IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

# Target Search on Road Networks with Range-Constrained UAVs and Ground-Based Mobile Recharging Vehicles

Kyle E. C. Booth, Chiara Piacentini, Sara Bernardini, and J. Christopher Beck

***Abstract*— We study a range-constrained variant of the multi-UAV target search problem where commercially available UAVs are used for target search in tandem with ground-based mobile recharging vehicles (MRVs) that can travel, via the road network, to meet up with and recharge a UAV. We propose a pipeline for representing the problem on real-world road networks, starting with a map of the road network and yielding a final routing graph that permits UAVs to recharge via rendezvous with MRVs. The problem is then solved using mixed-integer linear programming (MILP) and constraint programming (CP). We conduct a comprehensive simulation of our methods using real-world road network data from Scotland. The assessment investigates accumulated search reward compared to ideal and worst-case scenarios and briefly explores the impact of UAV speeds. Our empirical results indicate that CP is able to provide better solutions than MILP, overall, and that the use of a fleet of MRVs can improve the accumulated reward of the UAV fleet, supporting their inclusion for surveillance tasks.**

## I. Introduction

Unmanned aerial vehicles (UAVs) have had impact across a wide variety of industries, including logistics [1], agriculture [2], and surveillance [3]. In the context of the latter, the problem of searching for lost targets has a long history, with theoretical studies dating back to the 1940s [4]. The search problem involves routing a fleet of surveying units to try and find a moving target. Once the target has been found, it can then be tracked; the work in this paper focuses solely on the search phase of these operations.

While the existing literature surrounding UAV search and track problems is extensive [3], [5], [6], [7], there is little work that looks at the viability of real-world, large-scale target search capabilities using range-constrained, commercially available UAVs. Commercial UAVs have a considerably shorter flight time than fixed-wing military-grade designs [8]. For example, the DJI Matrice 200 has a maximum unloaded flight time of roughly 38 minutes [9], whereas it is not uncommon for military-grade designs to have flight times far exceeding 10 hours [8]. As such, we study the range-constrained multi-UAV target search problem based on commercially available UAV specifications and the use of mobile recharging vehicles (MRVs). The MRVs can travel, via the road network associated with the search area, to meet up with and recharge a UAV. Our investigation looks to identify the viability of commercially available UAVs for search-phase surveillance missions when utilized standalone, as well as when deployed with a fleet of MRVs.

The contributions of this paper are as follows:

- We propose a pipeline for representing the problem over real-world road networks, starting with a map of the road network and yielding a final routing graph that permits UAVs to recharge via rendezvous with MRVs.
- We model and solve the resulting graph representation of the problem using both *mixed-integer linear programming* (MILP) and *constraint programming* (CP), adopting recent modeling techniques from the mathematical programming literature.
- We conduct a simulation-based assessment of our methods using real-world road network data from Scotland.

The outline of this paper is as follows. Section II defines the problem and Section III summarizes related work from the literature. Section IV presents the pipeline used for representing the problem over real-world road networks, while Section V details the optimization models. Section VI details experimental setup, results, and analysis, and Section VII provides concluding remarks.

## II. Problem Definition

The problem studied involves cooperatively routing a fleet of homogeneous UAVs in pursuit of a mobile ground-based agent (the "target"). In this problem variant, in contrast to those presented before [3], [10], [11], each UAV is range-constrained due to limited battery capacity. Ground-based *mobile recharge vehicles* (MRVs) are available to provide the UAVs the opportunity to replenish their battery.

At a high-level, the problem can be posed as follows: given a mixed-fleet of UAVs and MRVs, a map of the road network, and a set of locations likely to contain the target at certain time intervals, determine a temporally feasible and range-compliant search plan for the fleet that maximizes the accumulated expectation of discovering the target. The plan consists of a sequence of *search patterns*, i.e., target search maneuvers performed by the UAVs at specific locations and times, as well as recharge actions involving both UAVs and MRVs. For a UAV to recharge, it must be in the same location as the MRV at the same time. In this work, we assume that UAVs do not suffer from communication-related range constraints, and that UAVs consume energy while traveling and while conducting a search pattern. The remainder of this section describes the components of the

K.E.C. Booth and J.C. Beck are with the Department of Mechanical & Industrial Engineering, University of Toronto, Toronto, Canada (email: kbooth@mie.utoronto.ca, jcb@mie.utoronto.ca).

C. Piacentini was with the Department of Mechanical & Industrial Engineering, University of Toronto, Toronto, Canada (email: chiarap@mie.utoronto.ca).

S. Bernardini is with the Department of Computer Science, Royal Holloway University of London, London, UK (email: sara.bernardini@rhul.ac.uk).

problem in detail, follow existing work for the majority of the problem definition [3], [11] with minor changes to notation.

### A. Environment and Target

Our problem considers an environment in two-dimensional Euclidean space characterized by a *road network* attained from the underlying map of the search area and defined by a series of roads, each of which is a sequence of road segments with varying speed limits (see Figure 1a). The road network is then discretized into cells with side length $\delta$ to produce a *problem graph*, $G = (V, E)$ (Figure 1b-1c). Each vertex, $v \in V$, in the problem graph represents a cell containing at least one road segment, and each undirected edge, $e \in E$, represents an adjacent pair of vertices connected by a road segment in the underlying road network. Each edge, $e \in E$, is labeled with its minimum and maximum travel speeds, $s_e^{min}$ and $s_e^{max}$, respectively. The target is characterized by a *last known position* (LKP), $v_0$, and a set of potential destinations, $d \in D \subset V$. The process of going from the real-world road network to the problem graph, $G$, with search patterns is illustrated by Figures 1a to 1c.

### B. Fleets

The fleet of homogeneous UAVs is denoted $o \in O$. UAVs move between locations of interest at a constant speed of $s_O$ in metres per second. $Q$ denotes the total battery capacity of a UAV (with 0 being the minimum capacity) and the per-unit-distance energy consumption of each UAV is given by $g$. Energy consumption for a search pattern is given by $h$.

The fleet of homogeneous *mobile recharging vehicles* (MRVs) is denoted $k \in K$. MRVs move along the road network subject to road segment speed limits. To recharge a UAV, the UAV and an MRV must meet in the same location at the same time, and this location must contain a road segment. In this work, the recharging of a UAV is executed via a constant time battery swap operation [12], where a UAV meets an MRV and has its battery replaced for a full charge. This operation is assumed to take $\xi$ seconds. MRVs are not range constrained and can recharge a single UAV at a time.

### C. Target Simulation and Search Patterns

Following previous work [10], the target's motion through the graph, $G$, is simulated with a standard Monte Carlo simulation (MCS). To identify vertices in the graph that have the highest probability of containing the target at various times, the MCS uses a probability distribution defined over the set of possible target destinations, $D$, shortest paths to each of those destinations from the LKP, and estimated target travel speeds. For each simulated time step, the MCS selects the vertices with the highest probability of containing the target, and creates a search pattern centered on them, with time windows assigned to reflect when the target can plausibly be in these areas. Additional detail regarding the MCS procedure can be found in existing work [3], [10].

The search pattern itself is a pre-planned target search maneuver performed by a UAV at a specific location. Previous work has identified a series of standard manoeuvres that the UAVs can perform, such as spirals and lawnmowers, depending on the topology of the search pattern location. Spiral manoeuvres are useful for covering more dense, urban areas, while the lawnmower is more effective at searching over elongated sections [11].

Each search pattern, $c \in C$, is characterized by a location, processing time, time window, and reward. In Figure 1c, the locations of possible search patterns are represented by grey circles. The reward for a UAV performing a search pattern is calculated following previous work [3] and represents a measure that the target will be in the area of the search pattern during its time window; a higher reward is better. The time at which a UAV starts a search pattern must be within its time window. The planning horizon, $H$, is the end of the latest time window, such that no search pattern can be finished after this time. A UAV must complete the full duration of an assigned search pattern before starting another.

### D. Solution

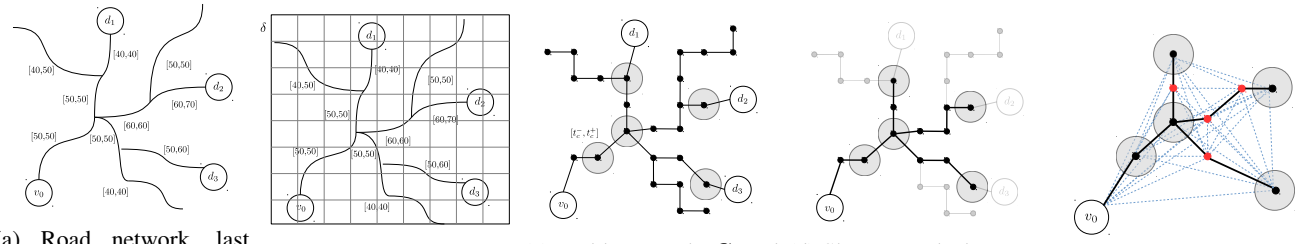
A solution to the problem is a sequence of search patterns and recharging events assigned to each UAV and a traveling route for each of the MRVs. UAV sequences must satisfy both temporal and energy constraints, while MRV sequences must only satisfy temporal constraints (i.e., we do not consider the need for the ground vehicle to refuel). The objective of the constrained optimization problem is to maximize the total accumulated reward of the UAV fleet.

## III. Related Work

Research on UAV deployment has received considerable interest, with early work introducing task assignment methods [13] and the development of capabilities such as automated battery swap-out systems for UAV recharging [12]. The impact of commercial UAVs on logistics resulted in rapid algorithmic development in the operations research and robotics literature [1], [14], [15], [16]. A review of optimization approaches to UAV routing is found in [17].

The routing of range-constrained UAVs while considering recharging has also seen considerable interest. A number of these works propose MILP models for the routing of UAV fleets with fixed location recharge stations [18], [19], while others propose heuristics and approximation algorithms [20]. Recent work incorporates recharging station placement into MILP-based UAV routing models in the context of logistics applications [21], [22]. In these works, the existence of pre-defined potential facility locations results in smaller networks that can be modeled using single-stage optimization. Conversely, our work, which looks to identify valuable locations for UAV/MRV synchronization over large real-world road networks requires significant discretization and filtering prior to optimization, as proposed by our pipeline.

The most relevant recent work considers mobile recharging stations similar our MRVs [23], [24]. In [23], they plan paths for a single UAV and single MRV, traveling via a road network, for surveillance operations. The authors propose a greedy solution, in contrast to our joint multiple vehicle UAV/MRV routing models, that first finds a path for the MRV



(a) Road network, last known position, $v_0$, and destinations, $d \in D$.

(b) Discretization with side length $\delta$. cells

(c) Problem graph, $G$, and search pattern locations.

(d) Shortest paths between search patterns.

(e) UAV and MRV routing graph, $\mathcal{G}$.

Fig. 1: UAV and MRV routing graph construction pipeline.

and then generates a path for the UAV. The recent work of [24] explores joint routing of a range-constrained UAV and MRVs for site visits. They investigate three problem variants: i) single UAV routing with multiple stationary recharging locations, ii) single UAV/single MRV joint routing, and iii) a single UAV/multiple MRV problem that, for a given UAV path, looks to minimize the number of MRVs used. Each of these problems is similar to ours, though we note that [24] allows the UAV to be transported by the ground vehicle, whereas we do not. Further differences between this work and our own: it does not start from a road network as input, it considers a single UAV, it does not consider time windows for site visits, and the underlying structure is a traveling salesman problem as opposed to the orienteering problem [25] structure in our work.

## IV. Routing Graph Construction

The problem graph, $G$, which is used to generate search pattern candidates, is too large to use directly in a monolithic optimization model. As such, following existing work in electric vehicle routing [26], [27], we define a *routing graph* to facilitate the development of mathematical models for the problem. Vertices in the routing graph represent locations of interest and arcs represent travel segments between them.

From the discretized problem graph, $G$ (Figure 1c), we generate a routing graph, $\mathcal{G} = (\mathcal{V}_{0,N+1}, \mathcal{E})$, where $\mathcal{V}_{0,N+1}$ is the set of $N+2$ vertices that UAVs/MRVs can possibly visit ($N$ is the number of non-depot vertices), and $\mathcal{E}$ the set of edges connecting them. We let $v_0$, the vertex representing target LKP, represent the start depot for all vehicles. We let $v_{N+1}$ represent the end depot for all vehicles with zero travel distance to all other vertices in $\mathcal{G}$; this vertex is simply introduced for modeling purposes. The set of vertices representing possible instances of search patterns is denoted by $\mathcal{V}^C$. The location, processing time, time window, and reward associated with a particular search pattern instance, $v_i \in \mathcal{V}^C$, are given by $\ell_i$, $p_i$, $[t_i^-, t_i^+]$, and $r_i$, respectively. We let $\Delta_{ij}^{Euc}$ represent the Euclidean distance between vertices $v_i$ and $v_j$, and $\Delta_{ij}^{SP}$ represent the shortest path MRV travel time between $v_i$ and $v_j$, following the road network and conforming to speed limits.

In the non-range-constrained variant of the problem (i.e., without MRVs), the final routing graph would simply be defined over the depot nodes, $\{v_0, v_{N+1}\}$, and $\mathcal{V}^C$. However, in the range-constrained variant, the graph must be augmented with vertices allowing opportunities for UAVs to meet MRVs to recharge. We let the set of these recharge vertices be represented by $\mathcal{V}^F$. Constructing the set of recharge vertices, $\mathcal{V}^F$, must be done carefully to avoid dramatically increasing the size of the graph. The procedure we use for accomplishing this is detailed in Algorithm 1.

**Algorithm 1** Construct recharge vertex set, $\mathcal{V}^F$

**Input**: Problem graph, $G = (V, E)$, filtering parameter, $\phi$
**Output**: $\mathcal{V}^F$

```
V+ ← ∅
for (v_i, v_j) ∈ V^C × V^C : (t_i^- + p_i + Δ_ij^Euc / s_O) ≤ t_j^+ do
    V+ ∪ ShortestPathNodesFiltered(v_i, v_j, G, φ)
end for
V^F ← ModifiedHittingSet(V+)
return V^F
```

For each pair of search pattern vertices that can feasibly be executed by the same UAV, the function ShortestPathNodesFiltered computes the shortest path through the problem graph, $G$, using Dijkstra's algorithm; the edge weights in this case represent the minimum travel time (dictated by road segment speed limits) between the two search pattern locations. The algorithm then returns the nodes involved in the path. To reduce the number of nodes considered, the final step of this function is to filter the set of returned nodes according to filtering parameter $\phi$, which ensures nodes from a given path added to $\mathcal{V}^+$ are separated by a distance of at least $\phi$. The path is traversed, and a node from the unfiltered set is selected every $\phi$ metres. For each pair of search patterns, only vertices that have not been seen before are added to $\mathcal{V}^+$.

As a final vertex selection step, ModifiedHittingSet takes as input the set of filtered shortest path vertices, $\mathcal{V}^+$, and solves the integer program (IP) detailed by Eqns. (1) through (6), similar to a previous approach based on geometric hitting sets [23]. The solution to the IP selects the set $\mathcal{V}^F \subseteq \mathcal{V}^+$ of recharge vertices. Variable $\alpha_i$ is 1 if vertex $v_i$ is selected, while $\beta_{ij}$ is 1 if vertex $v_i$ is assigned to selected vertex $v_j$. $\Delta_{ij}^{Euc}$ is the euclidean distance between vertices $v_i$ and $v_j$, and $R$ is the maximum permissible distance between a selected vertex and a vertex assigned to

it. We use the shorthand $i \in \mathcal{V}$ to indicate vertex $v_i \in \mathcal{V}$ in the remainder of the paper.

$$\min \quad \sum_{i \in \mathcal{V}^+} \alpha_i \tag{1}$$

$$\text{subject to:} \quad \sum_{j \in \mathcal{V}^+} \beta_{ij} = 1 \qquad \forall i \in \mathcal{V}^+ \tag{2}$$

$$\Delta_{ij}^{Euc} \beta_{ij} \leq R \alpha_j \qquad \forall i, j \in \mathcal{V}^+ \tag{3}$$

$$\alpha_i = 1 \qquad \forall i \in \mathcal{V}^+ \mid i \in \mathcal{V}^C \tag{4}$$

$$\alpha_i \in \{0, 1\} \qquad \forall i \in \mathcal{V}^+ \tag{5}$$

$$\beta_{ij} \in \{0, 1\} \qquad \forall i, j \in \mathcal{V}^+ \tag{6}$$

The IP model finds the minimum number of selected vertices subject to constraints. Constraint (2) ensures each vertex is assigned to a selected vertex. Constraint (3) dictates that a vertex can only be assigned to a selected vertex if it is less than $R$ from the selected vertex. Constraint (4) ensures that vertices coinciding with search pattern locations are always selected, while the remaining constraints dictate variable domains. Selected vertices $v_i$ with $\alpha_i = 1$ in the optimal solution to the IP are added to $\mathcal{V}^F$.

The final routing graph, $\mathcal{G} = (\mathcal{V}_{0,N+1}, \mathcal{E})$, consists of vertices $\mathcal{V}_{0,N+1} = \{v_0, v_{N+1}\} \cup \mathcal{V}^C \cup \mathcal{V}^F$. The set of non-depot vertices is defined as $\mathcal{V} = \mathcal{V}^C \cup \mathcal{V}^F$. Indices are used to specify sets that contain instances of the depot, e.g., $\mathcal{V}_0 = \{v_0\} \cup \mathcal{V}$, $\mathcal{V}_{N+1} = \mathcal{V} \cup \{v_{N+1}\}$, and $\mathcal{V}_0^F = \{v_0\} \cup \mathcal{V}^F$. The edge set is defined as $\mathcal{E} = \{(i, j) : i \in \mathcal{V}_0, j \in \mathcal{V}_{N+1}, i \neq j\}$.

## V. Target Search Optimization Models

In this section, we present both MILP and CP models for solving the UAV and MRV routing problem over the generated routing graph, $\mathcal{G}$. Recall that $\mathcal{V}^C$ are the vertices pertaining to search pattern executions and $\mathcal{V}^F$ those that pertain to possible recharge opportunities.

### A. Mixed-Integer Linear Programming

Our MILP model is defined by Eqns. (7)-(26). Binary decision variable $x_{ij}$ is 1 if a UAV travels from $v_i$ to $v_j$ and 0 otherwise. Binary decision variable $y_{ij}$ is 1 if a MRV travels from $v_i$ to $v_j$ and 0 otherwise. Continuous variable $\tau_i$ represents the arrival time of any vehicle at vertex $v_i$. Continuous variable $e_i$ indicates UAV energy level upon arrival at vertex $v_i$.

$$\max \sum_{i \in \mathcal{V}^C} \sum_{j \in \mathcal{V}_{N+1}} x_{ij} r_i \tag{7}$$

subject to:

$$\sum_{j \in \mathcal{V}} x_{0j} \leq |O| \tag{8}$$

$$\sum_{j \in \mathcal{V}^F} y_{0j} \leq |K| \tag{9}$$

$$\sum_{j \in \mathcal{V}_{N+1}, i \neq j} x_{ij} \leq 1 \qquad \forall i \in \mathcal{V} \tag{10}$$

$$\sum_{j \in \mathcal{V}_{N+1}^F, i \neq j} y_{ij} \leq 1 \qquad \forall i \in \mathcal{V}^F \tag{11}$$

$$\sum_{i \in \mathcal{V}_{N+1}, i \neq j} x_{ji} - \sum_{i \in \mathcal{V}_0, i \neq j} x_{ij} = 0 \qquad \forall j \in \mathcal{V} \tag{12}$$

$$\sum_{i \in \mathcal{V}_{N+1}^F, i \neq j} y_{ji} - \sum_{i \in \mathcal{V}_0^F, i \neq j} y_{ij} = 0 \qquad \forall j \in \mathcal{V}^F \tag{13}$$

$$\sum_{j \in \mathcal{V}_{N+1}} x_{ij} \leq \sum_{j \in \mathcal{V}_{N+1}^F} y_{ij} \qquad \forall i \in \mathcal{V}^F \tag{14}$$

$$\tau_0 + \tfrac{\Delta_{0j}^{Euc}}{s_O} x_{0j} - H(1 - x_{0j}) \leq \tau_j \qquad \forall j \in \mathcal{V}_{N+1} \tag{15}$$

$$\tau_i + (\tfrac{\Delta_{ij}^{Euc}}{s_O} + p_i) x_{ij} - H(1 - x_{ij}) \leq \tau_j \qquad \forall i \in \mathcal{V}^C, j \in \mathcal{V}_{N+1} \tag{16}$$

$$\tau_i + (\tfrac{\Delta_{ij}^{Euc}}{s_O} + \xi) x_{ij} - H(1 - x_{ij}) \leq \tau_j \qquad \forall i \in \mathcal{V}^F, j \in \mathcal{V}_{N+1} \tag{17}$$

$$\tau_0 + \Delta_{0j}^{SP} y_{0j} - H(1 - y_{0j}) \leq \tau_j \qquad j \in \mathcal{V}_{N+1}^F \tag{18}$$

$$\tau_i + (\Delta_{ij}^{SP} + \xi) y_{ij} - H(1 - y_{ij}) \leq \tau_j \qquad \forall i \in \mathcal{V}^F, j \in \mathcal{V}_{N+1}^F \tag{19}$$

$$e_i - (g\Delta_{ij}^{Euc} + h) x_{ij} + Q(1 - x_{ij}) \geq e_j \qquad \forall i \in \mathcal{V}^C, j \in \mathcal{V}_{N+1} \tag{20}$$

$$Q - g\Delta_{ij}^{Euc} x_{ij} + Q(1 - x_{ij}) \geq e_j \qquad \forall i \in \mathcal{V}_0^F, j \in \mathcal{V}_{N+1} \tag{21}$$

$$t_i^- \leq \tau_i \leq t_i^+ \qquad \forall i \in \mathcal{V}^C \tag{22}$$

$$0 \leq e_i \leq Q \qquad \forall i \in \mathcal{V}_{N+1} \tag{23}$$

$$\tau_i \geq 0 \qquad \forall i \in \mathcal{V}_{0,N+1} \tag{24}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in \mathcal{V}_0, j \in \mathcal{V}_{N+1} \tag{25}$$

$$y_{ij} \in \{0, 1\} \qquad \forall i \in \mathcal{V}_0^F, j \in \mathcal{V}_{N+1}^F \tag{26}$$

Objective (7) maximizes the total accumulated reward of the UAV search routes. Constraints (8) and (9) ensure the number of UAVs and MRVs routed are limited by their respective fleet sizes. Constraints (10) through (13) enforce the flow through the graph for both UAVs and MRVs. Constraint (14) ensures that if a UAV visits a recharge vertex, it must be also visited by an MRV. By the definition of $\tau_i$, all vehicles that visit vertex $v_i$ are synchronized in time. Constraints (15) and (16) dictate UAV arrival time at a vertex $v_j$ when the preceding vertex $v_i$ is the depot or a search pattern, while Constraint (17) does the same when the preceding vertex is a recharge vertex. Similarly, Constraints (18) and (19) enforce arrival times at vertices for MRVs. Note that a solution to the model may have a UAV or MRV waiting between vertex visits. The remaining energy of a UAV at vertex $v_j$ given the previous vertex $v_i$ was a search pattern is given by Constraint (20), while Constraint (21) details the remaining energy if the previous vertex $v_i$ was the depot or a recharge visit. Energy is consumed due to travel between vertices and search pattern execution (no energy is consumed while waiting). Finally, Constraints (22)-(26) express the various domains of the decision variables. Given a solution, the routes of the vehicles can be efficiently determined based on the binary decision variable values. The model contains $|\mathcal{V}_{0,N+1}|^2 + |\mathcal{V}_{0,N+1}^F|^2$ binary variables and $2 \cdot |\mathcal{V}_{0,N+1}|$ continuous variables.

### B. Constraint Programming

Our second approach uses constraint programming (CP). CP is an alternative technology to MILP, employing branch-and-infer tree search. It has been successfully applied to UAV-routing problems in recent work [28], [15], [14], [29], however, these do not consider the routing of a UAV fleet in tandem with a supporting MRV fleet for mobile recharging. As with existing work [26], [30], we use *optional interval*, *sequence*, and *cumulative function expression* variables.

*a) Optional Interval Variables:* Variables whose possible values are a convex interval: $\{\perp\} \cup \{[a, b) | a, b \in \mathbb{Z}, a \leq b\}$, where $a$ and $b$ are the start/end values of the interval and $\perp$ is a special value indicating the variable is absent in the solution. The presence and start time of an optional interval variable, $var$, can be expressed within a

CP model using $\textsc{Pres}(var) \in \{0,1\}$ and $\textsc{Start}(var)$. We use the notation $\textsc{IntervalVar}(p,[a,b])$ to define mandatory interval variables in our models (and $\textsc{OptIntervalVar}$ if the task is optional), where $p$ is the task processing time (duration). Model constraints are only enforced over present interval variables. For our problem, interval variables are used to represent search pattern and recharge tasks.

*b) Sequence Variables:* These variables are useful for expressing model constraints over a permutation of interval variables. Given a sequence variable, $\pi$, defined on a set of interval variables, various constraints can be expressed. The expression $\textsc{Prev}_\pi(var)$ returns the interval variable previous to $var$ in the sequence $\pi$. We also use the $\textsc{NoOverlap}(\pi, tt)$ constraint, which ensures the present interval variables in $\pi$ do not overlap, while considering a set of specified transition times between pairs of tasks, $tt$. For the studied problem, sequence variables represent a permutation of search pattern and recharge tasks.

*c) Cumulative Function Expressions:* These variables represent the usage of a renewable resource over time as the sum of interval variable contributions. With a cumulative function expression variable, $f$, we can express impact on the expression using the $f \pm \textsc{StepAtStart}(var, impact)$ expression, specifying that at the start of interval variable $var$, function $f$ is incremented or decremented by $impact$ (where $impact$ can be a range). The constraint $\textsc{AlwaysIn}(f,[a,b),[min,max])$ dictates that $min \leq f \leq max$ holds for all time points in $a$ up until, but not including, $b$, and a similar constraint $\textsc{AlwaysIn}(f, var, [min,max])$ ensures that $min \leq f \leq max$ holds during the processing of interval variable $var$. In the studied problem, cumulative expression variables are used to represent vehicle energy constraints in a way similar to previous work [31], [32].

*d) Proposed Model:* Our proposed CP model is defined by Constraints (27) through (44) and follows the single resource transformation technique recently proposed for electric vehicle routing to leverage the homogeneity of UAV and MRV fleets [26]. This transformation compactly represents the routes of multiple vehicles of the same type with a single augmented horizon (Figure 2). Given a solution to the model, the assignment of tasks to vehicles can be efficiently determined based on the start time of each task.

This technique uses a set of auxiliary depot instance vertices, $\mathcal{H} = \{v_{N+2}, \ldots, v_{N+\max(|O|,|K|)}\}$, to represent end depots of the additional horizon segments. Vertex and edge sets are extended to include these auxiliary vertices (e.g., $\mathcal{H}_{0,N+1} = \{v_0, v_{N+1}\} \cup \mathcal{H}$, $\mathcal{V}_{0,N+1,\mathcal{H}} = \mathcal{V}_{0,N+1} \cup \mathcal{H}$, and $\mathcal{E}' = \{(i,j) \mid i,j \in \mathcal{V}_{0,N+1,\mathcal{H}}, i \neq j\}$). A depot instance, represented as an interval variable, $x_i$, is assigned with null duration for $i \in \mathcal{H}_{0,N+1}$. These variables have fixed start time $\sigma_i$ such that $\sigma_0{=}0, \sigma_{N+1}{=}H, \sigma_{N+2}{=}2H$, etc. Then, optional interval variable $x_i$, for $i \in \mathcal{V}$, represents a visit to vertex $v_i$ by a UAV in the fleet. Similarly, optional interval variable $y_i$ represents a visit to vertex $v_i$, for $i \in \mathcal{V}^F$, by an MRV. We let sequence variable $\pi$ represent the sequence of UAV visits, and sequence variable $\rho$ the sequence of MRV visits. Cumulative function expression variable $\mathcal{Q}$ represents the energy level throughout the augmented UAV schedule.

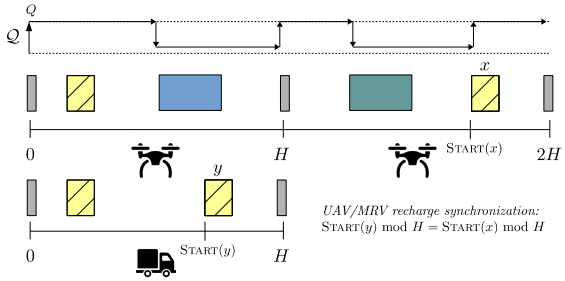



Fig. 2: CP model, single resource transformation technique. Two UAVs, $|O| = 2$, with augmented horizon ($2H$). Single MRV, $|K| = 1$. Each UAV conducts one search pattern (blue/green) and is recharged once (yellow) by the MRV.

$$\max \sum_{i \in \mathcal{V}^C} \textsc{Pres}(x_i) \cdot r_i \tag{27}$$

subject to:

$$\textsc{NoOverlap}(\pi, \{\tfrac{\Delta^{Euc}_{ij}}{s_O} : (i,j) \in \mathcal{E}'\}) \tag{28}$$

$$\textsc{NoOverlap}(\rho, \{\Delta^{SP}_{ij} : (i,j) \in \mathcal{E}'\}) \tag{29}$$

$$\textsc{ForbidStart}(x_i, \psi_i) \qquad \forall i \in \mathcal{V}^C \tag{30}$$

$$\mathcal{Q} = \textsc{StepAtStart}(x_0, Q) - \sum_{i \in \mathcal{V}^C} \textsc{StepAtStart}(x_i, g\Delta^{Euc}_{\textsc{Prev}_\pi(x_i),i} + h) + \sum_{i \in \mathcal{V}^F_{N+1,\mathcal{H}}} \textsc{StepAtStart}(x_i, [0, Q - g\Delta^{Euc}_{\textsc{Prev}_\pi(x_i),i}]) \tag{31}$$

$$\textsc{AlwaysIn}(\mathcal{Q}, [0, |O| \cdot H], [0, Q]) \tag{32}$$

$$\textsc{AlwaysIn}(\mathcal{Q}, x_i, [Q, Q]) \qquad \forall i \in \mathcal{V}^F \tag{33}$$

$$\textsc{Mod}(\textsc{Start}(y_i), H) = \textsc{Mod}(\textsc{Start}(x_i), H) \qquad \forall i \in \mathcal{V}^F \tag{34}$$

$$\textsc{Pres}(y_i) = \textsc{Pres}(x_i) \qquad \forall i \in \mathcal{V}^F \tag{35}$$

$$\textsc{First}(\pi, x_0), \textsc{Last}(\pi, x_{N+|O|}) \tag{36}$$

$$\textsc{First}(\rho, y_0), \textsc{Last}(\rho, y_{N+|K|}) \tag{37}$$

$$x_i : \textsc{IntervalVar}(0, [\sigma_i, \sigma_i]) \qquad \forall i \in \mathcal{H}_{0,N+1} \tag{38}$$

$$x_i : \textsc{OptIntervalVar}(p_i, [0, |O| \cdot H]) \qquad \forall i \in \mathcal{V}^C \tag{39}$$

$$x_i : \textsc{OptIntervalVar}(\xi, [0, |O| \cdot H]) \qquad \forall i \in \mathcal{V}^F \tag{40}$$

$$y_i : \textsc{IntervalVar}(0, [\sigma_i, \sigma_i]) \qquad \forall i \in \mathcal{H}_{0,N+1} \tag{41}$$

$$y_i : \textsc{OptIntervalVar}(\xi, [0, |K| \cdot H]) \qquad \forall i \in \mathcal{V}^F \tag{42}$$

$$\pi : \textsc{SequenceVar}(\{x_0, \ldots, x_{N+|O|}\}) \tag{43}$$

$$\rho : \textsc{SequenceVar}(\{y_0, \ldots, y_{N+|K|}\}) \tag{44}$$

Objective (27) maximizes accumulated search reward. Constraint (28) uses the $\textsc{NoOverlap}$ constraint to ensure UAV tasks do not interfere temporally, with consideration for travel times, and Constraint (29) enforces the same restriction for MRVs. Constraint (30) uses the $\textsc{ForbidStart}$ constraint to ensure that UAV search pattern tasks can only start during search pattern time windows; the set of forbidden starting times for $x_i$ is denoted by $\psi_i = \{0, \ldots, |O| \cdot H\} \setminus \bigcup_{\gamma \in \{0,\ldots,|O|-1\}} \{\gamma H + t_i^-, \ldots, \gamma H + t_i^+\}$. Constraint (31) dictates the effects of consumption and replenishment of UAV energy on the cumulative function expression variable $\mathcal{Q}$ (with a slight abuse of notation, $\Delta^{Euc}_{\textsc{Prev}_\pi(x_i),i}$ returns the Euclidean distance between the visit previous $x_i$ to

$x_i$). Constraint (32) ensures that throughout the entire augmented UAV planning horizon, $[0, |O| \cdot H]$, UAV battery level always remains within $[0, Q]$. Constraint (33) ensures that when a UAV swaps a battery, the battery is recharged completely. Constraint (34) ensures that UAV and MRV recharge tasks are synchronized using the MOD constraint based on the modulo operation. MOD(START($var$), $H$) returns the remainder when START($var$) is divided by the planning horizon, effectively yielding the non-augmented start time of the task. Constraint (35) enforces that both a UAV and MRV must be present for a recharge. The depot visits are constrained to be at the start and end of vehicle sequences through Constaints (36) and (37), respectively. The remainder of the Constraints (38) through (44) indicate the domains of the decision variables. The model contains $|\mathcal{V}_{0,N+1,\mathcal{H}}| + |\mathcal{V}^F_{0,N+1,\mathcal{H}}|$ interval variables, two sequence variables, and one cumulative function expression variable.

## VI. Experimental Evaluation

In this section we conduct an empirical investigation of the proposed routing graph construction pipeline and target search optimization models. We evaluate the relative performance of the models presented, and discuss whether our approaches are reasonable for use in real-time situations. To assess the advantages of using MRVs in tandem with the UAV fleet, we examine optimization results for *non-range constrained* (NRC), *range constrained no recharging* (RC-NC), and *range constrained with recharging* (RC-C) variants of the problem. NRC, when solved optimally, provides an upper-bound on the best possible solution to a given problem instance as it allows UAVs to fly without range constraints; in this case, accumulated reward is limited by the temporal aspects of the problem (i.e., time windows, transition times, and planning horizon). RC-NC, when solved optimally, provides a lower bound on the accumulated reward as each UAV has limited battery but is not able to recharge. The RC-C variant includes the fleet of MRVs and represents the range-constrained variant primarily studied in this work.

### A. Setup

The UAV and MRV routing graph construction pipeline is implemented in Python. All shortest path calculations use the NetworkX library [33] and the modified hitting set problem is solved using Gurobi 9.0 via the Python interface. All target search optimization model experiments are implemented in C++ and run on the Compute Canada Niagara computing cluster operated by SciNet (http://www.scinethpc.ca). The cluster runs the Linux CentOS 7 operating system and uses Skylake cores at 2.4 GHz. We use CP Optimizer for the CP models and CPLEX for the MILP model from the IBM ILOG CPLEX Optimization Studio version 12.9. All target search optimization model experiments are single-threaded with default inference settings and a time limit of one hour.

### B. Real-world Instances

For our simulations, we set UAV parameter values to follow the design specifications of the DJI Matrice 200, a popular and commercially-available rotary wing UAV [9]. UAV speed is set to 23 m/s, max range on a full charge is set to 17,940 metres (where energy consumption is assumed to be linear) following flight time with a loaded payload, and search pattern consumption is set to be the equivalent of the UAV traveling 3,450 metres. We investigate two battery swap durations: instantaneous, and 30 seconds, following existing research on automated battery swap technology [12]. We generate problem instances and run simulations for three problem classes: small, medium, and large as illustrated in Table I. Each problem class varies the number of UAVs and MRVs, as well as the scale of the road networks used. To construct our benchmark set, 20 instances are generated for each class, starting from real-world road networks in Scotland, upon which the problem graph is constructed (Figure 3), and ending with the routing graph via our pipeline. Each instance considers a different road network discretized with $\delta = 300$ metres, resulting in the average graph characteristics noted in Table I. As is detailed in the table, the scale of the road networks used for large instances is more than twice the size as those for small instances.

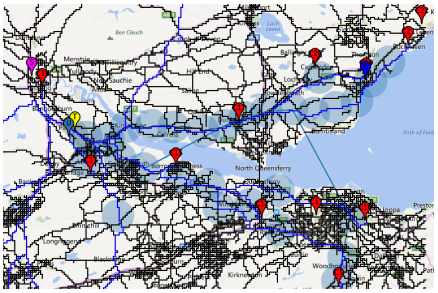


Fig. 3: Simulation instance problem graph derived from road network in Scotland. Search patterns (blue circles), destinations (red), target LKP (yellow), target actual start and end (purple/blue), UAV position (light blue).

TABLE I: Problem Instance Details

| Class | Small | Medium | Large |
|---|---|---|---|
| UAVs, $\|O\|$ | 1 | 3 | 5 |
| MRVs, $\|K\|$ | 1 | 2 | 3 |
| Graph vertices, $\|V\|$ (Avg.) | 7,645 | 10,042 | 16,756 |
| Graph edges, $\|E\|$ (Avg.) | 8,839 | 11,635 | 19,327 |

### C. Simulation Results

In this section we detail results for the routing graph construction pipeline and target search optimization.

*1) Routing Graph Construction:* The first phase of routing graph construction is running the MCS to identify search pattern locations and time windows, finalizing the problem graph, $G$. Following the existing approach [10], this phase was found to take an average of 5.2 seconds with a standard deviation of 0.6 seconds across all problems.

The second phase constructs the routing graph, $\mathcal{G}$, from the problem graph, $G$. For these experiments, we set filtering and coverage parameters, $\phi$ and $R$, respectively, to 25% of max UAV range, in metres, a value empirically found to yield sufficient filtering while providing strategic areas for

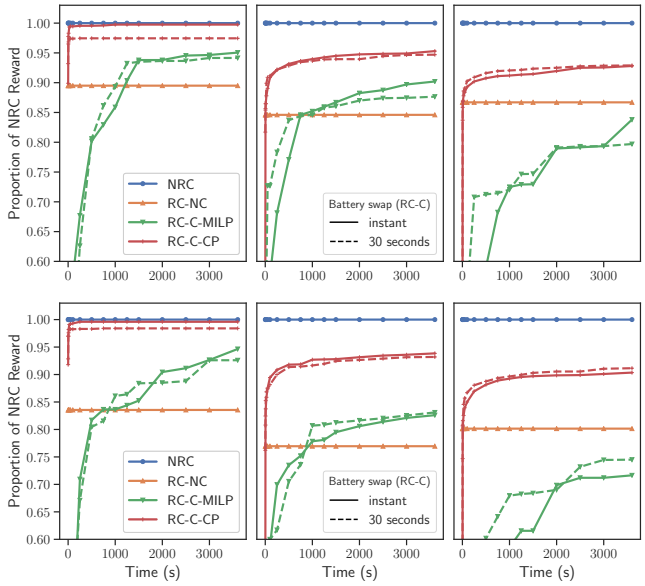



Fig. 4: Solution quality over time. Comparison of NRC, RC-NC, and RC-C optimization performance. All experiments given one hour of runtime. Plots represent proportion of reward compared to the optimal reward yielded by solving the NRC problem variant. RC-C results for instantaneous battery swap, and battery swap with duration of 30 seconds. *Top:* Experiments following DJI speed specifications. *Bottom:* 30% increase in UAV speed over specifications.

recharge vertex placement. Due to numerous shortest path calculations and the use of IP in ModifiedHittingSet, this step is quite sensitive to the size of the problem graph. For our experiments, the small and medium instances take, on average, roughly a minute to construct the routing graph, while the large instances can exceed five minutes. This suggests that, while the proposed pipeline is reasonable for use in real-time, small-to-medium sized situations, additional optimizations are required for large problems (e.g., using a heuristic to find the centres).

For each problem class, the produced routing graph, $\mathcal{G}$, varies in size. Small instances had routing graphs with an average of 44.4 vertices, medium instances with an average of 62.1 vertices, and large instances with an average of 216.7 vertices. Depending on the underlying topology of the instance, the routing graphs were found to include up to 20 strategically placed (i.e., not coinciding with a search pattern location) vertices to enable synchronized recharging.

*2) Target Search Optimization:* With the constructed routing graphs, we present target search simulation results using realistic DJI Matrice 200 speeds. These results are presented in Figure 4 (top). Each of the plots summarizes average results for the problem classes (e.g. small, medium, large). Each plot provides a time-based analysis capturing how closely the approaches approximate the optimal accumulated reward of the NRC variant. The NRC variant can be solved to optimality using MILP as it omits all of the energy constraints/variables and MRVs, making the problem considerably easier to solve. For similar reasons, the RC-NC variant can also be solved to proven optimality with MILP. At each time step, the average objective for a given method, as a proportion of the NRC objective, is plotted. For the RC-C experiments, plots with solid lines reflect optimization runs with instantaneous battery swaps, while those with dashed lines use a duration of 30 seconds.

In Figure 4 (top), the RC-NC variant (orange plot) indicates that deploying the UAV fleet on a single charge can attain from 85% to 90% of the NRC reward (blue plot). Therefore at realistic UAV speeds, there is less than 10-15% reward improvement to be gained by leveraging the MRVs. This observation is likely driven by the fact that, as a property of the instance, search patterns closer to the target's LKP are assigned larger rewards (and are likely to be reachable with a single charge); as candidate search areas get further away from the LKP, their reward values diminish to reflect a reduction in confidence that the target will be reacquired.

For strategies using MRVs (i.e., RC-C-MILP and RC-C-CP), it is clear that the CP model (red plots) is able to find the highest reward solutions overall, with average rewards exceeding the RC-NC baseline in less than 10 seconds of solver runtime, even for the largest problems, and approaching those of the NRC variant. The MILP method (green lines) exhibits reasonable performance for small instances, but only modestly improves on the RC-NC variant bound for medium class problems and is unable to improve upon it for large problems. Battery swap duration has relatively little effect on the CP approach for these experiments; for small problems, the solution quality is appreciably less when 30 second battery swaps are used (versus instantaneous swaps), however, for medium-to-large problems the effect is negligible. The impact of swap duration on the MILP approach is more pronounced, but does not drastically impact results. This finding is attractive as it indicates accumulated reward under realistic battery swap duration is not significantly eroded compared to an instantaneous swap scenario.

The results in Figure 4 (bottom) assess performance when UAV speed is increased to 30 m/s. We can see from the figure that the difference between the NRC "ideal" objective values and RC-NC lower bound objective values becomes more pronounced (20-25%) as UAV speed is increased. Faster UAVs are able to satisfy the time windows of more search pattern locations, thus improving the NRC objective, while still remaining range constrained in the RC-NC variant. The impact of battery swap duration on the CP approach remains neglible, while the MILP approach exhibits more pronounced variation. The primary assessment of the optimization methods remains consistent: the CP approach is the dominant method. This performance advantage is particularly apparent for large problems where CP finds solutions of 90% of the NRC rewards, while MILP struggles to improve over the RC-NC lower bound of 75-80%.

MILP is known to have a weak linear relaxation when constraints with large disjunctive constants ('big-M' constraints) are used, as in Constraints (15) through (21) of the MILP model. Possible techniques that can be investigated to im-

prove upon the baseline MILP approach include branch-and-price/branch-and-cut decompositions that have been applied to problems with similar characteristics [27], [16].

Overall, simulation results suggest that the inclusion of a fleet of MRVs can bolster UAV fleet search performance over a single charge bound, with cumulative reward values often exceeding 90% of the best-case NRC values. While we provide experimentation with UAV speed and battery swap duration in our analysis here, investigating the impact of other UAV characteristics (i.e., battery capacity) on performance represents possible future work. Furthermore, the speed at which the CP approach is able to find solutions exceeding the quality of the RC-NC baseline suggests that our overall approach (i.e., routing graph construction with target search optimization route planning) is promising for use in real-time, small-to-medium sized target search situations.

## VII. Conclusions

In this paper we studied a range-constrained variant of the multi-UAV target search problem, assessing the viability of commercially available UAVs for target search missions. We proposed a pipeline for representing the problem over real-world road networks and solved the problem using mixed-integer linear programming (MILP) and constraint programming (CP). Our empirical results indicate that CP is able to provide better solutions than MILP, overall, and that the use of a fleet of MRVs can significantly improve the accumulated reward of the UAV fleet.

## References


[1] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.

[2] G. Grenzdörffer, A. Engel, and B. Teichert, "The photogrammetric potential of low-cost UAVs in forestry and agriculture," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 31, no. B3, pp. 1207–1214, 2008.

[3] S. Bernardini, M. Fox, D. Long, and C. Piacentini, "Deterministic versus probabilistic methods for searching for an evasive target." in *AAAI*, 2017, pp. 3709–3715.

[4] B. Koopman, "Search and screening, operations evaluation group report 56," *Center for Naval Analysis, Alexandria, Virginia*, 1946.

[5] R. He, A. Bachrach, and N. Roy, "Efficient planning under uncertainty for a target-tracking micro-aerial vehicle," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 1–8.

[6] R. R. Pitre, X. R. Li, and R. Delbalzo, "UAV route planning for joint search and track missions—an information-value approach," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2551–2565, 2012.

[7] K. Kalyanam, D. Casbeer, and M. Pachter, "Graph search of a moving ground target by a UAV aided by ground sensors with local information," *Autonomous Robots*, pp. 1–13, 2020.

[8] T. J. Gleason and P. G. Fahlstrom, "Classes and missions of UAVs," *Encyclopedia of Aerospace Engineering*, pp. 1–10, 2010.

[9] L. SZ DJI Technology Co. (2020) Matrice 200 series specs. [Online]. Available: https://www.dji.com/matrice-200-series/info

[10] S. Bernardini, M. Fox, and D. Long, "Combining temporal planning with probabilistic reasoning for autonomous surveillance missions," *Autonomous Robots*, vol. 41, no. 1, pp. 181–203, 2017.

[11] C. Piacentini, S. Bernardini, and J. C. Beck, "Autonomous target search with multiple coordinated UAVs," *Journal of Artificial Intelligence Research*, vol. 65, pp. 519–568, 2019.

[12] B. Michini, T. Toksoz, J. Redding, M. Michini, J. How, M. Vavrina, and J. Vian, "Automated battery swap and recharge to enable persistent UAV missions," in *Infotech@ Aerospace 2011*, 2011, p. 1405.

[13] B. Bethke, M. Valenti, and J. P. How, "UAV task assignment," *IEEE robotics & automation magazine*, vol. 15, no. 1, pp. 39–44, 2008.

[14] Z. Tang, W.-J. van Hoeve, and P. Shaw, "A study on the traveling salesman problem with a drone," in *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Springer, 2019, pp. 557–564.

[15] A. M. Ham, "Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming," *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 1–14, 2018.

[16] S. G. Manyam, K. Sundar, and D. W. Casbeer, "Cooperative routing for an air-ground vehicle team–exact algorithm, transformation method, and heuristics," *IEEE Transactions on Automation Science and Engineering*, 2019.

[17] 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.

[18] B. D. Song, J. Kim, J. Kim, H. Park, J. R. Morrison, and D. H. Shim, "Persistent UAV service: An improved scheduling formulation and prototypes of system components," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 221–232, 2014.

[19] J. Kim and J. R. Morrison, "On the concerted design and scheduling of multiple resources for persistent UAV operations," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 479–498, 2014.

[20] E. Hartuv, N. Agmon, and S. Kraus, "Scheduling spare drones for persistent task performance under energy constraints," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 532–540.

[21] I. Hong, M. Kuby, and A. T. Murray, "A range-restricted recharging station coverage model for drone delivery service planning," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 198–212, 2018.

[22] D. Chauhan, A. Unnikrishnan, and M. Figliozzi, "Maximum coverage capacitated facility location problem with range constrained drones," *Transportation Research Part C: Emerging Technologies*, vol. 99, pp. 1–18, 2019.

[23] P. Maini and P. Sujit, "On cooperation between a fuel constrained UAV and a refueling ugv for large scale mapping applications," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2015, pp. 1370–1377.

[24] K. Yu, A. K. Budhiraja, S. Buebel, and P. Tokekar, "Algorithms and experiments on routing of unmanned aerial vehicles with mobile recharging stations," *Journal of Field Robotics*, vol. 36, no. 3, pp. 602–616, 2019.

[25] M. G. Kantor and M. B. Rosenwein, "The orienteering problem with time windows," *Journal of the Operational Research Society*, vol. 43, no. 6, pp. 629–635, 1992.

[26] K. E. C. Booth and J. C. Beck, "A constraint programming approach to electric vehicle routing with time windows," in *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Springer, 2019, pp. 129–145.

[27] G. Desaulniers, F. Errico, S. Irnich, and M. Schneider, "Exact algorithms for electric vehicle-routing problems with time windows," *Operations Research*, vol. 64, no. 6, pp. 1388–1405, 2016.

[28] A. M. Ham, "Drone-based material transfer system in a robotic mobile fulfillment center," *IEEE Transactions on Automation Science and Engineering*, 2019.

[29] G. Radzki, A. Thibbotuwawa, and G. Bocewicz, "UAVs flight routes optimization in changing weather conditions–constraint programming approach," *Applied Computer Science*, vol. 15, no. 3, 2019.

[30] P. Laborie, J. Rogerie, P. Shaw, and P. Vilím, "Ibm ilog cp optimizer for scheduling," *Constraints*, vol. 23, no. 2, pp. 210–250, 2018.

[31] J. Kinable, W.-J. van Hoeve, and S. F. Smith, "Optimization models for a real-world snow plow routing problem," in *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Springer, 2016, pp. 229–245.

[32] K. E. C. Booth, G. Nejat, and J. C. Beck, "A constraint programming approach to multi-robot task allocation and scheduling in retirement homes," in *International conference on principles and practice of constraint programming*. Springer, 2016, pp. 539–555.

[33] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.