. The results indicate that ignoring the communication constraints would cause significant operational disruption risk and this risk can be easily mitigated with a slight sacrifice from flight distances by incorporating the proposed communication constraints. In particular, the communication performance can be improved by up to 28.9% when the flight distance is increased by 19.1% at most on average.**

***Index Terms*—Drone delivery, genetic algorithm, outage, handover.**

I. Introduction

In the past decades, rapid technological advances have resulted in the increasing use of *unmanned aerial vehicles* (UAVs) in diverse sectors [1]. Among others, last-mile delivery has been one of the promising areas to utilize UAVs [2], [3]. In [4], it has been shown that the integration of drones into conventional delivery operations would substantially reduce the total cost of delivery operations. Industry leaders like Amazon [5] and UPS [6] have already implemented UAVs in their parcel delivery process, and it's predicted that the number of robots including these vehicles in the transportation industry will double by 2030 [7].

Following this trend, there have been several studies to optimally design drone routing operations. However, none of those studies have integrated the communication perspective into the problem, and consequently have fallen short of providing a comprehensive operation planning. This study addresses this gap and introduces a new variant of the *drone delivery problem* (DDP). The objective of the DDP is to determine the optimal routes of a set of drones that are dispatched from a fixed base point, e.g. depot, deliver an order to a customer, and return to its base.

This problem has been studied as a sub-class of routing problems with drones and distinguishes it from the truck-drone routing problem, where trucks and drones are jointly used for delivery tasks and trucks are used as launching/reverting platforms for drones [2]. In their recent survey paper, Macrina et al. [2] have argued that the DDP studies have been scarce, considered only the drone characteristics, and require more investigation by incorporating environment-related conditions into the problem description to achieve more realistic results. Note that while there are studies on routing problems involving multiple depots and multiple trips, as well as environmentally-friendly vehicles, such as those discussed in [8], [9], these studies did not include aerial vehicles in their analyses. As a result, we limit our literature review to routing problems specifically involving aerial vehicles. Based on that, our major contribution is to demonstrate how communication-related constraints affect the overall route planning and more precisely the total travel distance. To the best of our knowledge, this is the first study that considers communication constraints in the DDP together with common drone characteristics, such as limited battery capacity.

There have been several research attempts to study different variants of drone routing problems. The stochastic drone routing problem is addressed in [10], which monitors and directs drone traffic under uncertainty. The authors developed a policy by adapting the Bellman equation and using an approximate algorithm. The multi-trip drone routing problem is studied in [11], where the energy consumption of the drones is modeled as a nonlinear function of payload and distance traveled. The authors developed logical and subgradient cuts in the solution process to implement the complex convex energy function. [12] developed a genetic algorithm to solve the heterogeneous UAV routing problem. The problem of scheduling and sequencing drone routes for the delivery of medical items is studied in [13], where locations for charging stations (CSs), as well as the assignment of clinics to providers, are determined. The *vehicle routing problem* (VRP) with drones that incorporates time windows for customers and requires coordination between trucks and drones is addressed in [14]. The authors provided a mathematical model to minimize the total travel time of trucks and proposed a variable neighborhood search

C. T. Cicek is with the Department of Industrial Engineering, Atilim University, Ankara, Turkey (e-mail: ctcicek@atilim.edu.tr).

Ç. Koç is with the Department of Business Administration, Social Sciences University of Ankara, Ankara, Turkey (e-mail: cagri.koc@asbu.edu.tr).

H. Gultekin is with the Department of Mechanical and Industrial Engineering, Sultan Qaboos University, Muscat, Oman (e-mail: hgultekin@squ.edu.om).

G. Erdoğan is with the School of Management, University of Bath, UK (e-mail: ge277@bath.ac.uk).

algorithm. The drone routing problem with recharging stops is studied in [15]. The authors developed a learning-based heuristic algorithm to solve the problem. The drone routing for parcel delivery has been investigated in [16]–[18].

One of the important drawbacks of the above studies is the lack of communication environment in the problem description. To operate safely and effectively, a drone needs to stay connected to a reliable backhaul node like a ground base station. This connection is necessary for various reasons, including caching, authentication, navigation, and traffic updates [19]. It allows the drone to communicate with its control unit to complete assigned tasks, such as delivering a package [20]. Maintaining this connection is also crucial for the safety of people, the environment, and property. This is the main reason that countries have put regulations and guidelines into action to ensure drones stay connected, e.g., [21]. Even when in autopilot mode, a drone might still need a backhaul connection to send status updates. For example, if a delivery is unsuccessful, the drone needs to re-plan its return trip to consider the added weight of the undelivered package. In this case, a reliable connection would help the planner monitor the status of the drone and reroute the drone to a CS or a depot to prevent a dead battery. Maintaining the connection also enables real-time data collection, which can be used for performance analysis and optimization of the operations. Therefore, the underlying communication infrastructure must be considered in drone routing problems. We aim to address this requirement and attempt to solve a new problem for aerial package delivery applications. In particular, we introduce the *communication-aware drone delivery problem* (C-DDP), which addresses how to determine optimal drone trips to deliver customer packages subject to communication quality constraints.

Figure 1 presents an illustration of the C-DDP. Two alternative trips for a package delivery plan are depicted. In the first route, a drone is dispatched from the depot and follows a straight-line path to the customer, delivers the order, and visits a CS before returning to the depot. This trip requires the drone to pass through three different coverage areas, risking handover. In the second route, the drone modifies its trajectory and visits only two coverage areas. There is a trade-off between these two routes regarding flight distance and communication quality. The first route requires less flight distance but more handover operations, whereas the second route requires fewer handover operations, but more flight distance.

II. System Model

The C-DDP is defined on a complete directed graph $G = (V, A)$ with a set of vertices V and a set of arcs A. The vertices consist of the communication nodes (CN), i.e., base stations, $V^{\mathrm{CN}} = \{CN_0, CN_1, \ldots, CN_{n_{CN}-1}\}$, depots, $V^{\mathrm{D}} = \{D_0, D_1, \ldots, D_{n_D-1}\}$, CSs, $V^{\mathrm{CS}} = \{CS_0, CS_1, \ldots, CS_{n_{CS}-1}\}$, and customers $V^{\mathrm{C}} = \{C_0, C_1, \ldots, C_{n_C-1}\}$. As a result, we have $V = V^{\mathrm{CN}} \cup V^{\mathrm{D}} \cup V^{\mathrm{CS}} \cup V^{\mathrm{C}}$. The arcs consist of the pairs of vertices between which a drone can fly, i.e., $A = \{(i, j) : i, j \in V^{\mathrm{F}}\}$, where $V^{\mathrm{F}} = V^{\mathrm{D}} \cup V^{\mathrm{CS}} \cup V^{\mathrm{C}}$ denotes the flyable nodes.

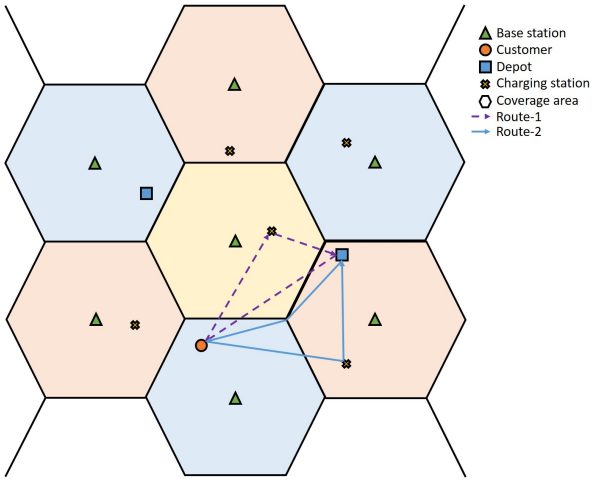


Fig. 1: ***Illustration of the C-DDP.*** *The dashed curves show a trip in which communication performance is disregarded. Consequently, the trip consists of straight flights between nodes to minimize the total flight distance. The solid curves show another trip in which the communication performance, e.g., the number of handovers, is considered.*

A. Communication Network

The communication network consists of multiple fixed, ground CNs that have access to the core communication network. The service area is divided into multiple coverage areas corresponding to different CNs. During a trip, a drone may not be within the *line-of-sight* (LoS) of a CN all the time; therefore, it must establish reliable communication links with the CNs to inform its status to a control center. Reliability in this context can be defined as experiencing a certain *quality-of-service* (QoS) level, e.g., *spectral efficiency* (SE), higher than a predefined threshold.

Since a CN on the ground (i.e., GBS) is more capable of supporting technologies, we assume that *power domain non-orthogonal multiple access* (PD-NOMA) and *orthogonal frequency-division multiple access* (OFDMA) are applied for the CNs and drones, respectively. Such technologies have already been proposed for aerial communications [22], [23]. We use the probabilistic *pathloss* model of [24] where some of the connections have a higher LoS probability while the remaining connections suffer from the lack of a LoS connection and experiences lower LoS probability (see the Appendix for the channel descriptions).

We assume that the drones operate at a fixed altitude H^{D} and establish a link with the closest CN in its corresponding coverage area. Based on these connections, two different QoS measures are considered. The first measure is the number of handovers. In particular, every time a flying drone establishes a new link to a new CN, a handover operation is required, i.e., transferring the bandwidth from one CN to the other. We expect to have as few handover operations as possible during a trip. The second measure is the expected outage duration.

The outage in this context can be defined as experiencing a SE that is less than a predefined SE threshold ($\bar{\gamma}$).

More precisely, let γ_i denote the SE when receiving from CN_i and let $v_1, v_2 \in V^{\mathrm{F}}$ denote two different nodes in the network. We divide the straight path between these two nodes into R equal line segments, yielding $R+1$ points on this particular path (including v_1 and v_2), since this discretization has been shown to be an effective approach for computing communication measures in cellular-connected trajectory design [25], [26]. Let b_r and γ_r denote the closest CN and the SE at a particular point r receiving from this closest CN, respectively. Then, the number of handovers, ($h_{v_1v_2}$), and the expected outage duration, ($o_{v_1v_2}$), on the arc (v_1, v_2) can be approximated as

$$h_{v_1v_2} = \sum_{r=2}^{R+1} \mathbb{1}(b_r \neq b_{r-1}), \tag{1}$$

$$o_{v_1v_2} = \frac{t_{v_1v_2}}{R+1} \sum_{r=1}^{R+1} \mathbb{1}(\gamma_r < \bar{\gamma}), \tag{2}$$

where $t_{v_1v_2}$ is the travel time on the arc (v_1, v_2) and $\mathbb{1}(z)$ is the indicator function that returns 1 if z is *True*, and 0 otherwise.

B. Delivery Network

We consider a delivery network in which drones, operated by a control center, deliver orders directly from the depots to customers. We assume that the customer locations are known, the packages are not heavier than the payload capacity of the drones, and a drone can deliver only one order on a trip. A trip is defined as a sequence of arcs starting from a depot node, visiting at most one customer node, and ending at a depot node.

Drones operate within a finite working day, and each drone is associated with a starting and an ending depot, which are not necessarily the same. A drone starts the day at its starting point, can operate multiple trips, and must return to its ending point. We assume that any depot can fulfill orders; thus, a drone is not forced to return to the same depot from which it was loaded. Note that when we use the term "depot," we refer to any type of storage facility that can facilitate fast delivery. Essentially, depots are capable of preparing final packages similar to warehouses in the proposed network.

A drone may visit a CS on a trip to swap its battery with a fully charged one to prevent crashing due to a depleted battery. We assume that the CSs have infinite capacities, i.e., swapping starts immediately after a drone arrives at a CS and requires a fixed and identical time at every station independent of the battery level of the arriving drone. Hence, the following constraints must be satisfied:

- The first trip of a drone must start from its starting depot.
- The last trip of a drone must end at its ending depot.
- Each trip starts from a depot, ends at a depot, and can include at most one customer.
- Each customer is served exactly once by one drone.
- Each customer must be served within its time window.

C. Problem Statement

We consider a fleet of homogeneous drones $U = \{U_1, U_2, \ldots, U_{n_U}\}$. Each customer $i \in V^{\mathrm{C}}$ has a time window $[a_i, b_i]$ within which the delivery must be completed. Each flyable node $i \in V^{\mathrm{F}}$ is associated with an operation time w_i denoting the time a drone must spend. This includes loading a package or battery swapping at a depot node, unloading a package at a customer node, or swapping at a CS. Each drone $u \in U$ is associated with a starting and an ending depot $D_u^S, D_u^E \in V^{\mathrm{D}}$. Each arc $(i,j) \in A$ is associated with a travel time t_{ij}, a travel distance d_{ij}, a travel cost c_{ij}, e.g., required battery power, a handover number h_{ij}, and an expected outage duration o_{ij}.

Let $T_u = \{T_u^1, T_u^2, \ldots, T_u^{n_C}\}$ be the feasible set of trips of drone $u \in U$, where each item in this set consists of arc sequences used in a particular trip, i.e., $T_u^k = \{(v_1, v_2), \ldots, (v_{n_k-1}, v_{n_k}) : v_1 = D_u^S, v_{n_k} = D_u^E, (v_{i-1}, v_i) \in A, i = 2, \ldots, n_k\}$ denotes the k^{th} trip of drone u with n_k arcs flown. Note that the maximum number of trips could be equal to the number of customers for a particular drone since each trip can only serve a single customer. In case a drone has fewer trips than the number of customers, then, the corresponding trip is empty.

For a given trip T_u^k, we introduce the following equations:

$$H(T_u^k) = \sum_{(i,j)\in T_u^k} h_{ij}, \tag{3}$$

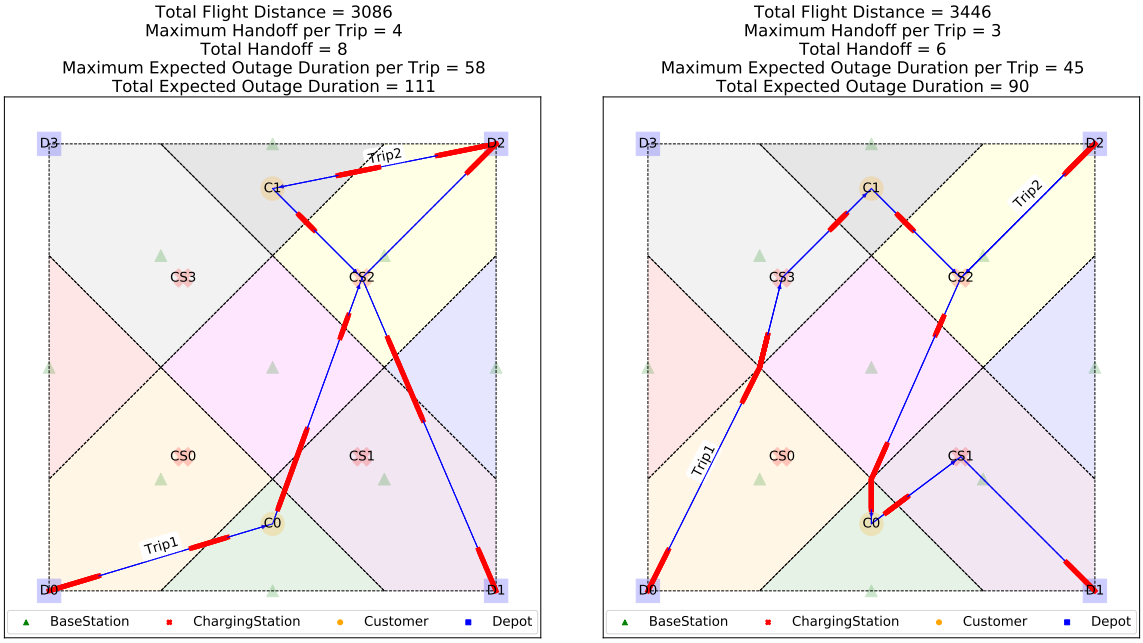
$$O(T_u^k) = \sum_{(i,j)\in T_u^k} o_{ij}, \tag{4}$$

$$\mathcal{D}(T_u^k) = \sum_{(i,j)\in T_u^k} d_{ij}, \tag{5}$$

where $H(T_u^k)$, $O(T_u^k)$, and $\mathcal{D}(T_u^k)$ define the number of handover operations, the outage duration, and the total flight distance, respectively, for the k^{th} trip of drone u. Let Ω be the set of all feasible trip sets for all drones. Then, given the handover and outage QoS thresholds $H^{\max}$ and $O^{\max}$, respectively, the C-DDP is defined as:

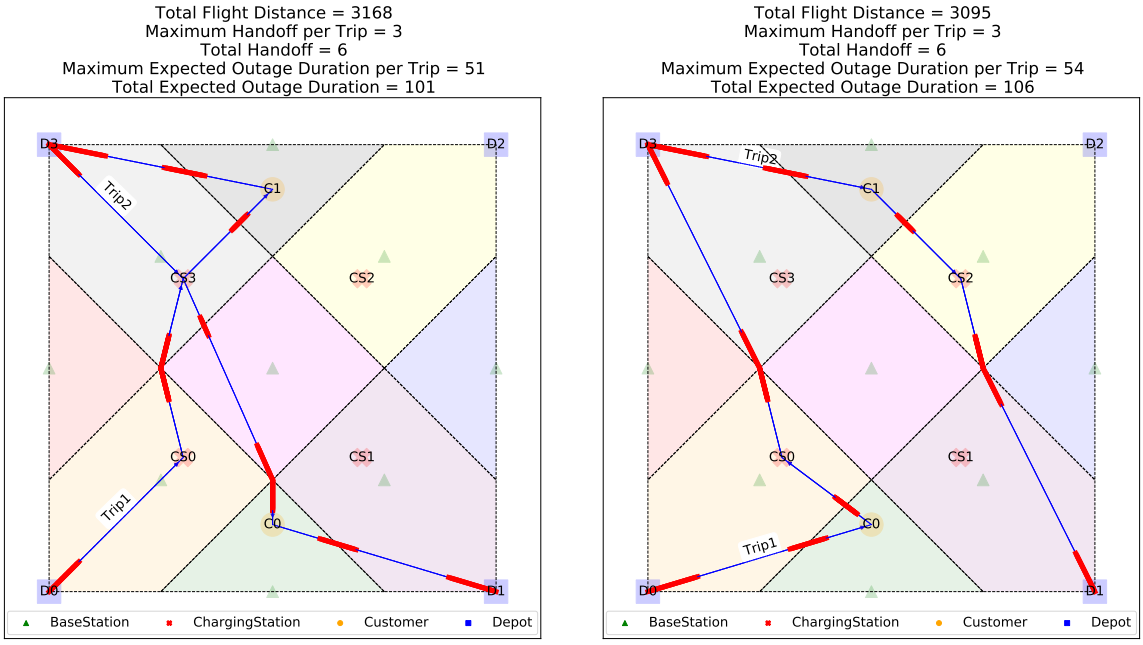
$$\begin{aligned}
\text{minimize} \quad & \sum_{u\in U} \sum_{k\in\{1,\ldots,n_C\}} \mathcal{D}(T_u^k) \\
\text{subject to} \quad & H(T_u^k) \leq H^{\max}, u \in U, k \in \{1, \ldots, n_C\}, && (6)\\
& O(T_u^k) \leq O^{\max}, u \in U, k \in \{1, \ldots, n_C\}, && (7)\\
& \mathbf{T} \in \Omega, && (8)
\end{aligned}$$

where $\mathbf{T}$ is the set of all possible trips of all drones, i.e., $\mathbf{T} = \{T_{U_1}, T_{U_2}, \ldots, T_{U_{n_U}}\}$. The objective is to minimize the total flight distance of trips while constraints (6) and (7) ensure that the number of handovers and expected outage duration of any trip do not exceed the QoS thresholds, respectively. For example, if $H^{\max} = 3$ and $O^{\max} = 30$, then a trip cannot have more than 3 handovers and more than 30 seconds of outage. Note that due to the massive number of trips, it is difficult to define Ω even for small problems explicitly, thus, we propose a MIP formulation in Section II-E.



(a) $H^{\max} = \infty$, $O^{\max} = \infty$.

(b) $H^{\max} = 3$, $O^{\max} = 45$.



(c) $H^{\max} = 3$, $O^{\max} = 51$.

(d) $H^{\max} = 3$, $O^{\max} = 56$.

Fig. 2: ***Illustrative example given*** $\bar{\gamma} = 2$. *The thicker red parts of the trips indicate the expected outage. The top-left panel shows the solution for the default problem. The top-right, bottom-left, and bottom-right panels show the case in which the outage duration constraint is relaxed by 10%, 20%, and 30%, respectively, while the handover constraint is set to its optimum.*

D. Illustrative Example

In Fig. 2, we illustrate the C-DDP on a 1000×1000 square region with two customers (C) located at $[500, 150]$ and $[500, 900]$, four depots (D) located at $[0, 0]$, $[1000, 0]$, $[1000, 1000]$, and $[0, 1000]$, and four CSs located at $[300, 300]$, $[700, 300]$, $[700, 700]$, and $[300, 700]$. The communication network includes 9 base stations located on equidistant diagonals and $\bar{\gamma}$ is set to 2.

We assume that a single drone will deliver the customer packages starting from depot "D0" and ending at depot "D1". For simplicity, we ignore the time windows in this example. The different colored regions show the coverage areas in the communication network, and the dashed lines depict the boundaries of these areas. The top left panel shows the optimal solution for the default case, where the objective is to minimize the total flight distance and

constraints (6), (7) are ignored. We have also solved the same problem by considering handover and outage objectives to determine optimal communication performance. In particular, we set the objective function of C-DDP to minimize the maximum number of handovers, i.e., $\min_{u\in U} \max_{k\in\{1,\dots,n_C\}} H(T_u^k)$, and the maximum expected outage duration per trip, i.e., $\min_{u\in U} \max_{k\in\{1,\dots,n_C\}} O(T_u^k)$, with no additional constraints. We obtained the optimal handover count and outage duration as 3 and 43, respectively.

We illustrate three different cases in Fig. 2, where we set the objective function to minimize the total flight distance and consider the optimal handover and outage duration as hard constraints. Note that the problem becomes infeasible once we enforce both the optimal handover and outage constraints, i.e., $H^{\max} = 3$ and $O^{\max} = 43$. Therefore, we illustrate the cases in which we relax the maximum outage duration by 10% (top right), 20% (bottom left), and 30% (bottom right).

We can easily observe that integrating the communication constraints increases the total flight distance. In particular, the flight distance increases by 12.3%, 11.7%, and 6.7% respectively when at most 10%, 20%, and 30% relaxation are allowed for the maximum expected outage duration. However, discarding the communication constraints would result in a 34.8% increase in the expected outage duration in the default case compared to the optimal expected outage duration per trip. Note that relaxing the handover constraint would not affect these solutions since only the outage constraint has become active in this particular example.

E. MIP Formulation of the C-DDP

Let $K = \{0, 1, 2, \dots, n_C-1\}$ denote the set of trips for each drone. Similar to the four-indexed variables defined in [9], we introduce the decision variables $x_{ijuk} \in \{0, 1\}$ to denote if arc $(i, j) \in A$ is flown by drone $u \in U$ on its k^{th} trip, 0 otherwise; $p_{uk} \in \{0, 1\}$ to denote if drone $u \in U$ operates trip $k \in K$, 0 otherwise; $0 \le y_{iuk} \le 1$ to denote the battery level of drone $u \in U$ at the time of arrival at node $i \in V^{\text{F}}$ on its k^{th} trip; $s^L_{iuk} \ge 0$ and $s^A_{iuk} \ge 0$, respectively, to denote leaving and arrival time of drone $u \in U$ at depot $i \in V^{\text{D}}$ on its k^{th} trip; and $s^V_{iuk} \ge 0$ to denote the arrival time of drone $u \in U$ at node $i \in V^{\text{F}}$ on its k^{th} trip. Recall that d_{ij}, h_{ij}, o_{ij}, c_{ij}, t_{ij} denote, respectively, the distance, number of handovers, expected outage duration, required battery power, and required time associated with arc (i, j); a_i and b_i denote the start and end of time window of node i, respectively; and w_i denotes the required operation time at node i. Then, the MIP formulation of C-DDP can be defined as

$$\text{minimize} \quad \sum_{(i,j)\in A} \sum_{u\in U} \sum_{k\in K} x_{ijuk} d_{ij}$$

$$\text{subject to} \quad \sum_{(i,j)\in A} h_{ij} x_{ijuk} \le H^{\max}, u \in U, k \in K, \tag{9}$$

$$\sum_{(i,j)\in A} o_{ij} x_{ijuk} \le O^{\max}, u \in U, k \in K, \tag{10}$$

$$\sum_{\substack{i\in V^{\text{F}},\\ (i,j)\in A}} \sum_{u\in U} \sum_{k\in K} x_{ijuk} = 1, j \in V^{\text{C}}, \tag{11}$$

$$\sum_{\substack{j\in V^{\text{F}}\\ (i,j)\in A}} x_{ijuk} - \sum_{\substack{j\in V^{\text{F}},\\ (j,i)\in A}} x_{jiuk} = 0, i \in V^{\text{F}} \setminus V^{\text{D}}, u \in U, k \in K, \tag{12}$$

$$p_{u0} \le \sum_{\substack{j\in V^{\text{F}}\\ (D^S_u,j)\in A}} x_{D^S_u,j,u,0}, u \in U, \tag{13}$$

$$p_{uk} - p_{u,k+1} \le \sum_{\substack{j\in V^{\text{F}}\\ (j,D^E_u)\in A}} x_{j,D^E_u,u,k}, u \in U, k \in K \setminus \{n_C - 1\}, \tag{14}$$

$$p_{u,n-1} \le \sum_{\substack{j\in V^{\text{F}}\\ (j,D^E_u)\in A}} x_{j,D^E_u,u,n-1}, u \in U, \tag{15}$$

$$x_{ijuk} \le p_{uk}, (i, j) \in A, u \in U, k \in K, \tag{16}$$

$$p_{u,k+1} \le p_{uk}, u \in U, k \in K \setminus \{n_C - 1\}, \tag{17}$$

$$\sum_{i\in V^{\text{D}}} \sum_{\substack{j\in V^{\text{F}}\\ (i,j)\in A}} x_{ijuk} \le 1, u \in U, k \in K, \tag{18}$$

$$\sum_{i\in V^{\text{D}}} \sum_{\substack{j\in V^{\text{F}}\\ (j,i)\in A}} x_{jiuk} \le 1, u \in U, k \in K, \tag{19}$$

$$\sum_{i\in V^{\text{C}}} \sum_{\substack{j\in V^{\text{F}}\\ (j,i)\in A}} x_{jiuk} \le 1, u \in U, k \in K, \tag{20}$$

$$\sum_{\substack{j\in V^{\text{F}}\\ (i,j)\in A}} x_{i,j,u,k+1} - \sum_{\substack{j\in V^{\text{F}}\\ (j,i)\in A}} x_{jiuk} \le 0, i \in V^{\text{D}}, u \in U, k \in K \setminus \{n_C - 1\}, \tag{21}$$

$$x_{ijuk} - 1 \le y_{juk} - (1 - c_{ij}) \le 1 - x_{ijuk}, \quad i \in V^{\text{CS}} \cup V^{\text{D}}, \quad j \in V^{\text{F}}, (i, j) \in A, u \in U, k \in K, \tag{22}$$

$$x_{ijuk} - 1 \le y_{juk} - (y_{iuk} - c_{ij}) \le 1 - x_{ijuk}, \quad i \in V^{\text{F}} \setminus \{V^{\text{CS}} \cup V^{\text{D}}\}, \quad j \in V^{\text{F}}, (i, j) \in A, \quad u \in U, k \in K, \tag{23}$$

$$a_i - M\left(1 - \sum_{\substack{j\in V^{\text{F}}\\ (j,i)\in A}} x_{jiuk}\right) \le s^V_{iuk} \le b_i + M\left(1 - \sum_{\substack{j\in V^{\text{F}}\\ (j,i)\in A}} x_{jiuk}\right), \quad i \in V^{\text{C}}, u \in U, k \in K, \tag{24}$$

$$s^A_{iuk} - M(1 - p_{u,k+1}) \le s^L_{i,u,k+1}, i \in V^{\text{D}}, u \in U, k \in K \setminus \{n_C - 1\}, \tag{25}$$

$$s^L_{iuk} + w_i + t_{ij} - M(1 - x_{ijuk}) \le s^V_{juk}, \quad i \in V^{\text{D}}, j \in V^{\text{F}} \setminus V^{\text{D}}, (i, j) \in A, \quad u \in U, k \in K, \tag{26}$$

$$s^V_{iuk} + w_i + t_{ij} - M(1 - x_{ijuk}) \le s^V_{juk},$$

$$i, j \in V^{\mathrm{F}} \setminus V^{\mathrm{D}}, (i,j) \in A, u \in U, k \in K, \tag{27}$$

$$s^V_{iuk} + w_i + t_{ij} - M(1 - x_{ijuk}) \leq s^A_{juk}, \quad i \in V^{\mathrm{F}} \setminus V^{\mathrm{D}}, j \in V^{\mathrm{D}}, (i,j) \in A, u \in U, k \in K, \tag{28}$$

$$s^L_{iuk} + t_{ij} - M(1 - x_{ijuk}) \leq s^A_{juk}, \quad i, j \in V^{\mathrm{D}}, (i,j) \in A, u \in U, k \in K, \tag{29}$$

$$x_{ijuk} \in \{0,1\}, (i,j) \in A, u \in U, k \in K, \tag{30}$$

$$p_{uk} \in \{0,1\}, u \in U, k \in K, \tag{31}$$

$$0 \leq y_{iuk} \leq 1, i \in V^{\mathrm{F}}, u \in U, k \in K, \tag{32}$$

$$0 \leq s^L_{iuk}, s^A_{iuk}, i \in V^{\mathrm{D}}, u \in U, k \in K, \tag{33}$$

$$0 \leq s^V_{iuk}, i \in V^{\mathrm{F}} \setminus V^{\mathrm{D}}, u \in U, k \in K. \tag{34}$$

The objective function minimizes the total flight distance. Constraints (9) and (10) ensure that the entire delivery operation is fulfilled without exceeding the handover and outage thresholds, respectively. Constraints (11) state that each customer is visited once by one drone. Constraints (12) define the flow balance at each vertex. Constraints (13)-(15) ensure that a drone starts the workday from its starting depot and ends the day at its ending depot. In particular, (13) enforces having an outgoing arc from the starting depot of a drone if that drone is assigned its first trip, while (14) and (15) enforce to have an incoming arc to the ending depot of a drone if that drone is not assigned a new trip after operating its last trip. Constraints (16) ensure that arcs can be flown by a particular drone on a particular trip if that trip is assigned to that drone. Constraints (17) organize the trip sequences. Constraints (18)-(20) require that each trip can have at most one leaving depot, one arrival depot, and a customer.

Constraints (21) ensure that a drone can start its next trip from a particular depot if and only if the previous trip ends at that particular depot. More precisely, this constraint holds for equality when a trip is immediately followed by another trip for a drone. In this case, the left-hand side becomes 0. However, it holds for inequality when a trip does not have a successive trip. For example, given a problem with four customers, if a drone only makes a single trip to one of these customers, then, the left-hand side of this constraint becomes -1 as some of the x variables corresponding to this drone with $k = 0$ would be set to 1, and all other x variables with $k > 0$ would be set to 0. Therefore, constraint (21) holds for inequality for $k = 0$, and for equality for $k = 1, 2$.

Constraints (22)-(23) determine the battery levels. In particular, if a trip involves traveling from a source node of depot or CS i to a node j, i.e., $x_{ijuk} = 1$ for $i \in V^{\mathrm{CS}} \cup V^{\mathrm{D}}$, then the battery level at the source node i is assumed to be full. The constraint (22) ensures that the battery level at the node of interest j is equal to 1 minus the energy required to travel between nodes i and j. On the other hand, if a trip involves traveling from a source node i that is neither a depot nor a CS to a specific node j, i.e. $x_{ijuk} = 1$ for $i \in V^{\mathrm{F}} \setminus \{V^{\mathrm{CS}} \cup V^{\mathrm{D}}\}$, then the battery level at node j is equal to the battery level at the source node i minus the energy required to travel between nodes i and j. Note that constraints (32) ensure that the battery levels are always within 0 and 1 at any node visited by any drone on any trip.

Constraints (24)-(29) state the time relationships, where M is a sufficiently large number. In particular, constraints (24) ensure that a customer's delivery is fulfilled within its time window. Constraints (25) ensure that the leaving time of the next trip for a drone cannot be earlier than the arrival time of the previous trip. Constraints (26) set the visiting time of the first non-depot node in a trip to the leaving time from the starting depot of the trip plus the operation time at the depot plus the travel time from the depot to the first node. Constraints (27) determine the visiting time of a non-depot node as the visiting time of the previous non-depot node plus operation time spent in the previous node plus the travel time from the previous node. The operation times in these constraints correspond to loading, unloading, and charging times for depots, customers, and charging nodes, respectively. Constraints (28) set the arrival time to a depot node from a non-depot node to the visiting time of the last non-depot node plus the operation time at that last node plus the travel time from that last node to the depot node. Constraints (29) set the arrival time to a depot node from another depot node to the leaving time of the outgoing depot plus the travel time between depots. Note that this constraint does not include any operation time since we assume that flying between two depot nodes does not require any loading/unloading operation. Finally, constraints (30)-(34) define the domains of the variables.

III. Solution Approach

The MIP formulation moves out of our computational reach for large problem instances. Therefore, we propose a *Genetic Algorithm* (GA) to solve larger instances effectively. @R1.C1 / @R2.C1 In this section, we describe how to represent a solution, how to create the initial population, how to generate offspring, and how to mutate a solution. We have used the *pymoo* package [27] to implement the special crossover and mutation operators.

A. Solution Representation

A chromosome in GA defines how a solution is represented. In our context, the chromosome is defined as the combination of drone routes where each route consists of labels. We use a classical label encoding scheme where each node is represented with a unique integer. Fig. 3 illustrates this chromosome scheme for a problem with 4 depots, 2 CSs, 2 customers, 16 waypoints, and 2 drones. Note that we encode all nodes in the network plus *waypoints* (WPs) representing the intersection points of coverage areas to identify handovers.

B. Evaluation of Chromosomes

We use Eq. 5 as the basis of the fitness function. More precisely, we first calculate the total flight distance over all

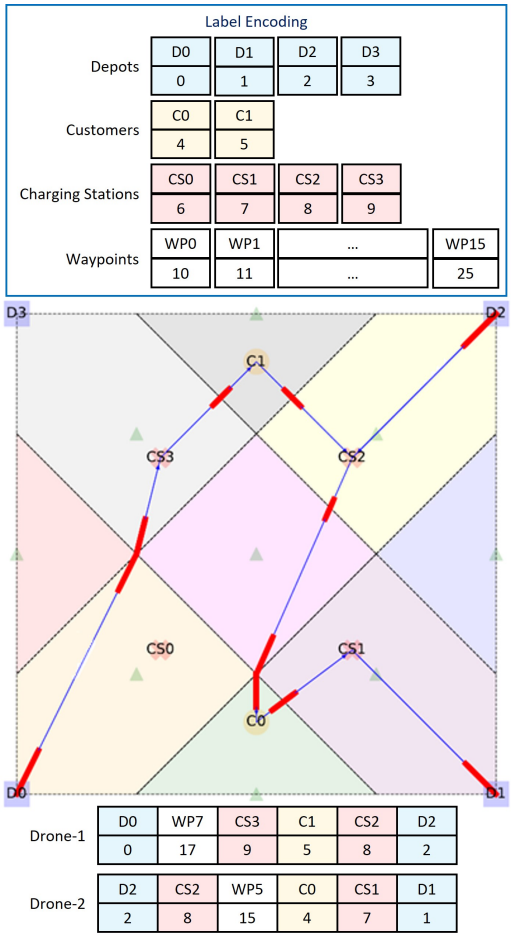


Fig. 3: *An illustration of the chromosome for a solution where the starting and ending depots of Drone-1 are D0 and D2, while the starting and ending depots of Drone-2 are D2 and D1, respectively. The top and bottom rows in the label encoding and routes indicate the nodes and their corresponding labels, respectively.*

drones that are used in a solution and then add the sum of constraint violation values over all constraints to this total if a chromosome does not satisfy the constraints. This approach has been widely used in the GA literature to handle constraints, e.g. [28], [29]. In this way, feasible solutions are encouraged to be used in the subsequent generations of the algorithm.

C. Initial Population

We create the initial population with an approach similar to [9]. We start with generating a uniform random number between 0 and $n_u - 1$ for each customer. The integer part of this number is used to assign the customers to drones. Then, we aggregate the customers that have been assigned to the same drone and sort the customers based on their random numbers in a non-decreasing order. Then, the initial route of a drone starts from its starting depot, visits all customers in this sorted order, where a random depot is inserted after each customer visit to do the collection for the next customer, and returns to the drone's ending depot. However, the initial routes might be infeasible due to the battery constraints. Therefore, we introduce a greedy CS insertion to recover the initial routes. In particular, we go through every route and if the battery level drops down to zero at a particular node of a route, we insert the CS that is closest to the predecessor of that particular node. We make sure that the insertion of CSs guarantees that the drone can finish its route. In case a route is still infeasible after the CS insertion, the solution is removed from the population. Fig. 4 illustrates how this process is executed for a problem with 4 drones, 4 depots, and 6 customers.

D. Crossover

We use the binary tournament selection, i.e., random selection with uniform probability, to create offspring from two parents in the population. Let P_1 and P_2 be the parents and $n^d_{P_1}$ and $n^d_{P_2}$ denote the number of customers assigned to drone d in P_1 and P_2, respectively. We generate two random numbers for each drone which denote the number of customers inherited from different parents. Then, we randomly select the customers to pass to the offspring while ensuring that no duplicate customers are passed. When a customer is passed to the offspring, we keep the partial route from the collection depot to the depot visited after the customer delivery in the same structure for this particular customer. In case a customer is unassigned in this process, we randomly select a drone to assign this customer. The customer selection process is duplicated such that we create two offsprings from every two parents selected. In the first and second offsprings, the customers are selected by starting from P_1 and P_2, respectively.

Fig. 5 illustrates the crossover operation for a problem with 4 drones and 6 customers. Drone 0 visits customers 6 and 2 in Parent 1 while visiting customers 1 and 3 in Parent 2. To create an offspring, customers 6 and 2 from Parent 1 and customer 3 from Parent 2 are inherited for this drone. A similar inheritance is applied to other drones and as a result, Drone 2 becomes idle in the offspring. Note that we keep the collection depot of customers inherited from Parent 2 when the route is reconstructed in the offspring. For instance, after delivering customer 2 in the route of Drone 0 in the offspring, the drone visits depot 2 since the collection depot of customer 3 was depot 2 in Parent 2. In case no customer is inherited from Parent 2, we keep the same partial route of a particular customer from Parent 1.

E. Mutation

We use the traditional *customer reinsertion*, *customer swap*, *CS insertion*, and *CS removal* operators [30]. In the customer reinsertion, a customer is removed and reinserted to a different position in the same route. In the customer swap, two customers from two different routes are swapped. Note that these reinsertion and swapping operations are executed by preserving the partial route structure of customers, i.e., the partial route starting from the collection depot of a customer and ending at another depot after the delivery of the customer. In the CS insertion (removal), a CS is inserted into (removed from) a random position in a route.

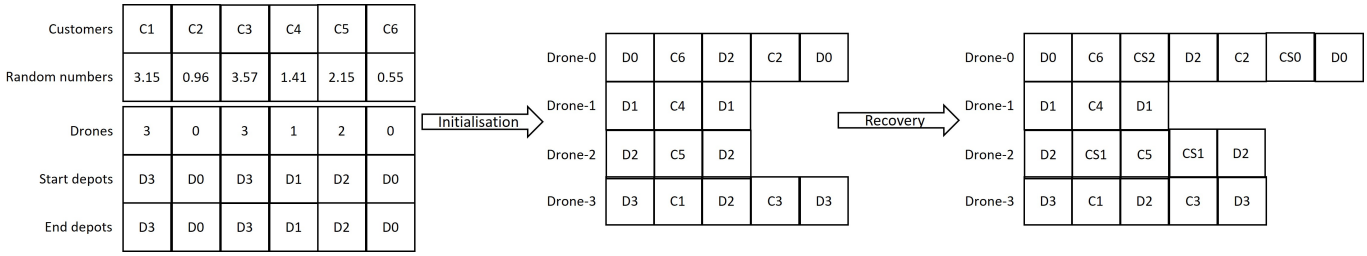


Fig. 4: @R2.C1 *An illustration of the initial chromosome generation including the CS insertion.*

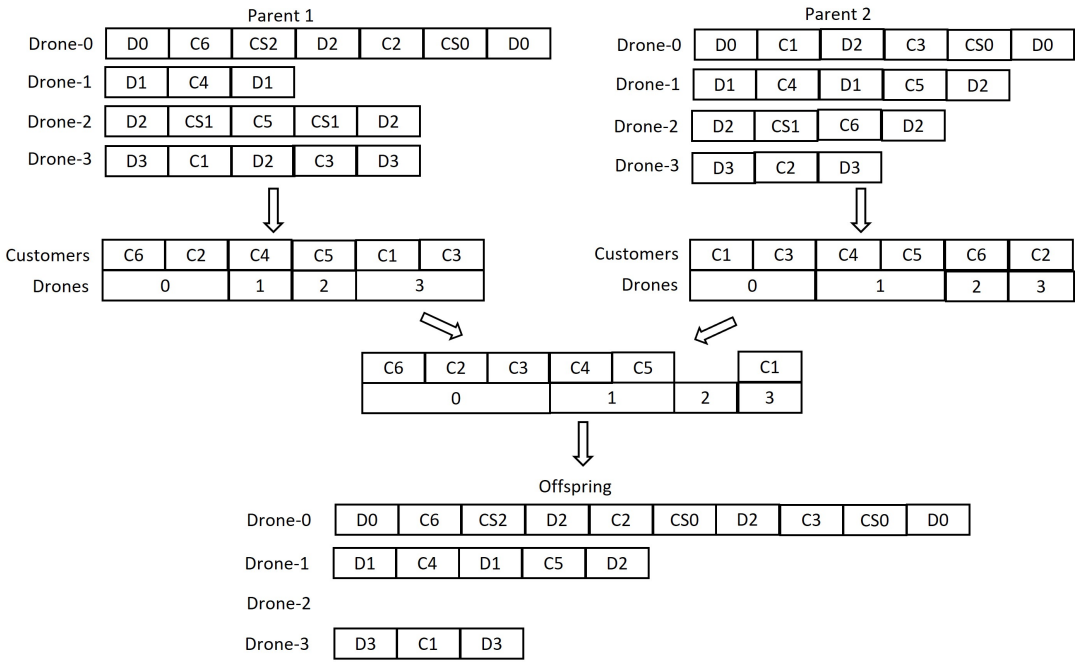


Fig. 5: @R2.C1 *An illustration of the crossover operation.*

The above operators do not consider the variety in handover and outage constraints and do not allow changing the collection depot of a particular customer in a route. To incorporate these, we introduce three new mutation operators: *waypoint insertion*, *waypoint removal*, and *depot replacement*.

Waypoint insertion: A waypoint is inserted in a route with an expectation to improve the fitness value. Our motivation is that inserting a waypoint would decrease the degree of communications constraint violations, and so would the fitness value. To insert a waypoint, we randomly select a route and a node. We then insert the closest waypoint to this node as a predecessor.

Waypoint removal: A waypoint is removed from a random route among all routes that have at least one waypoint. If no route has a waypoint, then this mutator is skipped. Our motivation is that removing a waypoint would decrease the total flight distance which would have a more significant impact on the fitness value compared to the degree of communications constraint violations.

Depot replacement: An intermediate depot, which is the depot visited to collect the order before a customer delivery, is replaced with another depot. Since all other mutation operators preserve the depots before and after a customer delivery, this operator is expected to reduce the impact of bad depot selections in the initial population.

The mutation operators are applied to randomly selected 10% of the population. Then, any chromosome that becomes infeasible after the mutation is recovered using the same recovery operation applied to the initial population. In case the recovery operation fails, the mutated chromosome is removed from the population.

IV. Computational Study

In this section, we present the results of our computational experiments to validate and assess our proposed model as well as the performance of the GA. We conducted all experiments on a desktop computer with an Intel i-7@3.50 GHz processor, 16 GB RAM, and Windows 10 operating system. The MIP models were solved with Gurobi 9.5 [31].

A. Test Bed

Since there are no publicly available benchmark instances for the C-DDP, we have generated test instances to conduct numerical experiments[1]. We built our instances using a

[1]Instances are available at https://github.com/cihantugrulcicek/CDDP.

5×5 km zone for a working day of 8 hours (28,800 seconds). We located the depot and CSs for all instances such that there exists one depot every 2 km and one CS every 1 km. Then, we generated the communication network with respect to two different spatial distributions. In particular, we first created a hexagonal grid with 1 km radius and located one GBS at the center of each hexagon for the uniform case. We generated another network setting where we perturbed the center location of hexagons such that the coverage areas overlap asymmetrically.

To determine the customer locations, we used two different distributions. For the first case, the locations were determined according to *Uniform* (U) distribution. For the second case, we used the Poisson Point process, where we first generated multiple hotpoints, and then determined customer locations around these hotpoints. The number of customers around each hotpoint was also determined based on a discrete uniform distribution.

The time window for visiting a customer was determined as follows: we first determined the earliest time that a customer can be visited according to its travel time to the closest and the farthest depot. Let D' and D'' be the closest and farthest depots to customer $i \in V^{\mathrm{C}}$, respectively. We set the center of the time window by generating a uniform random number from $\bar{a}_i = \mathrm{U}(t_{D',i}, 8 - t_{i,D''})$. We then generate another uniform random number as the width of the time window and modify the exact time window with respect to this width. Note that in case of having the earliest time less than $t_{D',i}$ or the latest time greater than $(8 - t_{i,D''})$, we adjust the time window such that the earliest time is set to $t_{D',i}$ and the latest time is set to $(8 - t_{i,D''})$. To test the impact of time windows, we generate the width from two different Uniform distributions, i.e., we draw a random integer number from $\mathrm{U}(2, 8)$ for the loose case, and $\mathrm{U}(1, 4)$ for the tight case.

In all instances, the number of drones is determined based on one drone per 25 customers. The starting and ending depots of drones are randomly determined, assuring that each depot hosts at least one drone at the beginning and end of the planning horizon if the number of drones is greater than the number of depots. The communication parameters are set to dense-urban settings provided by [32]: $\alpha_1 = 12.08$, $\alpha_2 = 0.11$, $\alpha_3 = 2.5$, $\mu_{\mathrm{LoS}} = 1.6$, $\mu_{\mathrm{NLoS}} = 23$, $\sigma^2 = -173$ dB, $P_i = 46$ dBm for all $i \in V^{\mathrm{CN}}$.

To succinctly describe the instances, we use a three-digit notation. "U" or "P" in the first digit denotes Uniform and Perturbed communication network, "U" or "P" in the second digit denotes Uniform and Poisson Process customer distribution, and "L" or "T" in the last digit denotes loose and tight time windows, e.g., "PUT" indicates the setting with the Perturbed communication network, Uniform delivery network, and Loose time windows.

B. Results

We set the Gurobi parameters to their defaults except that the time limit has been set to @R2.C2 7,200 seconds. In GA, we set the initial population size to 10,000 and the number of offsprings to 2,000. The GA is terminated whenever the run time exceeds @R2.C2 7,200 seconds, or the number of generations exceeds ten times the number of customers ($10n_C$) in a particular instance, e.g., 500 generations for an instance with 50 customers.

We first compare the GA solution performance (optimality gap) against Gurobi. In particular, let P^{best} denote the best bound reported by Gurobi and P^{G} denote the best feasible Gurobi objective value when it terminates with a time limit. Also, let P^{GA} denote the best feasible GA objective value. Then, $(P^{\mathrm{G}} - P^{\mathrm{best}})/P^{\mathrm{best}} \times 100$ and $(P^{\mathrm{GA}} - P^{\mathrm{best}})/P^{\mathrm{best}} \times 100$ are the optimality gaps of Gurobi and GA, respectively.

Table I presents the optimality gaps of Gurobi and the GA in percentages for different settings. We first solve each instance to find singular optimal values of maximum handover and expected outage. In particular, we first disabled the communication constraints, Eqs. (9) and (10), modified the objective function to find singular optimal values for the maximum number of handovers, i.e., $\min_{u \in U} \max_{k \in \{1,\dots,n_C\}} H(T_u^k)$, and outage duration, i.e., $\min_{u \in U} \max_{k \in \{1,\dots,n_C\}} O(T_u^k)$, at a time. Then, we solve the same instances to find the optimal flight distances where we include the communication performance as hard constraints based on these singular optimal values. The columns in Table I denote how much we have relaxed each communication constraint where the default case indicates that the singular optimal values were used without any relaxation and the other cases indicate the percentage we relaxed $H^{\max}$ and $O^{\max}$, respectively, e.g., if the singular optimal values for handovers and outage was 5 and 30, respectively, then we set $H^{\max} = 6$ and $O^{\max} = 33$ for the (20,10) case. Note that the more relaxed these constraints are, the more we are closer to the conventional drone routing problem with no communication consideration.

@R2.C2 As can be seen in Table I, the GA effectively solves small instances with up to 20 customers to near-optimality in a short time. The average difference in the gap values is less than 0.01% where the worst and the best performances occur for the PPT setting with 0.02% higher objective value and for the UPT setting with 1.1% lower objective value, respectively. The average solution times of the GA were almost 19 times shorter than Gurobi and more than 32 times shorter for instances with 20 customers. The average CPU time of Gurobi was 1025 seconds, while the GA terminated under 55 seconds.

Fig. 6 shows the impact of communication constraints on the total flight distance. Note that the figure depends on the results obtained from the GA due to the limited computational reach of Gurobi in solving large instances. This figure confirms our theoretical expectation that the lower the relaxation of a communication measure is, the higher the total flight distance. @R2.C2 The average total flight distance compared to the default case decreases by 6.6%, 11.4%, 15.25%, and 19.1% for communication cases (10,10), (10,20), (20,10), and (20,20), respectively. On the other hand, the default case has 12.1%, 9.6%, 7.5%, and 6.9% improvement in the number of handover activities, and 28.9%, 24.4%, 19.6%, 15.3% improvement in the expected outage duration on average for communication cases (10,10),

TABLE I: ***Optimality gaps of Gurobi and the GA in percentages.*** The gap values are calculated against the best bound found by Gurobi. The default column indicates that the $H^{\max}$ and $O^{\max}$ were set to their singular optimal values. The following columns indicate the % relaxations on handover and outage constraints.

		Gurobi					GA				
Setting	n_C	Default	(20,20)	(20,10)	(10,20)	(10,10)	Default	(20,20)	(20,10)	(10,20)	(10,10)
PPL	5	0	0	0	0	0	0	0	0	0	0
	10	2	0	0	0	0	2.1	<1	0	0	0
	20	1.3	<1	<1	<1	<1	1.8	<1	<1	<1	<1
PPT	5	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	<1	0	0	0	0
	20	1	0	0	0	<1	<1	<1	<1	<1	0
PUL	5	0	0	0	0	0	0	0	0	0	0
	10	1.5	0	0	0	0	<1	0	0	0	0
	20	0	0	0	0	0	0	0	0	0	0
PUT	5	0	0	0	0	0	0	0	0	0	0
	10	6.1	0	<1	0	<1	4.4	<1	<1	<1	0
	20	14.3	6.5	10.1	10.7	11.7	9.5	9.9	8.7	5.8	5
UPL	5	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0
	20	0	0	0	0	0	0	0	0	0	0
UPT	5	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	1.4	0	0	0	0
	20	11.1	2.5	3.6	4.2	6.1	5.4	1.6	<1	<1	<1
UUL	5	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0
	20	7.2	1.9	5.1	2.8	6.5	7.7	3.5	2.2	1.9	1.9
UUT	5	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0
	20	8.4	1.4	2.9	3.6	4.8	11.4	<1	3.1	3.3	4.4

(10,20), (20,10), and (20,20), respectively. This supports our claim that ignoring the communication constraints would result in operational disruption risk, which can be mitigated by sacrificing slightly from flight distance by the incorporation of $H^{\max}$ and $O^{\max}$.

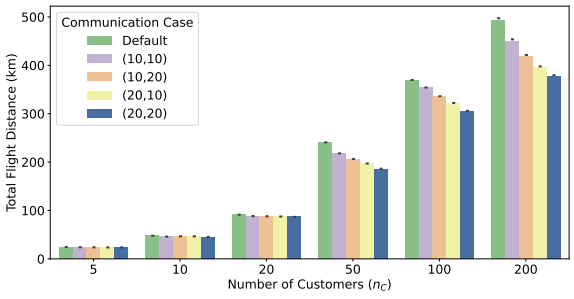
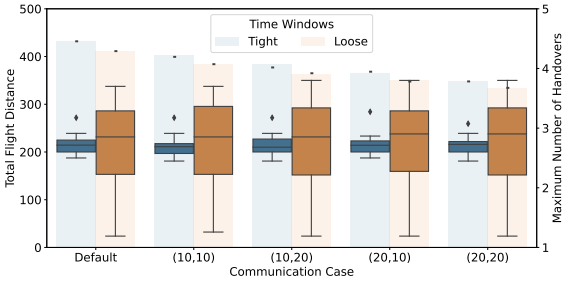


Fig. 6: ***The change in total flight distance concerning different levels of communication constraints.***

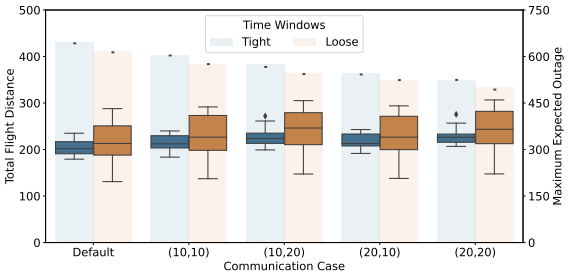
@R1.C1 Since Gurobi was unable to provide a lower bound or a feasible solution for larger problems with more than 20 customers, we assess the performance of GA on larger instances with another evolutionary algorithm. To make the comparison fair in terms of the implementation of the algorithm, we select the *Particle Swarm Optimization* (PSO) algorithm, that is included in *pymoo*. PSO was introduced by [33] and has been used in different variants of routing problems [34]–[36]. The PSO utilizes the swarm of particles, which are analogous to the chromosomes in the GA. Each particle has a velocity and can be influenced by local neighbors as well as the globally best-found solutions while moving in the search space. At each iteration, a particle determines its new location based on its historical locations as well as the best locations obtained by the other particles in the swarm. A fitness function is used to evaluate the performance of a particle at each location. Ultimately, the swarm is expected to move towards the global best solution while the members are collectively informing the other members. A more detailed explanation of the PSO can be found in [37].

Table II shows the gaps of GA and PSO in percentages for different numbers of customers. In particular, let P^{PSO} denote the objective value found by the PSO and P^{best} denote the best objective value found among the GA and the PSO. Then, the values in Table II can be calculated as $(P^{\text{GA}} - P^{\text{best}})/P^{\text{best}}$ and $(P^{\text{PSO}} - P^{\text{best}})/P^{\text{best}}$ for the GA and the PSO, respectively. The CPU times of PSO and GA were 67 and 84 seconds on average. Although the GA was slightly slower, it outperformed the PSO in all settings and customer numbers. On average, the objective values obtained by the PSO were 7% worse than the objective values obtained by the GA. Additionally, the improvement in the GA's performance was more pronounced with the higher number of customers. The GA was able to achieve 4.2%, 7.9%, and 8.7% better objective values on average for the problems with 50, 100, and 200 customers, respectively.

Fig. 7 and 8 illustrate the impact of time windows and

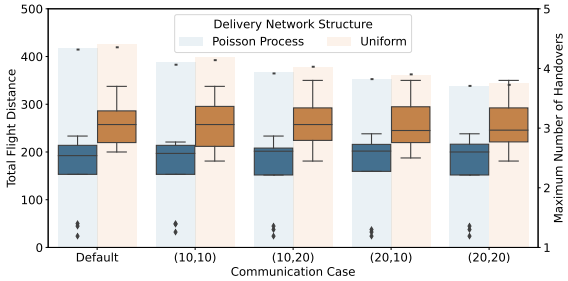


(a) Maximum Number of Handovers.

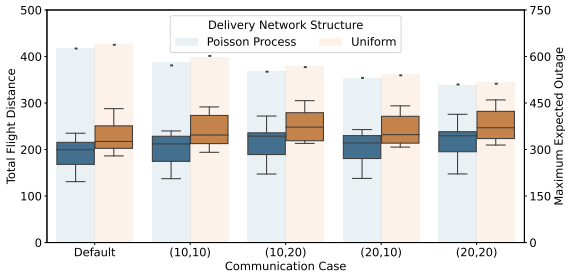


(b) Maximum Expected Outage.

Fig. 7: ***The impact of time windows on (a) maximum number of handovers and (b) maximum expected outage per trip.*** *The transparent bars show the total flight distance in both figures while the boxplots show the maximum number of handovers and maximum expected outage in the top and bottom figures, respectively. The default in the x-axis indicates the cases where the maximum handover and outage thresholds were set to their singular optimal values. The other cases show the percentage of relaxation from these thresholds.*



(a) Maximum Number of Handovers.



(b) Maximum Expected Outage.

Fig. 8: ***The impact of customer locations on (a) maximum number of handovers and (b) maximum expected outage per trip.*** *The transparent bars show the total flight distance in both figures while the boxplots show the maximum number of handovers and maximum expected outage in the top and bottom figures, respectively. The default in the x-axis indicates the cases where the maximum handover and outage thresholds were set to their singular optimal values. The other cases show the percentage of relaxation from these thresholds.*

clustering of customer requests on the total flight distance, number of handovers, and expected outage duration. The transparent bars in the background denote the average total flight distances, and the boxes indicate the corresponding communication measure. Fig. 7 indicates that it is likely to make longer flights when the time windows are tight, e.g., the total flight distance is 2% higher in the tight case on average. This can be considered normal in routing problems because it is typically harder to meet tighter time windows than looser ones. However, the type of time windows does not have a statistically significant effect on the number of handovers and the expected outage.

On the other hand, the delivery network structure significantly affects the communication performance and has a slight impact on the flight distance for all communication cases. In particular, the average number of handovers and expected outage duration decreases by 18% and 16% for instances with clustered customers. The clustered requests have also decreased the total flight distance by 3% as the trips are likely to be organized from the closest depot to a hot point rather than flying to different depots in subsequent trips. This trip strategy requires fewer handover activities and consequently decreases the expected outage duration.

TABLE II: @R1.C1 ***Comparison of PSO and GA for the large instances.*** The results are averaged over all communication cases and show the gap against the best solution obtained by the GA or the PSO concerning different numbers of customers.

	GA			PSO		
Setting	50	100	200	50	100	200
PPL	0	0	0	0.9	0.8	0.8
PPT	0	0	0	0.1	0.4	0.9
PUL	0	0	0	4.7	11.2	12.3
PUT	0	0	0	7.9	8.9	9.9
UPL	0	0	0	4.4	11.8	9.6
UPT	0	0	0	2.6	7.5	9.8
UUL	0	0	0	6.4	11.9	13.7
UUT	0	0	0	6.9	11.2	12.5

V. Conclusions

In this study, we have addressed an emerging problem in last-mile delivery networks which we believe will attract more attention soon. In particular, we have integrated two significant communication performance measures, handover, and expected outage duration, to the classical multi-depot multi-trip DDP. These measures help to improve communication performance without significantly sacrificing operational efficiency. Although it is not a straightforward task to determine the budget on these measures, e.g., determining the $H^{\max}$ and $O^{\max}$ values, we introduced an

intelligent procedure to set the maximum allowed values and presented the results showing how sensitive the solutions were against these changes. These measures could be also determined by a trial-and-error procedure, e.g., starting from an initial value and increasing/decreasing the value by a step size until the desired operational performance is satisfied.

We have developed a new mathematical formulation, which can be solved by the off-the-shelf MIP solver Gurobi for instances with up to 20 customers. @R2.C1 To solve larger problems, we have implemented a GA based on a special chromosome representation and introduced problem-specific crossover and mutation operators. Our numerical study has shown that the GA can find near-optimal solutions in small instances within reasonable times. @R1.C1 We compared the GA performance against the PSO for the larger instances and the results indicated that the GA performs better within slightly higher CPU times.

We believe that our MIP model and solution approach can easily be adapted for various applications that involve the use of aerial vehicles. For instance, our model can be modified for use in a UAV-aided wireless sensor network where drones collect data from sensor nodes. In this scenario, the sensor nodes can be considered as the customer nodes and the service time of a customer can be considered as the time required to collect the package that a sensor must deliver. Another potential application area is the agriculture industry where drones can transport protective chemicals to vulnerable plants within a specific time frame. The number of applications can be increased, however, it is important to note that all these applications require a reliable communication system and our study can be used as a guide to manage this system.

Before we conclude, it is important to acknowledge some limitations of our study and suggest areas for future research to address these limitations. The communication constraint introduced in our MIP formulation is an approximation to the actual handover and outage definitions. However, due to the complexity of these metrics, we had to use a discretization approach that assumes the communication link is the same in each segment. This assumption may not hold in real-time applications where wireless communications can be highly sensitive to slight changes in a segment. Conversely, using the exact formulations of these metrics could result in non-linear equations. A promising research area could be to develop a non-linear MIP model and propose a solution approach based on the characteristics of this model. @R3.C1 Integrating bilevel optimization approaches [38] could be another area to research. This is particularly suggested as we used a $\min-\max$ problem in the experiments stage where we first found the singular optimal values for the communication measures. This approach can be extended to include the flight distance.

APPENDIX
COMMUNICATION NETWORK DETAILS

Recall that our pathloss model assumes that there exist two groups with different LoS probabilities. The first group is assumed to have a LoS connection with probability $\mathbb{P}^{\text{LoS}}$, while the second group can have a poor LoS connection with probability $(1-\mathbb{P}^{\text{LoS}})$. Now, assume that we have a drone flying at an altitude of H^{D} in the coverage area of a particular $CN_i \in V^{\text{CN}}$. Given the location of this CN (g_i^{CN}) and the projected location of the drone on the ground (g^{D}), the probability of LoS connection can be defined as

$$\mathbb{P}_i^{\text{LoS}} = \frac{1}{1+\alpha_1 e^{-\alpha_2\left(\frac{180}{\pi}\tan^{-1}\left(\frac{H^{\text{D}}}{\|g_i^{\text{CN}}-g^{\text{D}}\|}\right)-\alpha_1\right)}}, \tag{35}$$

where α_1 and α_2 are environment-dependent parameters. Then, the mean pathloss between the drone and CN_i, (L_i), can be formulated as,

$$L_i = 10\alpha_3\log_{10}\left(\frac{4\pi f_c}{c}\sqrt{\|g_i^{\text{CN}}-g^{\text{D}}\|^2+(H^{\text{d}})^2}\right) + \left(\mu_{\text{LoS}}\mathbb{P}_i^{\text{LoS}}+\mu_{\text{NLoS}}\mathbb{P}_i^{\text{NLoS}}\right), \tag{36}$$

where f_c is the carrier frequency in Hz, c is the speed of light in m/s, α_3 is the pathloss exponent, and μ_{LoS} and μ_{NLoS} are the environment-dependent average additional losses to the free-space propagation for LoS and *non-line-of-sight* (NLoS) connections, respectively [39]. Then, the signal-to-interference-plus-noise-ratio (*SINR*) at the drone receiving from CN_i, (ρ_i), is defined as,

$$\rho_i = \frac{P_i 10^{-L_i/10}}{\sigma^2+\sum_{j\in V^{\text{CN}},j\neq i}P_j 10^{-L_j/10}} \tag{37}$$

where P_i is the transmit power of CN_i and σ^2 is the noise power at the drone receiver. Consequently, the SE when receiving from CN_i, (γ_i), is given as

$$\gamma_i = \log_2(1+\rho_i). \tag{38}$$

REFERENCES

[1] 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.

[2] G. Macrina, L. D. P. Pugliese, F. Guerriero, and G. Laporte, "Drone-aided routing: A literature review," *Transportation Research Part C: Emerging Technologies*, vol. 120, p. 102762, 2020.

[3] A. Thibbotuwawa, G. Bocewicz, P. Nielsen, and Z. Banaszak, "Unmanned aerial vehicle routing problems: a literature review," *Applied sciences*, vol. 10, no. 13, p. 4504, 2020.

[4] F. Borghetti, C. Caballini, A. Carboni, G. Grossato, R. Maja, and B. Barabino, "The use of drones for last-mile delivery: A numerical case study in milan, italy," *Sustainability*, vol. 14, no. 3, 2022.

[5] Amazon, "Amazon Prime Air prepares for drone deliveries," 2022. [Online]. Available: https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries

[6] UPS, "UPS operates first ever U.S. drone COVID-19 vaccine delivery," 2021. [Online]. Available: https://about.ups.com/us/en/our-stories/innovation-driven/drone-covid-vaccine-deliveries.html

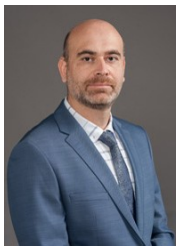
[7] McKinsey & Co., "Last-mile package delivery in 2030," 2019. [Online]. Available: https://www.mckinsey.com/industries/travel-logistics-and-transport-infrastructure/our-insights/last-mile-package-delivery-in-2030

[8] M. E. H. Sadati and B. Çatay, "A hybrid variable neighborhood search approach for the multi-depot green vehicle routing problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 149, p. 102293, 2021.

[9] L. Zhen, C. Ma, K. Wang, L. Xiao, and W. Zhang, "Multi-depot multi-trip vehicle routing problem with time windows and release dates," *Transportation Research Part E: Logistics and Transportation Review*, vol. 135, p. 101866, 2020.
[10] J. Y. Chow, "Dynamic UAV-based traffic monitoring under uncertainty as a stochastic arc-inventory routing policy," *International Journal of transportation science and technology*, vol. 5, no. 3, pp. 167–185, 2016.
[11] C. Cheng, Y. Adulyasak, and L.-M. Rousseau, "Drone routing with energy function: Formulation and exact algorithm," *Transportation Research Part B: Methodological*, vol. 139, pp. 364–387, 2020.
[12] X. Yuan, J. Zhu, Y. Li, H. Huang, and M. Wu, "An enhanced genetic algorithm for unmanned aerial vehicle logistics scheduling," *IET Communications*, 2021.
[13] Z. Ghelichi, M. Gentili, and P. B. Mirchandani, "Logistics for a fleet of drones for medical item delivery: A case study for Louisville, KY," *Computers & Operations Research*, vol. 135, p. 105443, 2021.
[14] R. Kuo, S.-H. Lu, P.-Y. Lai, and S. T. W. Mara, "Vehicle routing problem with drones considering time windows," *Expert Systems with Applications*, p. 116264, 2021.
[15] U. Ermağan, B. Yıldız, and F. S. Salman, "A learning based algorithm for drone routing," *Computers & Operations Research*, vol. 137, p. 105524, 2022.
[16] P. P. Repoussis, C. D. Tarantilis, and G. Ioannou, "Arc-guided evolutionary algorithm for the vehicle routing problem with time windows," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 624–647, 2009.
[17] X. Yan, H. Huang, Z. Hao, and J. Wang, "A graph-based fuzzy evolutionary algorithm for solving two-echelon vehicle routing problems," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 129–141, 2019.
[18] D. Rojas Viloria, E. L. Solano-Charris, A. Muñoz-Villamizar, and J. R. Montoya-Torres, "Unmanned aerial vehicles/drones in vehicle routing problems: a literature review," *International Transactions in Operational Research*, vol. 28, no. 4, pp. 1626–1657, 2021.
[19] G. Karabulut Kurt and H. Yanikomeroglu, "Communication, computing, caching, and sensing for next-generation aerial delivery networks: Using a high-altitude platform station as an enabling technology," *IEEE Vehicular Technology Magazine*, vol. 16, no. 3, pp. 108–117, 2021.
[20] N. Cherif, W. Jaafar, H. Yanikomeroglu, and A. Yongacoglu, "3D aerial highway: The key enabler of the retail industry transformation," *IEEE Communications Magazine*, vol. 59, no. 9, pp. 65–71, 2021.
[21] Federal Register, "Spectrum rules and policies for the operation of unmanned aircraft systems," 2023. [Online]. Available: https://www.federalregister.gov/documents/2023/02/07/2023-00961/spectrum-rules-and-policies-for-the-operation-of-unmanned-aircraft-systems
[22] A. Azizi, S. Parsaeefard, M. R. Javan, N. Mokari, and H. Yanikomeroglu, "Profit maximization in 5G+ networks with heterogeneous aerial and ground base stations," *IEEE Transactions on Mobile Computing*, vol. 19, no. 10, pp. 2445 – 2460, 2019.
[23] C. T. Cicek, Z.-J. M. Shen, H. Gultekin, and B. Tavli, "3-D dynamic UAV base station location problem," *INFORMS Journal on Computing*, vol. 33, no. 3, pp. 839–860, 2021.
[24] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, Dec. 2014.
[25] S. Zhang, Y. Zeng, and R. Zhang, "Cellular-enabled UAV communication: A connectivity-constrained trajectory optimization perspective," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2580–2604, 2019.
[26] Y.-J. Chen and D.-Y. Huang, "Trajectory optimization for cellular-enabled UAV with connectivity outage constraint," *IEEE Access*, vol. 8, pp. 29 205–29 218, 2020.
[27] J. Blank and K. Deb, "pymoo: Multi-objective optimization in Python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020.
[28] K. Deb and S. Agrawal, "A niched-penalty approach for constraint handling in genetic algorithms," in *Artificial Neural Nets and Genetic Algorithms*. Springer Vienna, 1999, pp. 235–243.
[29] S. Venkatraman and G. Yen, "A generic framework for constrained optimization using genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 4, pp. 424–435, 2005.
[30] S. Karakatič and V. Podgorelec, "A survey of genetic algorithms for solving multi depot vehicle routing problem," *Applied Soft Computing*, vol. 27, pp. 519–532, 2015.
[31] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2022. [Online]. Available: https://www.gurobi.com
[32] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-D placement of an aerial base station in next generation cellular networks," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–5.
[33] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.
[34] S. MirHassani and N. Abolghasemi, "A particle swarm optimization algorithm for open vehicle routing problem," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11 547–11 551, 2011.
[35] Y. Marinakis, M. Marinaki, and G. Dounias, "A hybrid particle swarm optimization algorithm for the vehicle routing problem," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 4, pp. 463–472, 2010.
[36] Y. Marinakis and M. Marinaki, "A hybrid genetic – particle swarm optimization algorithm for the vehicle routing problem," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1446–1455, 2010.
[37] M. O. Okwu and L. K. Tartibu, "Particle swarm optimisation," in *Metaheuristic Optimization: Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications*. Cham: Springer International Publishing, 2021, pp. 5–13.
[38] F. Alesiani, "Implicit bilevel optimization: Differentiating through bilevel optimization programming," *arXiv:2302.14473*, 2023.
[39] Y. Zeng, Q. Wu, and R. Zhang, "Accessing from the sky: A tutorial on UAV communications for 5G and beyond," *Proceedings of the IEEE*, vol. 107, no. 12, pp. 2327–2375, 2019.

Cihan Tugrul Cicek received his Ph.D. degree in Industrial Engineering from TOBB University of Economics and Technology, Ankara, Turkey, in 2019. He is currently a Data Scientist with the Microlise Limited, UK. His research interests include mathematical optimization and algorithms with applications in logistics, non-terrestrial communication networks, facility location, and smart grids.

Çağrı Koç received his Ph.D. degree (2015) in Operational Research/Management Science from the Southampton Business School of University of Southampton. He worked as a Postdoctoral Fellow at HEC Montréal and at CIRRELT. He is an Associate Professor in the Department of Business Administration at the Social Sciences University of Ankara. His research mainly focuses on the application of mathematical and metaheuristic optimization techniques to logistics problems.



Hakan Gultekin received his Ph.D. degree in Industrial Engineering from Bilkent University in 2007. After his postdoc at the University of Liege, he was employed as an Assistant Professor and as an Associate Professor between 2007-2018 at TOBB University of Economics and Technology. He is an Associate Professor at Sultan Qaboos University since 2018. His research interests include scheduling, optimization modeling, and exact and heuristic algorithm development.

Güneş Erdoğan received his Ph.D. in Industrial Engineering from Bilkent University, Turkey in 2007. He was employed in CIRRELT, Montréal (2007-2008) as a post-doctoral fellow, Ozyegin University (2008-2011) as an Assistant Professor, and University of Southampton (2011-2014) as a Lecturer. He started working at the University of Bath in 2014 as a Reader and was appointed as a Professor in 2019. His research focuses on the applications of optimization algorithms to the problems arising in logistics and healthcare.

Thank you for allowing the second resubmission of our manuscript with an opportunity to address the reviewers' comments. Special thanks are due to the reviewers for their careful reading of the paper and for providing valuable comments and suggestions which have helped to improve both the content and the presentation.

Our responses to all reviewers' comments and corresponding revisions are outlined in the "Response to Reviewers" letter. Once again we would like to thank all the reviewers, associate editor, and editor-in-chief for their time and effort.