Journal of Intelligent & Robotic Systems (2025) 111:12
https://doi.org/10.1007/s10846-024-02209-w

REGULAR PAPER

# Cooperative Multi-Agent Planning Framework for Fuel Constrained UAV-UGV Routing Problem

Md Safwan Mondal[1] · Subramanian Ramasamy[1] · James D. Humann[3] · James M. Dotterweich[2] · Jean-Paul F. Reddinger[2] · Marshal A. Childers[2] · Pranav A. Bhounsule[1]

Received: 17 August 2023 / Accepted: 30 November 2024


**Abstract**
Unmanned Aerial Vehicles (UAVs), adept at aerial surveillance, are often constrained by their limited battery capacity. Refueling on slow-moving Unmanned Ground Vehicles (UGVs) can significantly enhance UAVs' operational endurance. This paper explores the computationally complex problem of cooperative UAV-UGV routing for vast area surveillance, considering speed and fuel constraints. It presents a sequential multi-agent planning framework aimed at achieving feasible and optimally satisfactory solutions. By considering the UAV fuel limit and utilizing a minimum set cover algorithm, we determine UGV refueling stops. This, in turn, facilitates UGV route planning as the first step. Through a task allocation technique and energy-constrained vehicle routing problem modeling with time windows (E-VRPTW), we then achieve the UAV route in the second step of the framework. The effectiveness of our multi-agent strategy is demonstrated through the implementation on 30 different task scenarios across three different scales. This work provides significant insight into the collaborative advantages of UAV-UGV systems and introduces heuristic approaches to bypass computational challenges and swiftly reach high-quality solutions.

**Keywords** Multi-agent planning · VRP · UAV · UGV

✉ Md Safwan Mondal
mdsafwanmondal@gmail.com

Subramanian Ramasamy
sramas21@uic.edu

James D. Humann
james.d.humann.civ@army.mil

James M. Dotterweich
james.m.dotterweich.civ@army.mil

Jean-Paul F. Reddinger
jean-paul.f.reddinger.civ@army.mil

Marshal A. Childers
marshal.a.childers.civ@army.mil

Pranav A. Bhounsule
pranav@uic.edu

[1] Department of Mechanical and Industrial Engineering, University of Illinois Chicago, Chicago 60607, IL, USA

[2] DEVCOM Army Research Laboratory, Aberdeen Proving Grounds, Aberdeen 21005, MD, USA

[3] DEVCOM Army Research Laboratory, Los Angeles 90094, CA, USA

## 1 Introduction

Over the past decade, unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) have seen a surge in applications across various sectors, from military and civilian uses [1–3] to intelligence gathering, surveillance, reconnaissance tasks [4, 5], search and rescue operations [6], and agricultural activities [7]. UAVs, known for their cost-effectiveness, ease of control, and high maneuverability, are ideal for quickly scanning or surveying vast areas, yet their application to larger-scale problems is limited by their short battery lifespan and small payload capacity. In contrast, UGVs, equipped with a large payload capacity and extended battery life, can withstand lengthier task duration. Nevertheless, obstacles such as challenging ground terrain, limited visibility, and slower movement speed often compromise their efficacy, frequently resulting in incomplete task completion. To counteract these individual drawbacks, a cooperative routing strategy involving both UAVs and UGVs can be employed, enhancing the operation coverage speed and task endurance. For instance, in a persistent surveillance

Published online: 09 January 2025

problem, UAV can visit a set of distant task points while being periodically refueled by the UGV, which acts as a mobile refueling depot. Simultaneously, the UGV can cover task points along the road network, thus reducing the UAV's workload and ensuring the operation's swift completion. By focusing on such a **UAV-UGV cooperative vehicle routing problem**, this study aims to find approximate near-optimal solutions in quick runtime that account for the UAV's fuel constraint and the UGV's speed and terrain limitations. The goal is to devise strategic collaborative routes between the UAV and UGV that enable effective UAV recharging by the UGV, ensuring a complete visit to designated task points in the shortest possible time.

### 1.1 Related Works

Extensive research has been conducted on the cooperative routing of UAVs with ground vehicles across the field of Robotics and Transportation. In Transportation, the Truck-Drone coordinated delivery problem, analogous to the UAV-UGV cooperative routing problem is conceptualized as a variation of the Vehicle Routing Problem with Drones (VRP-D) [8] or the Traveling Salesman Problem with Drones (TSP-D) [9]. Here, Murray and Chu [10] introduced a Mixed Integer Linear Programming (MILP) formulation, supplemented by heuristics, for a hybrid truck-drone parcel delivery system optimized for last-mile delivery, leveraging the advantages of both truck and drone. Similarly, Chen et al. [11] explored path planning for heterogeneous robot systems, including UGVs and UAVs, for urban parcel delivery, where UGVs navigate along road networks and UAVs transfer parcels from UGVs to customers. Bouman et al. [12] contributed an exact solution approach to the Traveling Salesman Problem with Drones (TSP-D) using dynamic programming, capable of solving problems with up to 16 customers. Furthermore, Ha et al. [13], inspired by Murray and Chu [10], aimed to minimize the total operational cost of a drone-truck delivery system, proposing a MILP formulation and a heuristic called TPS-LS, providing a cost-focused perspective alongside the efficiency-driven approaches in drone-truck collaborative routing.

Similarly, in Robotics, Levy et al. [14] explored the routing problem for multiple fuel-constrained UAVs with access to several static recharging depots, employing rapid variable neighborhood descent (VND) and variable neighborhood search (VNS) heuristics to generate viable solutions for a large number of problem instances. Extending this problem for multiple fuel-constrained vehicles with multiple depots, Sundar et al. [15] developed a mixed-integer linear programming model (MILP), which they solved optimally using a standard MILP solver. In contrast to fixed recharging stations, Maini et al. [16] addressed a cooperative routing problem involving a single UAV-UGV system, where the UGV has the ability to recharge the UAV while in transit on a road. They proposed a greedy heuristic for determining the meeting points for recharging along the UGV route and later used a MILP model to solve both UAV and UGV routes. Continuing the work, Manyam et al. [17] examined the cooperative routing of air and ground vehicle teams considering communication constraints, framing the problem as a MILP model and developing a branch-and-cut algorithm to solve it optimally. Several researchers have delved deeper into the UAV-UGV cooperative vehicle routing problem, exploring it in a tiered, two-echelon manner [18]. For instance, Luo et al. [19] introduced a binary integer programming model, supplemented by two heuristics, to tackle the two-echelon cooperative routing problem. Similarly, Liu et al. [20] devised a two-stage, route-focused framework for a parcel delivery system utilizing a truck and a drone, aiming to optimize both the truck's primary route and the associated drone's aerial routes. To quickly generate a feasible solution, they developed a hybrid heuristic combining nearest-neighbor and cost-saving strategies. Robert B. Anderson [21] investigated the impact of different objective functions and the split vehicle routing problem (SVRP) over the classical VRP in the routing and control of unmanned aerial vehicles (UAVs) for payload delivery and aerial reconnaissance missions. Ropero et al. [22] presented a hybrid UAV-UGV system designed to cooperatively visit a set of target points, with the UAV periodically recharged by the UGV at refuel stops. They employed a classical combinatorial approach combined with Genetic algorithm metaheuristics to identify refuel stops in the preliminary stages and thereby solve the routing problem with a genetic algorithm. In a related problem, Seyedi et al. [23] and Lin et al. [24] proposed a scalable and robust approximation algorithm for persistent surveillance tasks by energy-constrained UAVs and UGVs. However, their methodologies rely on forming uniform UAV-UGV teams and partitioning the environment among them, which can compromise the solution quality of the cooperative routes.

Significant experimental efforts have been conducted in UAV routing problems. Nigam et al. [25, 26] explored scalable control techniques for UAVs engaged in persistent surveillance within uncertain stochastic environments using a hardware testbed. Frew et al. [27] demonstrated road following, obstacle avoidance, and convoy protection in flight tests involving two UAVs, while Jodeh et al. [28] provided an overview of cooperative control algorithms for heterogeneous UAVs developed by the Air Force Research Laboratories (AFRL). Extensive UAV flight testing has also been conducted in indoor lab-scale setups by Boeing's Vehicle Swarm Technology Laboratory (VSTL) [29, 30] and MIT's RAVEN laboratory [31]. Wu et al. [32] developed a prototype UAGVR (Unmanned Aerial and Ground Vehicle Recharging) system that addresses the limited battery life of UAVs by enabling on-demand wireless recharging via a UGV, demonstrating its feasibility in extending UAV operation time through experimental validation and the creation of a cooperative mission planning algorithm. In UAV routing, Yu et al. [33] focused on planning and executing UAV tours incorporating both stationary and mobile recharging stations, such as UGVs, to minimize mission time. This work is significant for its validation through both simulations and proof-of-concept experiments using a custom UAV and a Clearpath Husky UGV, effectively demonstrating the feasibility of these coordinated recharging strategies in real-world scenarios. Karapetyan et al. [34] demonstrated practical deployment in coverage path planning for an energy-constrained UAV and UGV, with the UGV acting as a mobile recharging station. Using a VOXL m500 drone and a Clearpath Jackal ground vehicle, their heuristic method significantly reduced rendezvous overhead compared to a greedy approach, showcasing the system's robustness from algorithm development to real-world execution.

In our previous works [35–37], we explored a hierarchical, bi-level optimization framework for the cooperative routing of multiple fuel-limited UAVs and a single UGV. The outer level of this framework employed K-means clustering to determine UGV visit points, which were then connected using a Traveling Salesman Problem (TSP) approach to establish the UGV route. On the inner level, using the determined UGV path, we formulated and solved a vehicle routing problem that accounted for capacity constraints, time windows, and dropped visits for the UAV. Further expanding on this work, we demonstrated that optimizing heuristic parameters using Genetic Algorithm (GA) and Bayesian Optimization (BO) methods could lead to substantial improvements in solution quality [38, 39]. Given the intricacy of this problem, exact methods of solving this combinatorial optimization problem or generalizing a solution framework for diverse scenarios pose significant challenges. In this research endeavor, we propose a generalized multi-agent cooperative framework for addressing this fuel-constrained UAV-UGV cooperative routing problem. The key contribution of our study is the creation of **a heuristics-based, multi-staged framework** aimed at facilitating a rapid solution to the two-echelon UAV-UGV routing problem, considering fuel and speed constraints. To this end, our novel contributions include the following:

1. The proposed framework utilizes a sequential optimization strategy with a task allocation technique. Coupled with the constraint programming-based formulations, it can provide an effective solution for fuel constrained UAV-UGV cooperative routing problems within a quick time.
2. A task allocation technique based on the minimum set cover algorithm is proposed, which breaks down the entire problem into smaller subproblems, leading to a substantial simplification of the problem-solving process.
3. Our formulation of a constraint programming-based vehicle routing problem accommodates time windows, and fuel constraints, thereby enabling swift solutions for each subproblem.
4. We present extensive computational results on different kinds of scenarios to affirm the effectiveness and robustness of our proposed framework. This underscores the practicality of our framework in a diverse set of real-world applications.

The rest of the article is structured as follows: Section 2 presents the problem statement, Section 3 illustrates the framework methodology and solution heuristics. Section 4 introduces the experiments of different random instances in three different scales and shows the results part, offering a concrete view of our findings. The results are analyzed in Section 5 and finally, Section 6 presents the conclusion and outlines future works.

## 2 Problem Description

The problem objective is to configure an optimal cooperative route for a team comprising a UAV and UGV to visit a set of $n$ task points $\mathcal{M}_n = \{m_0, m_1, ..., m_n\}$ in a euclidean space (see Fig. 3a). The UAV $A \equiv (v^a, F^a, \mathcal{P}^a)$ and the UGV $G \equiv (v^g, F^g, \mathcal{P}^g)$ have heterogeneous vehicle characteristics; with the UAV having a higher velocity, i.e. $v^a > v^g$ but lower fuel capacity than that of the UGV, i.e. $F^a < F^g$. They also differ in their power consumption profiles (Eqs. 1 and 2), with the UAV demonstrating greater energy efficiency per unit distance traversal when operating at standard speeds (see Fig. 1).

$$\mathcal{P}^a = 0.0461{v^a}^3 - 0.5834{v^a}^2 - 1.8761v^a + 229.6 \tag{1}$$
$$\mathcal{P}^g = 464.8v^g + 356.3 \tag{2}$$

For the study, we specifically consider a multi-rotor UAV, and a large wheeled robot as UGV, with vehicle parameters modeled after the existing works [40, 41]. Furthermore, the proposed framework can accommodate other types of UAVs and UGVs with proper modeling of their characteristics. For instance, the framework can be extended to include quadrupedal-legged robots as UGVs or fixed-wing drones and VTOL (Vertical Take-Off and Landing) hybrid drones as UAVs, acknowledging their fuel efficiency for extended endurance. The task points can be visited by a free flyover of the UAV, $\tau^a$ or a visit by the UGV along the road network, $\tau^g$.

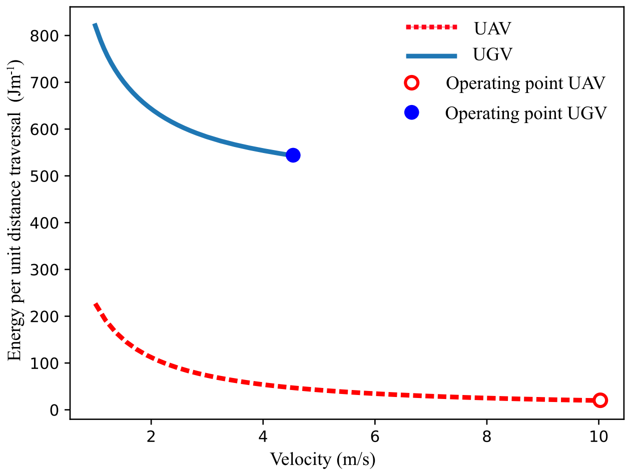



**Fig. 1** Energy consumption per unit distance traversal of UAV & UGV

The cost of travel between a pair of task points is equal to the time of traversal between them $t_{ij} = t_j - t_i$. Both the UAV and UGV commence their journeys from the same starting depot and return to it upon completion. The total task duration is the time span from when the first vehicle departs the depot until the last one returns. Due to having a limited battery capacity, the UAV has to get recharged periodically from the UGV, which acts as a mobile recharging depot, or from the starting depot (which acts as a fixed recharging depot) besides visiting the task points. The recharging time of the UAV at the UGV is not instantaneous, it depends upon the amount of fuel consumed by the UAV. Since the fuel capacity of the UGV is significantly larger compared to the UAV, it is assumed to be infinite to simplify the problem.

With all these above configurations, we have to find the time optimal cooperative route, $\tau = \tau^a \cup \tau^g$ between the UAV and UGV for visiting all the task points at least once; given the UAV will never run out of fuel. A typical sequence for this cooperative route can be as follows: Both the UAV and UGV commence their journey from the starting depot and visit several task points. As they proceed, the UGV will reach an appropriate location to recharge the UAV. After recharging, both vehicles will resume their task, continuing to visit task points until they reach the next recharging stop. This pattern will continue until all task points have been visited, after which both the UAV and UGV will return to the starting depot to conclude their task. However, for an optimal cooperative route, it is important to figure out:

1. Suitable refueling stop locations $\mathcal{M}_r = \{m_0^r, m_1^r, ..., m_n^r\}$, **where** the UAV and UGV will rendezvous for recharging.
2. Appropriate time intervals during the task **when** the UAV and UGV will meet at the refuel stops i.e, i.e, $t_i^r \ \forall \ m_i^r \in \mathcal{M}_r$.
3. Optimal routes for the UAV, $\tau^a$ and UGV, $\tau^g$ based on the determined refuel stop locations $m_i^r$ and time intervals $t_i^r$ to cover the entire task scenario in the quickest possible time.

## 3 Methods

We have devised a two-tiered optimization framework (as depicted in Fig. 2) for executing this fuel-constrained cooperative routing between the UAV and UGV. This framework is inspired by the "UGV First, UAV Second" heuristic approach for UAV-UGV cooperative routing [22, 42].

At the first stage of this framework, we utilize a ***UGV-Planner*** to establish the route $\tau^g = (X^g, T^g)$ for the UGV. Here, $X^g$ and $T^g$ are the spatial and temporal components of the UGV route $\tau^g$. This route is constructed by identifying appropriate recharging stations $\mathcal{M}_r$ and formulating the UGV's movement along the road network accordingly. The UGV's navigation is a combination of two-step processes. The initial phase involves movement along waypoints on the road network to cover the task points, while the second phase necessitates waiting for the UAV at the recharging stops.

At the second tier of the framework, the ***UAVPlanner*** devises the route $\tau^a = (X^a, T^a)$ for the UAV. Here, $X^a$ and $T^a$ are the spatial and temporal components of the UAV route $\tau^a$. The formation of this UAV route significantly relies on the UGV route $\tau^g$ created at the outer level of the framework. Because of the slower speed of the UGV, the ***UAVPlanner*** takes into consideration the *availability* time window constraint at the refuel stops. This planning approach effectively divides the entire scenario into a series of manageable subproblems, each of which can be solved by modeling it as an energy-constrained vehicle routing problem with time window constraints (E-VRPTW).

### 3.1 Ugv Routing

At the outer level of the proposed framework, the initial objective is to determine suitable recharging rendezvous locations $\mathcal{M}_r$ for the UAV-UGV system. Subsequently, an optimal route $\tau^g$ is generated for the UGV, taking into account the refuel stop locations $\mathcal{M}_r$ and its operational speed $v^g$ in the ***UGVPlanner***. Previous research by Maini et al.[16, 43] emphasized the importance of including refueling stops within the UAV's fuel coverage radius to ensure a viable route in fuel-constrained cooperative routing problems. It was also noted that minimizing the number of recharging instances can reduce the time spent on recharging and minimize the detour required in the UAV's route, resulting in a faster cooperative route. With these aforementioned considerations, we implement the minimum set cover algorithm (MSC) to find out minimum number of refueling stops and

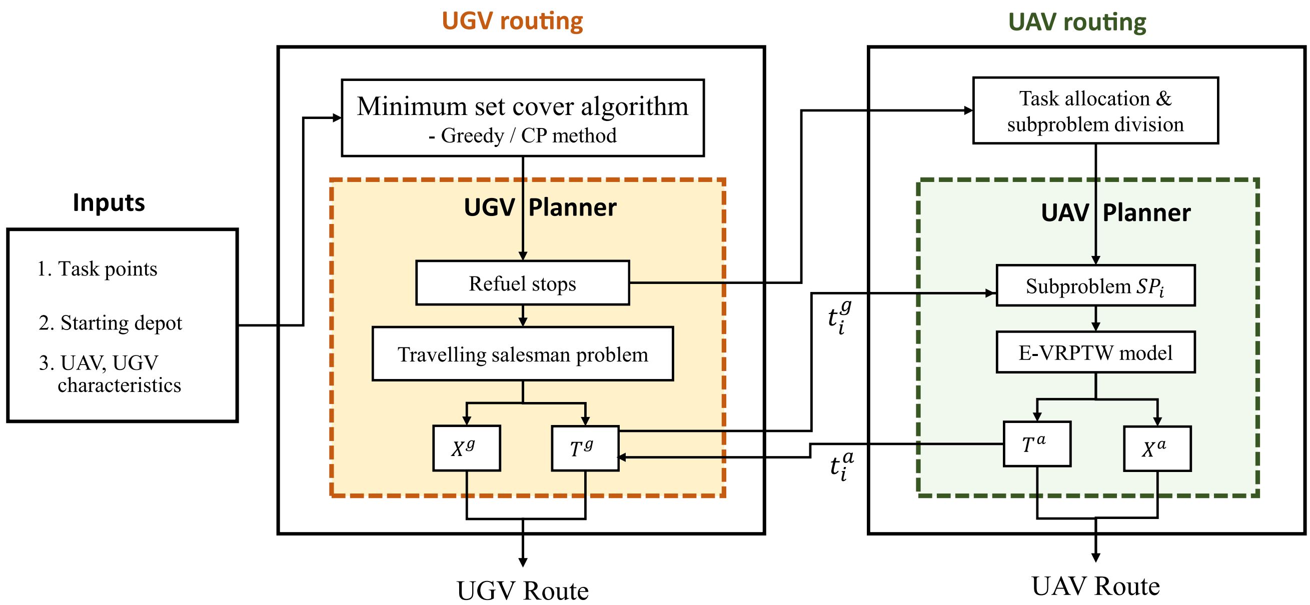



**Fig. 2** Proposed bilevel optimization framework for UAV-UGV cooperative routing problem

their locations $\mathcal{M}_r$ that cover the entire task scenario. The minimum set cover problem has been extensively studied [44, 45], and various methods, including greedy approaches [16, 46] have been proposed. However, we come up with an alternative constraint programming formulation for solving the minimum set cover problem in the context of the cooperative routing problem.

#### 3.1.1 Minimum Set Cover Algorithm

1. **Greedy heuristics approach** :
   The minimum set cover algorithm is an NP-hard problem; however, through greedy heuristics, the complexity of the problem can be reduced significantly. In the context of the cooperative routing problem to find optimal refuel stops through the greedy algorithm, we start with the task points $\mathcal{M}_n$ that needed to be covered, the fuel capacity $F^a$ of the UAV and the starting depot $m_0$ of the scenario. Our goal is to obtain the smallest possible subset of $\mathcal{M}_n$ that can act as refueling stops $\mathcal{M}_r$. As shown in Algorithm 1, the greedy algorithm includes the stating depot $m_0$ as the first refueling stop $m_0^r$, then it sequentially adds the task points $m_i$ which covers the maximum number of other task points into the refueling stop set $m_i^r$ until all the points are covered (lines 4-8).
   Greedy heuristics can quickly generate optimal result for a minimum set cover problem. But in many situations, a minimum set cover problem can have multiple optimal results for a particular scenario; because we are implementing a bilevel optimization framework, it is important to take into account the other optimal solutions of the outer level algorithm. As it is not possible to acquire all the optimal solutions through greedy heuristics, we used the constraint programming method, which can generate multiple optimal results (if any) in a quick span of time.

**Algorithm 1** Greedy Minimum Set Cover Algorithm

**Input:** task points $\mathcal{M}$, UAV fuel limit $F^a$, starting depot $m_0$;
**Output:** Refueling stops $\mathcal{M}_r$;
1: Initialize $\mathcal{M}_r = \{m_0^r = m_0\}$, Tasks = $\mathcal{T} = \mathcal{M}$;
2: $C_0 = Covered(m_0^r) = \{m_i : m_i \in \mathcal{T}$ and $\|m_i - m_0^r\| < 0.5F^a\}$;
3: $\mathcal{T} = \mathcal{T} \setminus C_0$;
4: **while** $\mathcal{T} \neq \emptyset$ **do**
5: $\quad m_{i_{\max}}^r = \arg\max\ Covered(m_i^r)$;
6: $\quad \mathcal{M}_r = \mathcal{M}_r \cup \{m_{i_{\max}}^r\}$;
7: $\quad C_{\max} = \{m_i : m_i \in \mathcal{T}$ and $\|m_i - m_{i_{\max}}^r\| < 0.5F^a\}$;
8: $\quad \mathcal{T} = \mathcal{T} \setminus C_{\max}$;
9: **end while**

2. **Constraint programming method :**
   To determine the minimum number of refueling stops $\mathcal{M}_r$ required to cover the entire task scenario ($\mathcal{M}$), we employ linear integer programming and utilize a constraint programming method (CP method) for solving. The problem is modeled using binary decision variables, $x_j$ (indicating whether a task point is chosen as a refueling stop) and $y_{ij}$ (indicating whether a task point $m_i$ is assigned to a refueling stop $m_j^r$). The objective function (Eq. 3) aims to minimize the total number of refueling stops. Constraint Eq. 4 ensures that each task point $m_i$ is assigned at least one refueling stop $m_j^r$. Constraint Eq. 5 ensures that a task point $m_i$ can be allocated to a refueling stop $m_j^r$ only if the refueling stop is selected. Furthermore, constraint Eq. 6 guarantees that a task point $m_i$ is assigned to a refueling stop $m_j^r$ only if the refueling stop

falls within the fuel coverage radius of the UAV, allowing for a round trip from the refueling stop.

$$\text{Objective: } \min \sum_{m_j^r \in \mathcal{M}_r} x_j \tag{3}$$

Subject to,

$$\sum_{m_j^r \in \mathcal{M}_r} y_{ij} \geq 1, \ \forall\, m_i \in \mathcal{M} \tag{4}$$

$$y_{ij} \leq x_j, \ \forall\, m_i \in \mathcal{M} \text{ and } \forall\, m_j^r \in \mathcal{M}_r \tag{5}$$

$$y_{ij} = 0, \ \text{if } d_{ij} > 0.5F^a, \ \forall\, m_i \in \mathcal{M} \text{ and } \forall\, m_j^r \in \mathcal{M}_r \tag{6}$$

$$y_{ij}, x_j \in \{0, 1\} \tag{7}$$

We use Google's OR-Tools Constraint programming solver ( CP-SAT solver [47]) to solve the above linear integer formulation. It is possible to record all the solutions if there are multiple optimal solutions through the solver. Once, the optimal refuel stops $\mathcal{M}_r$ are obtained from the MSC algorithm, it is sent to the ***UGVPlanner*** to construct the UGV route $\tau^g = (X^g, T^g)$ based on it.

#### 3.1.2 UGV Planner

Upon identifying the refueling stop locations $\mathcal{M}_r$ using the minimum set cover algorithm, the ***UGVPlanner*** proceeds to map out a feasible UGV route for the overall task through a sequential phase process (see Algorithm 2). Initially, it connects the refueling stops optimally on the road network by solving a simple Travelling Salesman Problem (TSP), which yields the spatial components $X^g \in \tau^g$ of the UGV route, denoting the sequence $x_i^g$ in which the task points on the road network will be visited. Next, the planner calculates the temporal components $T^g \in \tau^g$ of the UGV route till the first refuel stop, which details the time instances at which the UGV will visit those task points. We operate under the assumption that the UGV will not wait at any task point, except at the refueling stops. Therefore, the arrival times at the task points are computed based on the UGV's constant operational speed $v^g$ (line 4). This also gives the UGV's arrival time at the refueling stops (line 7), which serve as an $availability$ time window constraint in the ***UAVPlanner***. Utilizing the UAV's arrival time at the first refuel stop and the recharging time $\mathcal{R}_t$ (contingent on the UAV's fuel consumption level) from the UAV's route sortie we can estimate the UGV's waiting time at the first refuel stop (line 9), which is taken into account when computing the temporal component of the UGV route up to the next refuel stop. This process is reiterated until the UGV arrives at the final refuel stop.

At the end of this process, the temporal components are integrated with their respective spatial components to provide a comprehensive UGV route, outlining the sequence in which the UGV visits the task points and their corresponding time instances.

**Algorithm 2** UGV Planner

**Input:** Refuel stops $\mathcal{M}_r \leftarrow MSC$, UGV velocity $v^g$, starting depot $m_0$
**Output:** UGV route $\tau^g = (X^g, T^g) = [(x_i^g, t_i^g)]$

```
1: UGV navigation waypoints X^g ← TSP(M_r, m_0, v^g)
2: UGV route starting instance τ^g = [(x_0^g, t_0^g)]
3: for x_i^g in X^g do
4:     t_i^g = t_{i-1}^g + (x_i^g − x_{i-1}^g) / v^g
5:     τ^g.append((x_i^g, t_i))
6:     if x_i^g ∈ M_r then
7:         send t_i^g → UAVPlanner
8:         t_i^a, R_t ← UAVPlanner
9:         t_i^g = t_i^a + R_t
10:        τ^g.append((x_i^g, t_i^g))
11:    end if
12: end for
```

### 3.2 Uav Routing

At the inner level of the proposed framework, we split the full task scenario into subproblems, taking information about the refuel stops provided by the ***UGVPlanner***. A task allocation technique is employed to assign distinct task points to each subproblem. These subproblems are then individually addressed by formulating them as Energy Constrained Vehicle Routing Problems with Time Windows (E-VRPTW).

#### 3.2.1 Allocation of Task Points

Given the scenario and the obtained refuel stops $\mathcal{M}_r$ from the MSC algorithm, we can divide the entire problem into $r - 1$ number of subproblems ($r$ = number of refuel stops with starting depot) with an assumption that UGV travels only between two refuel stops in each subproblem. For the subproblem $SP_i$, the origin node is refuel stop $m_{i-1}^r$ and the destination node is refuel stop $m_i^r$. The subproblems are assigned with separate task points. The UAV task points covered by the destination refuel stop $m_i^r$ are assigned to that subproblem $SP_i$. Only, for the first subproblem $SP_1$ the task points covered by both origin $m_0^r$ and destination node $m_1^r$ is assigned to it.

Figure 3 demonstrate the process of subproblem division and task allocation. Figure 3b shows the refuel stops

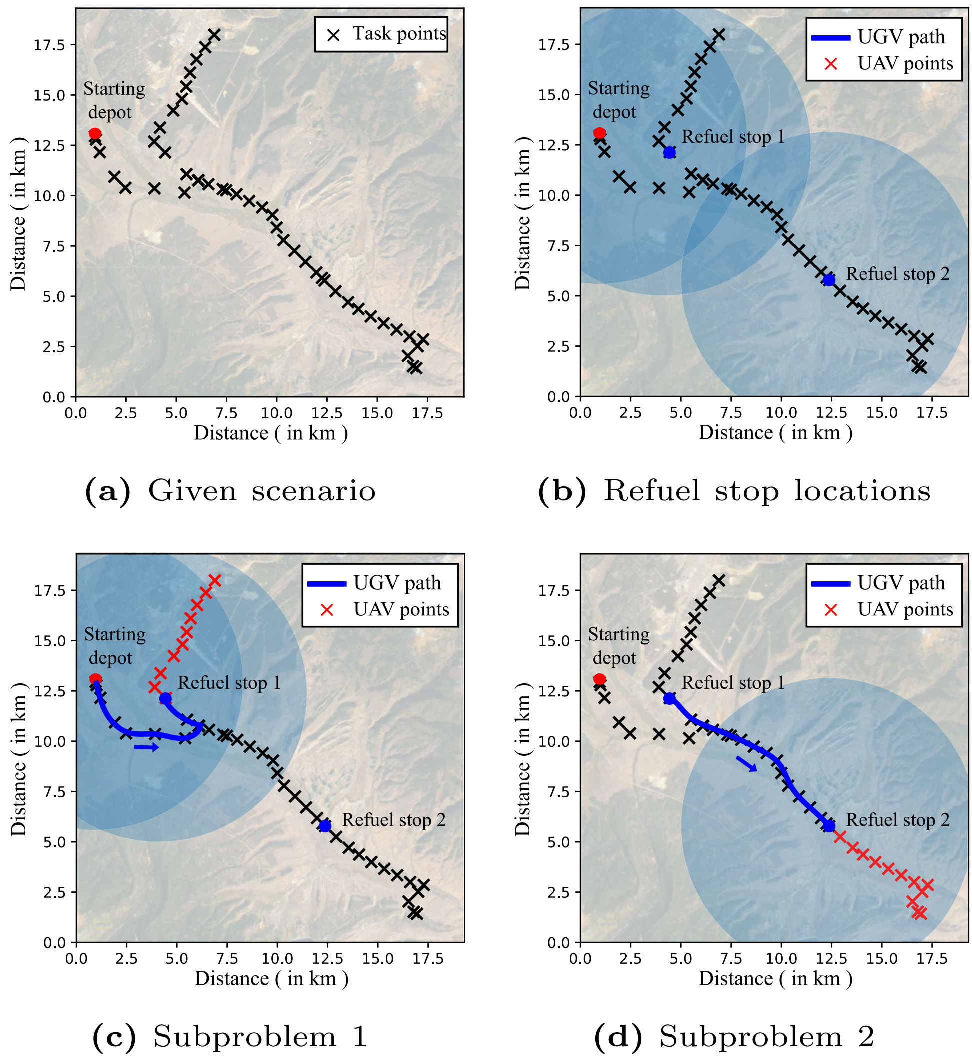



**Fig. 3** a) Given scenario with task points and starting depot b) Refuel stops in UGV route obtained from minimum set cover algorithm; the blue circles are indicating the radial coverage of the UAV c) Subproblem 1 with allocated UAV task points, here UGV travels between starting depot and refuel stop 1 d) Subproblem 2 with allocated UAV task points, here UGV travels between refuel stop 1 and refuel stop 2

obtained from minimum set cover algorithm which are taken into account for UGV route construction. Based on refuel stops, the first subproblem (Fig. 3c) is created by taking the starting depot as the origin node and the refuel stop 1 as the destination node. The UAV task points covered by origin node (starting depot) and destination node (refuel stop 1) are assigned for subproblem 1. Similarly, the second subproblem (Fig. 3d) is created by taking the refuel stop 1 as origin node and refuel stop 2 as destination node and the task points covered by the destination node (refuel stop 2) are assigned for this subproblem.

Now in the subproblems, the destination nodes $m_i^r$ have an *availability* time window constraint because UAV can recharge only when the UGV has already reached the refuel stops. This *availability* time period $t_i^g$ is obtained from the ***UGVPlanner*** and taken in account while modelling the subproblems as energy constrained vehicle routing problem with time window constraints (E-VRPTW).

### 3.2.2 E-VRPTW Formulation

The formulation of the E-VRPTW can be described with a graph theory. Consider an undirected graph $G = (V, E)$ where $V$ is the set of vertices $V = \{S, 0, 1, 2, ...D\}$ and $E$ is the set of edges between the vertices $i$ and $j$ as $E = \{(i, j) \parallel i, j \in V, i \neq j\}$. The non-negative arc cost between the vertices $i$ and $j$ is expressed as $t_{ij}$ and $x_{ij}$ is a binary decision variable whose value will be 1 if a vehicle travels from $i$ to $j$, and 0 otherwise. The UAV will start from refuel stop $S$ and meet the UGV at destination stop $D$. We then formulated the objective function of the E-VRPTW problem with fuel constraint, time window constraint, optional node constraints as follow:

$$\min \sum_i \sum_j t_{ij} x_{ij} \quad \forall i, j \in V \tag{8}$$

$$\sum_{j \in V} x_{ij} = 1 \quad \forall i \in V \setminus \{S, D\} \tag{9}$$

$$\sum_{i \in V} x_{ij} = 1 \quad \forall j \in V \setminus \{S, D\} \tag{10}$$

$$\sum_{j \in V} x_{Sj} = \sum_{i \in V} x_{iD} = 1 \tag{11}$$

$$f_j^a \leq f_i^a - \left(\mathcal{P}^a(v^a) t_{ij} x_{ij}\right) + L_1 \left(1 - x_{ij}\right) \quad \forall i, j \in V \setminus \{S, D\} \tag{12}$$

$$f_j^a = F^a \quad \forall j \in D \tag{13}$$

$$0 \leq f_j^a \leq F^a, \quad \forall j \in V \tag{14}$$

$$t_j \geq t_i + (t_{ij} x_{ij}) - L_2 \left(1 - x_{ij}\right) \quad \forall i, j \in V \tag{15}$$

$$t_{j,start} \leq t_j \leq t_{j,end}, \quad \forall j \in D \tag{16}$$

$$x_{ij} = 0, \quad \forall i \in D, \forall j \in V \tag{17}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in V \tag{18}$$

$$f_i > 0, f_i \in \mathbb{R}_+ \quad \forall i \in V \tag{19}$$

$$t_i > 0, t_i \in \mathbb{Z} \quad \forall i \in V \tag{20}$$

$$L_1, L_2 > 0, \quad L_1, L_2 \in \mathbb{R}_+ \tag{21}$$

The objective of Eq. 8 is to minimize the total time spent by the UAV. Constraints in Eqs. 9 and 10 represent flow conservation, where the inflow should equal the outflow at any of the task point vertices. Following that, constraint in Eq. 11 represents flow conservation for start and end vertices, where the number of UAVs leaving the start vertex must equal the number of UAVs arriving at the end vertex. The Miller-Tucker Zemlin (MTZ) formulation [48] for sub-tour elimination is the constraint in Eq. 12. The MTZ constraint ensures that each node is visited sequentially by keeping track of values such as fuel capacity and power consumption of the UAV corresponding to each node. It ensures that if a node is visited twice, the constraint is broken. This constraint allows the UAV's energy not to be fully drained out while eliminating loops. $L_1$ denotes a large number in this constraint. This constraint activates only when there is a flow between vertices $i$ and $j$ and drains the UAV energy based on the time taken between the two vertices. The $\mathcal{P}^a$ represents the UAV's power consumption curve during traversal.

According to constraint Eq. 13, if the vertex is the destination stop (recharging stop), the UGV must refuel the UAV to its full capacity $F^a$. Constraint Eq. 14 states that the UAV's fuel should be between 0 and maximum fuel capacity $F^a$ at any vertex in $V$. The cumulative arrival time at the $j^{th}$ node is equal to the sum of the cumulative time at the node $i$, $t_i$ and the travel time between nodes $i$ and $j$, $t_{ij}$. Here, $L_2$ is a large number that aids in the elimination of sub-tours in Eq. 15.

Equation 16 imposes a time window constraint, instructing the vehicle to visit the destination node within its time window. This means the UAV is allowed to visit the destination node only after the UGV has arrived, as the UAV can only be recharged following the UGV's arrival. However, the UGV can arrive early at refueling stops, and the difference between its arrival and the UAV's landing determines the UGV's recharge waiting time. The constraint in Eq. 17 indicates that there should be no flow once the vehicle reaches the end node and the route will end there. Equation 18 is a binary decision variable in charge of flow between the edges. The continuous decision variable, Equation 19, monitors the fuel level at any node and has zero as the lower bound value. Equation 20 denotes the integer decision variable that computes the cumulative time of the UAV's route and has a lower bound of zero. To solve the E-VRPTW model with its associated constraints, we utilize Google's OR-Tools CP-SAT solver [47], known for its effectiveness in solving the Traveling Salesman Problem (TSP) and Vehicle Routing Problem (VRP) via constraint programming (CP). The OR-Tools solver combines search tree strategies, local search techniques, and metaheuristics to efficiently find good approximate optimal solutions. Our choice of Google's OR-Tools is driven by its rapid solution capabilities, particularly in heuristic implementations for TSP and VRP problems. At the core of OR-Tools lies the CP-SAT solver, which uses a *DecisionBuilder* for managing decision variables. This involves selecting which variable to assign next and determining the value to assign. It applies the *Path Cheapest Arc Strategy* for an initial feasible solution, starting from the "start" node and connecting to the nearest subsequent node until the route is complete. OR-Tools then refines this solution through local search, using a move operator to explore feasible, cost-effective configurations, continuing until no further improvements can be made. We also set an execution time limit of $T_s = 60$ seconds for the solver to generate a solution within a feasible time frame.

#### 3.2.3 UAV Planner

By solving this E-VRPTW for subproblem $SP_i$, ***UAVPlanner*** gets the optimal UAV route $\tau^a$, (both spatial component

**Algorithm 3** UAV Planner

**Input:** Set of subproblems $\mathcal{S} = [SP_i] \leftarrow Task\ Allocation$
**Output:** UAV route $\tau^a = (X^a, T^a) = [(x_i^a, t_i^a)]$

$\tau^a = [\,]$
**for** $SP_i$ in $\mathcal{S}$ **do**
  $t_i^g \leftarrow UGV\ Planner$
  send $t_i^g \rightarrow SP_i$
  $(x_i^a, t_i^a), \mathcal{R}_t \leftarrow EVRPTW$
  $\tau^a$.append$(x_i^a, t_i^a)$
  send $t_i^a, \mathcal{R}_t \rightarrow UGV\ Planner$
**end for**

$X^a$ and temporal component $T^a$); i.e, $\tau^a = (X^a, T^a) = [(x_i^a, t_i^a)]$, as well as the time instance $t_i^a$ at which the UAV will arrive at the refuel stop $m_i^r$ to recharge with the UGV and the recharging time $\mathcal{R}_t$ of it, which is dependent on its fuel consumption level. These information are fed back to the ***UGVPlanner*** again to calculate the UGV *availability* time window for next subproblem $SP_{i+1}$. This reciprocal and iterative process (line 3 - 7 in Algorithm 3) between the ***UAVPlanner*** and ***UGVPlanner*** is what facilitates the cooperative route for the entire task scenario. In Fig. 4a and b, we got the routes for the UAV and the UGV which are combined together to get the complete routes of UAV and UGV for the entire task scenario (Fig. 4c).

The proposed algorithm's overall time complexity is determined by the execution time limit $T_s$ set for solving the E-VRPTW subproblems, rather than by traditional algorithmic complexity measures. The overall time complexity of the proposed algorithm inherently depends on the fixed time limit ($T_s$) set for the solver to address each individual E-VRPTW subproblem. As the problem sizes increase, the number of subproblems ($s$) grows linearly. Consequently, the overall time complexity of the framework can be considered as scaling with $s \times T_s$. This approach ensures that the algorithm operates within practical bounds of time, making it suitable for applications where a predictable execution time is crucial.

## 4 Results

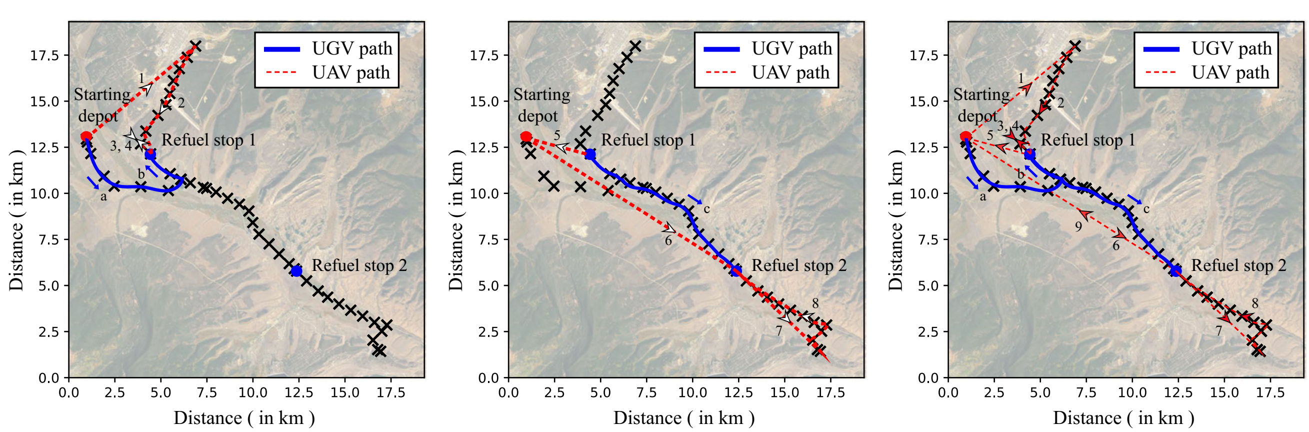
We implement the proposed framework across diverse random task scenarios to evaluate its proficiency. The task scenarios, generated at three distinct scales, help us investigate the impact of UAV fuel capacity on the overall routing process. In these tests, we compare the results of the greedy and constraint programming methods when applied at the outer-loop of our proposed framework. Additionally, to ascertain the upper limit of the performance metrics, we also construct a UGV-only route in each scenario, which facilitates the assessment of the practicality and advantages of the cooperative UAV-UGV route in each specific scenario. In each problem scenario, it is assumed that the UAV and UGV commence their journey from a fixed starting depot with constant speeds of 10 m/s and 4.5 m/s respectively and UAV has a fuel capacity of 287.7 kJ [40].

**Table 1** Specifications of task Scenario

| Scale | Map size | *scale factor* | No. of task points |
|---|---|---|---|
| Small | 16 km x 16 km | 1.5 | 30 |
| Medium | 25 km x 25 km | 3 | 60 |
| Large | 40 km x 40 km | 9 | 100 |

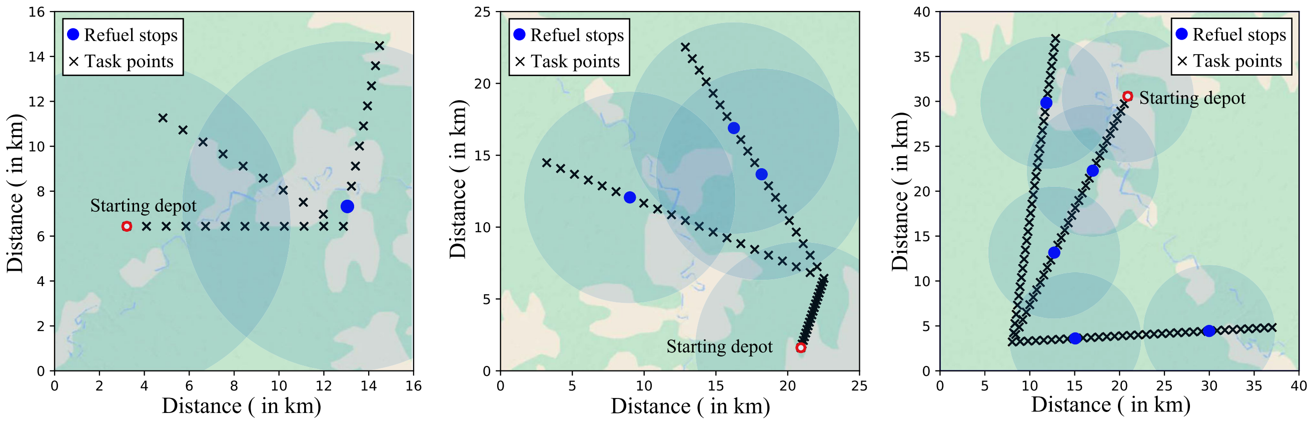
### 4.1 Design of Experiments

The efficacy of our proposed framework is tested across numerous random task scenarios generated at three separate scales. We design the scenarios such that the farthest task point from the starting depot is always outside the UAV's radial coverage, guaranteeing that at least one refueling stop is necessary for the UAV to complete the task. To substantiate the robustness and adaptability of the suggested methodology, we experiment with three distinct scales of task instances, as exhibited in Table 1. We introduce a *scale factor*, a non-dimensional number, to represent the relationship between the scenario map size and the UAV's radial fuel coverage area. Three examples of scenarios from three different




**(a)** solution of subproblem 1 **(b)** solution of subproblem 2 **(c)** entire solution

**Fig. 4** a) UAV-UGV routes from subproblem 1 b) UAV-UGV routes from subproblem 2 c) UAV-UGV routes for entire task scenario after combining subproblem 1 & 2. The animation can be found at http://tiny.cc/3wgnxz




(a) small scale scenario (b) medium scale scenario (c) large scale scenario

**Fig. 5** Sample scenarios with starting depot, task points, and the obtained refuel stops from minimum set cover algorithm. Radial circles represent the UAV's coverage area from the refuel stops. As the task scenario's scale increases, both the number of task points and the number of refueling stops grow proportionally

**Table 2** Time metrics of different scenarios

| Map Size | Scenarios | Route Time (min.) | | | Improvement (%) | |
|---|---|---|---|---|---|---|
| | | Cooperative Routing | | UGV only | | |
| | | Greedy Method | CP Method | | Greedy Method | CP Method |
| Small scale | Scenario 1 | 200 | 210 | 249 | 19.68 | 15.66 |
| | Scenario 2 | 91 | 82 | 133 | 31.58 | 38.35 |
| | Scenario 3 | 117 | 115 | 194 | 39.69 | 40.72 |
| | Scenario 4 | 153 | 153 | 178 | 14.04 | 14.04 |
| | Scenario 5 | 148 | 148 | 159 | 6.92 | 6.92 |
| | Scenario 6 | 222 | 222 | 289 | 23.18 | 23.18 |
| | Scenario 7 | 190 | 149 | 196 | 3.06 | 23.98 |
| | Scenario 8 | 128 | 128 | 198 | 35.35 | 35.35 |
| | Scenario 9 | 222 | 210 | 303 | 26.73 | 30.69 |
| | Scenario 10 | 223 | 128 | 214 | -4.21 | 40.19 |
| Medium Scale | Scenario 1 | 411 | 404 | 497 | 17.30 | 18.71 |
| | Scenario 2 | 414 | 409 | 622 | 33.44 | 34.24 |
| | Scenario 3 | 364 | 364 | 511 | 28.77 | 28.77 |
| | Scenario 4 | 340 | 338 | 452 | 24.78 | 25.22 |
| | Scenario 5 | 297 | 297 | 341 | 12.90 | 12.90 |
| | Scenario 6 | 396 | 417 | 524 | 24.43 | 20.42 |
| | Scenario 7 | 296 | 298 | 370 | 20.00 | 19.46 |
| | Scenario 8 | 324 | 325 | 477 | 32.08 | 31.87 |
| | Scenario 9 | 292 | 295 | 403 | 27.54 | 26.80 |
| | Scenario 10 | 291 | 257 | 459 | 36.60 | 44.01 |
| Large Scale | Scenario 1 | 461 | 407 | 438 | –5.25 | 7.08 |
| | Scenario 2 | 440 | 431 | 467 | 5.78 | 7.71 |
| | Scenario 3 | 537 | 532 | 484 | –10.95 | –9.92 |
| | Scenario 4 | 611 | 519 | 462 | –32.25 | –12.34 |
| | Scenario 5 | 744 | 744 | 680 | –9.41 | –9.41 |

**Table 2** continued

| Map Size | Scenarios | Route Time (min.) | | | Improvement (%) | |
|---|---|---|---|---|---|---|
| | | Cooperative Routing | | UGV only | | |
| | | Greedy Method | CP Method | | Greedy Method | CP Method |
| | Scenario 6 | 452 | 412 | 436 | –3.67 | 5.50 |
| | Scenario 7 | 655 | 701 | 607 | –7.91 | –15.49 |
| | Scenario 8 | 682 | 684 | 588 | –15.99 | –16.33 |
| | Scenario 9 | 620 | 620 | 570 | -8.77 | –8.77 |
| | Scenario 10 | 613 | 604 | 537 | –14.15 | –12.48 |

scales are shown in Fig. 5.

$$scale\ factor = \frac{\text{Area of scenario}}{\text{UAV coverage area on a single charge}} \tag{22}$$

For each instance, two types of cooperative routes (if different) are generated by employing the Greedy method and the CP method at the outer loop of the suggested framework. The UGV-only route (UGV operates alone) is also determined for the specific scenarios. There is no benchmark solution exists to this specific problem due to its complex combinatorial nature. Hence, we treat the UGV-only route as the baseline method for comparison. Comparison is made between the cooperative route and UGV only route, which signifies the impact of cooperation between UAV and UGV on the task execution. The total task completion time and total energy consumption are treated as the metrics for the evaluation of routes.

## 4.2 Time Metrics

In Table 2, the total task completion time of the route obtained by the three aforementioned methods has been displayed. For all instances in the small-scale scenarios, cooperative routing with the constraint programming method in the framework's outer loop is proved more time-efficient than the UGV-only routing. The task completion time for UGV-only routes is reduced by approximately 6% to 40% through the cooperation between the UAV and UGV in small-scale scenarios. Although cooperative routing with the Greedy method in the outer loop baseline doesn't perform as well as the CP method, it is more time-efficient than the UGV-only route in most instances. As the only exception, for scenario 10 on a small scale, the Greedy method can't improve the task completion time through the operative route.

For medium-scale scenarios, the task completion time is improved by 12% up to 45% through the CP method-based cooperative routing, while for the Greedy method-based cooperative route, the improvement range is 12% up to 30%. The cooperative route is more economical than the UGV individual route for most scenarios with the CP method at the outer loop. However, the improvement range is less than that in the small-scale scenarios.

However, for most large-scale scenarios, the cooperative route can't improve the total task completion time, making the UGV-only route the optimal choice. Figure 6 depicts the improvement in the total task completion time achieved by the cooperative route with the CP method and the Greedy method at the outer loop of our framework over the UGV-only route for three types of scenarios.

## 4.3 Energy Metrics

The total energy consumed by the UAV and UGV during the routing process is also analyzed across the same scenarios, and depicted in Table 3. The improvement percentage reflects the relative gain in total energy consumption that is achieved through UAV-UGV cooperation. The results confirmed that cooperative routing is more energy-efficient than UGV-only routing. Among the cooperative routing methods, the CP method applied in the outer loop outperformed the Greedy method.

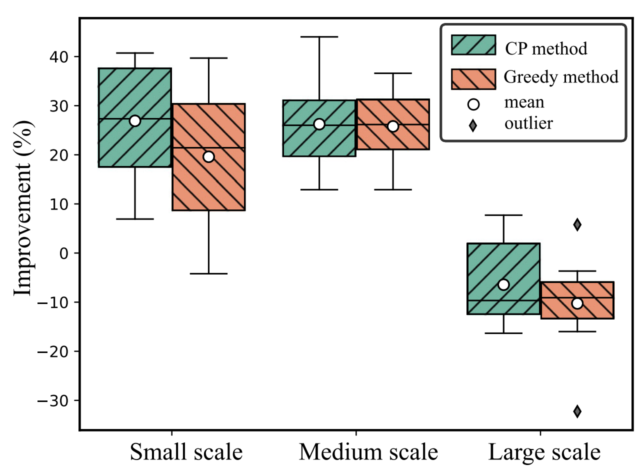



**Fig. 6** Time metrics across three different scale of scenarios

**Table 3** Energy metrics of different scenarios

| Map Size | Scenarios | Total energy consumption (MJ) | | | Improvement (%) | |
|---|---|---|---|---|---|---|
| | | Cooperative Routing | | UGV only | | |
| | | Greedy Method | CP Method | | Greedy Method | CP Method |
| Small scale | Scenario 1 | 20.56 | 20.61 | 37.07 | 44.54 | 44.40 |
| | Scenario 2 | 6.73 | 7.19 | 19.80 | 66.00 | 63.68 |
| | Scenario 3 | 12.53 | 11.82 | 28.88 | 56.62 | 59.08 |
| | Scenario 4 | 17.08 | 17.08 | 26.50 | 35.56 | 35.56 |
| | Scenario 5 | 12.46 | 12.46 | 23.67 | 47.38 | 47.38 |
| | Scenario 6 | 23.47 | 23.47 | 43.03 | 45.44 | 45.44 |
| | Scenario 7 | 14.14 | 16.72 | 29.18 | 51.56 | 42.71 |
| | Scenario 8 | 13.49 | 13.49 | 29.48 | 54.25 | 54.25 |
| | Scenario 9 | 25.47 | 24.86 | 45.11 | 43.55 | 44.89 |
| | Scenario 10 | 19.73 | 13.60 | 31.86 | 38.07 | 57.32 |
| Medium Scale | Scenario 1 | 39.40 | 34.00 | 73.99 | 46.75 | 54.04 |
| | Scenario 2 | 41.90 | 41.63 | 92.60 | 54.75 | 55.04 |
| | Scenario 3 | 41.82 | 41.82 | 76.08 | 45.03 | 45.03 |
| | Scenario 4 | 38.15 | 38.05 | 67.29 | 43.31 | 43.46 |
| | Scenario 5 | 30.10 | 30.10 | 50.77 | 40.71 | 40.72 |
| | Scenario 6 | 46.52 | 46.85 | 78.01 | 40.37 | 39.94 |
| | Scenario 7 | 31.56 | 31.57 | 55.09 | 42.71 | 42.70 |
| | Scenario 8 | 39.75 | 39.67 | 71.02 | 44.03 | 44.15 |
| | Scenario 9 | 33.01 | 33.14 | 60.00 | 44.98 | 44.76 |
| | Scenario 10 | 31.49 | 30.71 | 68.34 | 53.92 | 55.06 |
| Large Scale | Scenario 1 | 43.63 | 41.27 | 65.21 | 33.09 | 36.71 |
| | Scenario 2 | 46.17 | 45.35 | 69.53 | 33.59 | 34.78 |
| | Scenario 3 | 62.94 | 62.67 | 72.06 | 12.66 | 13.03 |
| | Scenario 4 | 56.05 | 59.50 | 68.78 | 18.51 | 13.50 |
| | Scenario 5 | 92.74 | 92.74 | 101.24 | 8.39 | 8.39 |
| | Scenario 6 | 49.15 | 42.32 | 64.91 | 24.28 | 34.81 |
| | Scenario 7 | 81.63 | 64.41 | 90.37 | 9.68 | 28.73 |
| | Scenario 8 | 80.17 | 79.96 | 87.54 | 8.42 | 8.66 |
| | Scenario 9 | 75.80 | 75.80 | 84.86 | 10.67 | 10.67 |
| | Scenario 10 | 72.37 | 71.89 | 79.95 | 9.48 | 10.07 |

For the small-scale instances, cooperative routing enables energy savings ranging from 28% up to 58%. For medium-scale scenarios, the improvement ranges from 40-55%, while for large-scale scenarios, the range is between 8-37%. This data affirms that cooperative routing, particularly when employing the CP method in the outer loop, can significantly enhance energy efficiency across a variety of scenarios (see Fig. 7).

## 4.4 Computational Time

For real-time applications, the computational time of the vehicle routing problem is a crucial factor. The greedy method and the CP method, when implemented at the outer loop of the proposed framework, display notable differences in computational time. As shown in Fig. 8, the greedy method requires substantially less computational time compared to the CP method. All our computations are performed in a Python 3.9 environment running on a 3.7 GHz Intel Core i9 processor with 48 GB RAM on a 64-bit Windows 10 operating system.

Given the subproblem division approach at the inner loop, the computational time increases for both the Greedy and CP methods as the scale of the scenarios increases. However, the Greedy method consistently outperforms the CP method, and the gap between their respective computational times grows

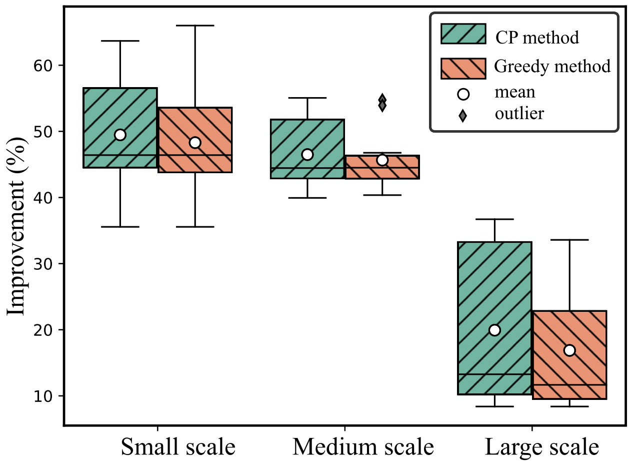



**Fig. 7** Energy metrics across three different scale of scenarios

in proportion to the scale of the scenario. This highlights the Greedy method's efficiency and its suitability for larger-scale scenarios that require rapid computations.

### 4.5 Case Study

As an illustrative example, we examine a real-world task site consisting of 44 task points spread across a 20 km × 20 km area, as presented in Fig. 3a. The UAV and UGV both start their journey from the starting depot and meet at refuel stops for recharging while visiting task points, determined by the minimum set cover algorithm in the UGV routing of the proposed framework. For comparison evaluation, we apply both the CP method and the Greedy approach to solve the minimum set cover algorithm. The cooperative routes obtained from both methods, as illustrated in Fig. 9, are compared. We evaluate the total mission completion time and total energy consumption from the two cooperative routes, as shown in Table 4. Table 4 also highlights the improvements achieved

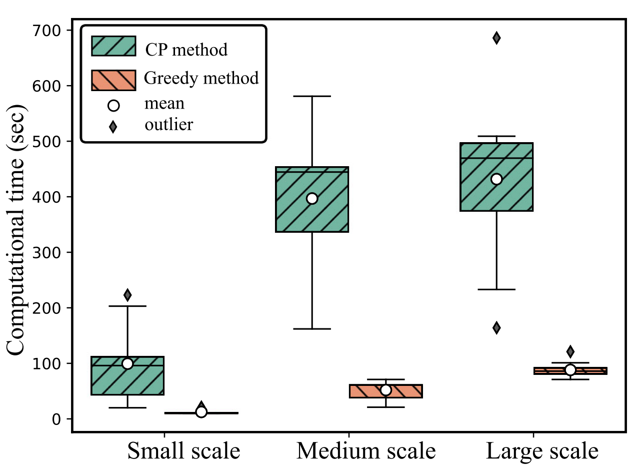



**Fig. 8** Computational time across three different scale of scenarios

by UAV-UGV cooperation over the UGV-only route, which serves as an upper limit. Cooperative routing proves to be more energy-efficient than the UGV-only route. Both the CP method and the Greedy method at the outer loop show positive improvements, reducing total energy consumption in the mission by 36-39%. However, for total mission time, the Greedy method at the outer loop has a negative impact. This is due to the positioning of refuel stops (see trajectory in Fig. 9b), which causes the UAV to take frequent detours (6 times) for recharging at the refuel sites, thereby elongating the total mission time. Conversely, appropriate refuel stop locations determined by the CP method (see trajectory in Fig. 9a) enables the UAV to complete its route with fewer recharging detours (4 times), effectively reducing the total mission time.

## 5 Discussion

The scenario size and the geometric configuration of task points within a scenario critically influence the potential for improving total task completion time through cooperative routing. As observed in Fig. 6, there are positive improvement percentages for small and medium scale scenarios, but a negative trend emerges for large scale scenarios. The total time taken to complete the task via a cooperative route depends on three elements: UAV traversal time, UGV traversal time, and the waiting time of the UGV for UAV refueling. Conversely, the total task completion time for the UGV-alone route depends solely on the UGV traversal time, as it does not involve refueling.

When the UGV operates alone, it has to visit all task target points alone, leading to a significant UGV traversal time. This duration increases proportionally with the scale of the scenario, as demonstrated in Fig. 10. This UGV traversal time can be reduced through cooperative routing, which divides the task points between the UAV and UGV. However, a drawback of cooperative routing is the addition of waiting time during which the UAV is refueled by the UGV.

In small and medium-scale scenarios, the number of task points is lower within smaller areas, resulting in fewer refueling stops. This configuration ensures that the extra waiting time at refueling stops never exceeds the reduction in UGV traversal time, leading to a shorter overall task completion time for the cooperative route compared to the UGV-alone route. However, in large-scale scenarios, where there are more refueling stops to cover a greater number of task points spread over a large area, the additional waiting time can surpass the decrease in UGV traversal time. This results in the cooperative route having a longer overall mission time than the UGV-alone route. Within the cooperative routing methods, the task point density significantly influences the performance of the CP method and the Greedy approach.

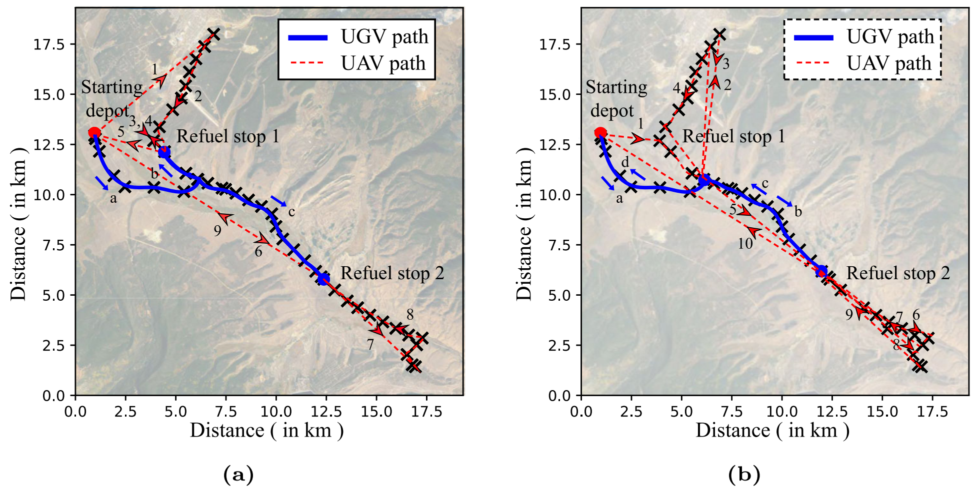



**Fig. 9** UAV-UGV trajectory obtained from bilevel optimization with Greedy and CP method at the outer loop. Numerical and alphabetical order shows the UAV and UGV motion respectively. a) CP method based trajectory b) greedy method based trajectory. The animation of the route can be found at http://tiny.cc/3wgnxz

If the task points are densely spread, having fewer refuel stops leads to fewer recharging detours for the UAV and reduced recharging waiting time. This configuration leads to a decreased overall task completion time for the CP method compared to the Greedy method, as the former is more effective in minimizing the number of refuel stops. Conversely, if task points are not densely clustered, fewer refueling stops may negatively impact the mission time, as the UAV might need to undertake more detours, thereby extending the overall mission duration. In such scenarios, the Greedy method tends to outperform the CP method.

In terms of energy consumption metrics, the cooperative route consistently outperforms the UGV-alone route, regardless of the scenario scale. This is because, as demonstrated in Fig. 1, the UAV consumes five times less energy than the UGV per unit distance traveled, making the UGV the dominant influence on total energy consumption. Hence, the UGV-alone route, which involves a longer UGV traversal distance, consumes more energy than the cooperative route, where the UGV covers a smaller distance due to task division with the UAV.

Nevertheless, as the scale of scenarios increases, the gap in total energy consumption between the UGV-alone route and the cooperative route narrows. This is because in larger scenarios, despite cooperation with the UAV, the UGV must still cover a considerable distance to provide suitable refueling stops, leading to higher overall energy consumption.

The proposed framework can be implemented using a receding horizon approach to account for dynamic environmental changes, such as the random emergence of new task points, disappearance of old task points, and obstacle avoidance. Given that we assume UAVs and UGVs can communicate only during the recharging process, we propose updating the scenario map at every recharging instance to incorporate these changes. Consequently, the ***UGVPlanner*** can resolve the MSC problem to update the UGV route as necessary. Based on the revised UGV route, the UAV routes can be constructed by applying task allocation and the E-VRPTW modeling within the ***UAVPlanner***, as described in Section 3.2. Dynamic planning proves extremely useful for continuous monitoring, disaster management, and traffic monitoring by capturing the most recent state of the environment. We recognize the potential of dynamic planning and suggest it as an avenue for future research in our study.

## 6 Conclusion

In this work, we focus on a cooperative vehicle routing problem involving a multirotor UAV and a wheeled UGV team with fuel and speed constraints. Both vehicles are required to cover a set of assigned task points, with the UAV periodically recharging from the UGV to complete the task in the minimum possible time. Finding the optimal recharging

**Table 4** Impact of the optimal solution of the cooperative routing on case study scenario

| Metrics | Cooperative route | | UGV only route | Improvement (%) | |
|---|---|---|---|---|---|
| | CP method | Greedy method | | CP method | Greedy method |
| Time consumption (min.) | 200 | 272 | 233 | 14.16 | -16.74 |
| Energy consumption (MJ) | 21.98 | 21.14 | 34.69 | 36.62 | 39.06 |

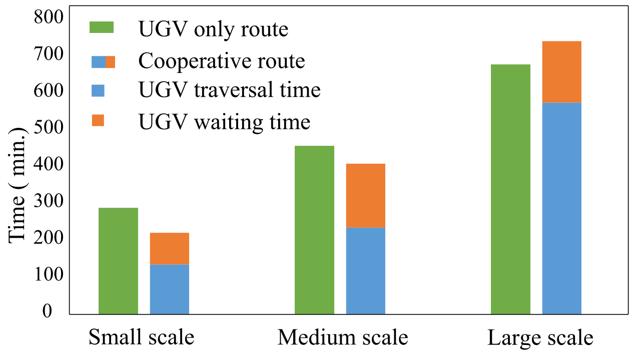



**Fig. 10** Task completion time in cooperative route vs UGV-only route

rendezvous points, in terms of both location and timing, between the UAV and UGV is important to achieve an optimal route in this cooperative routing problem. We introduce a sequential optimization framework that operates in two primary steps. The initial step involves the utilization of a minimum set cover algorithm to determine the locations for refueling stops. These identified locations serve as an input to the ***UGVPlanner***, which then creates the UGV route employing a Traveling Salesman Problem model. In the subsequent step, a task allocation technique is employed to partition the entire problem into smaller, more manageable subproblems. The ***UAVPlanner*** then develops the UAV route by framing these subproblems as instances of the Energy-Constrained Vehicle Routing Problem with Time Windows constraints (E-VRPTW).

Our framework has been successfully applied to 30 distinct task scenarios across three different scales, showcasing its effectiveness and practicality. The cooperative routes resulting from our framework have been benchmarked against the UGV-only routes for the same scenarios that serve as an upper limit for comparison. The results reveal substantial improvements, with time consumption reduced by 10-30% and energy consumption diminished by 15-50% in most instances through the cooperative routing. In the future, we will expand the framework for persistent surveillance of the task points and consider stochasticity in the scenarios. Insights from this study suggest a potential enhancement of leveraging the UGV's idle waiting times during refueling to establish a mobile recharging rendezvous, which will be a focal point in our subsequent investigations. Additionally, we intend to refine our approach by incorporating reinforcement learning techniques to tackle this UAV-UGV cooperative routing problem, thereby providing an additional benchmark for future comparative analyses.

**Acknowledgements** The authors would like to express their gratitude to the DEVCOM Army Research Laboratory for their financial support in funding the projects under grant W911NF-14-S-003.

**Author Contributions** The conceptualization of ideas and methodologies was carried out by MSM, SR, and PAB. Implementation was undertaken by MSM and SR. The initial draft was penned by MSM, while the review and editing process involved JH, JPR, JD, CM, and PAB. The project was overseen and supervised by JH, JPR, JD, CM, and PAB, with PAB also handling project administration. All contributing authors have reviewed and given their consent for the final version of the manuscript to be published.

**Funding** The work is funded by Army Research Laboratory grant W911NF-14-S-003.

**Data Availability** The code and data backing this study's outcomes aren't publicly shared, however it can be obtained upon a credible request.

## Declarations

**Conflict of Interests** The authors declare no conflict of interest.

**Ethics Approval** Not applicable.

**Consent to Participate** Not applicable.

**Consent for Publication** All authors consent to publication.



## References

1. Liu, Y., Luo, Z., Liu, Z., Shi, J., Cheng, G.: Cooperative routing problem for ground vehicle and unmanned aerial vehicle: The application on intelligence, surveillance, and reconnaissance missions. IEEE Access. **7**, 63504–63518 (2019)
2. Stolfi, D.H., Brust, M.R., Danoy, G., Bouvry, P.: UAV-UGV-UMV multi-swarms for cooperative surveillance. Frontiers in Robotics and AI. **8**, 616950 (2021)
3. Puri, A., Valavanis, K., Kontitsis, M.: Statistical profile generation for traffic monitoring using real-time UAV based video data. In: 2007 Mediterranean Conference on Control & Automation. IEEE; p. 1–6 (2007)
4. Chua, C.N.: Integration of multiple UAVs for collaborative ISR missions in an urban environment. California. Naval Postgraduate School, Monterey (2012)
5. Wu, Y., Wu, S., Hu, X.: Cooperative path planning of UAVs & UGVs for a persistent surveillance task in urban environments. IEEE Internet Things J. **8**(6), 4906–4919 (2020)

6. Li, J., Deng, G., Luo, C., Lin, Q., Yan, Q., Ming, Z.: A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems. IEEE Trans. Veh. Technol. **65**(12), 9585–9596 (2016)
7. Tokekar, P., Vander Hook, J., Mulla, D., Isler, V.: Sensor planning for a symbiotic UAV and UGV system for precision agriculture. IEEE Trans. Rob. **32**(6), 1498–1511 (2016)
8. Wang, Z., Sheu, J.B.: Vehicle routing problem with drones. Transportation research part B: methodological. **122**, 350–364 (2019)
9. Tang, Z., Hoeve, W.J.v., Shaw, P.: A study on the traveling salesman problem with a drone. In: Integration of Constraint Programming, Artificial Intelligence, and Operations Research: 16th International Conference, CPAIOR 2019, Thessaloniki, Greece, June 4–7, 2019, Proceedings 16. Springer; 2019. p. 557–564
10. Murray, C.C., Chu, A.G.: The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. Transportation Research Part C: Emerging Technologies. **54**, 86–109 (2015)
11. Chen, Y., Chen, M., Chen, Z., Cheng, L., Yang, Y., Li, H.: Delivery path planning of heterogeneous robot system under road network constraints. Computers & Electrical Engineering. **92**, 107197 (2021)
12. Bouman, P., Agatz, N., Schmidt, M.: Dynamic programming approaches for the traveling salesman problem with drone. Networks **72**(4), 528–542 (2018)
13. Ha, Q.M., Deville, Y., Pham, Q.D., Hà, M.H.: On the min-cost traveling salesman problem with drone. Transportation Research Part C: Emerging Technologies. **86**, 597–621 (2018)
14. Levy, D., Sundar, K., Rathinam, S.: Heuristics for routing heterogeneous unmanned vehicles with fuel constraints. Math. Probl. Eng. **2014**, 131450 (2014)
15. Sundar, K., Venkatachalam, S., Formulations, Rathinam S., algorithms for the multiple depot, fuel-constrained, multiple vehicle routing problem. In,: American Control Conference (ACC). IEEE **2016**, 6489–6494 (2016)
16. Maini, P., Sujit, P.: 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. p. 1370–1377
17. Manyam, S.G., Sundar, K., Casbeer, D.W.: Cooperative routing for an air-ground vehicle team–exact algorithm, transformation method, and heuristics. IEEE Trans. Autom. Sci. Eng. **17**(1), 537–547 (2019)
18. Li, H., Chen, J., Wang, F., Bai, M.: Ground-vehicle and unmanned-aerial-vehicle routing problems from two-echelon scheme perspective: A review. Eur. J. Oper. Res. **294**(3), 1078–1095 (2021)
19. Luo, Z., Liu, Z., Shi, J.: A two-echelon cooperated routing problem for a ground vehicle and its carried unmanned aerial vehicle. Sensors. **17**(5), 1144 (2017)
20. Liu, Y., Liu, Z., Shi, J., Wu, G., Pedrycz, W.: Two-echelon routing problem for parcel delivery by cooperated truck and drone. IEEE Transactions on Systems, Man, and Cybernetics: Systems. **51**(12), 7450–7465 (2020)
21. Anderson, R.B.: Routing and Control of Unmanned Aerial Vehicles for Performing Contact-Based Tasks. Virginia Tech (2021)
22. Ropero, F., Muñoz, P., R-Moreno, M.D. TERRA: A path planning algorithm for cooperative UGV–UAV exploration. Engineering Applications of Artificial Intelligence. **78**, 260–272 (2019)
23. Seyedi, S., Yazicioğlu, Y., Aksaray, D.: Persistent surveillance with energy-constrained uavs and mobile charging stations. IFAC-PapersOnLine. **52**(20), 193–198 (2019)
24. Lin, X., Yazıcıoğlu, Y., Aksaray, D.: Robust planning for persistent surveillance with energy-constrained UAVs and mobile charging stations. IEEE Robotics and Automation Letters. **7**(2), 4157–4164 (2022)
25. Nigam, N., Bieniawski, S., Kroo, I., Vian, J.: Control of multiple UAVs for persistent surveillance: Algorithm and flight test results. IEEE Trans. Control Syst. Technol. **20**(5), 1236–1251 (2011)
26. Nigam, N.: The multiple unmanned air vehicle persistent surveillance problem: A review. Machines. **2**(1), 13–72 (2014)
27. Frew, E., Xiao, X., Spry, S., McGee, T., Kim, Z., Tisdale, J., et al.: Flight demonstrations of self-directed collaborative navigation of small unmanned aircraft. In: AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit; 2004. p. 6608
28. Jodeh, N., Mears, M., Gross, D.: An overview of the cooperative operations in urban terrain (counter) program. In: AIAA Guidance, Navigation and Control Conference and Exhibit; 2008. p. 6308
29. Redding, J., Toksoz, T., Ure, N.K., Geramifard, A., How, J.P., Vavrina, M.A., et al.: Distributed multi-agent persistent surveillance and tracking with health management. In: Proceedings of the AIAA Guidance Navigation and Control Conference, Portland, OR, USA; 2011. p. 8–11
30. Saad, E., Vian, J., Clark, G., Bieniawski, S.: Vehicle swarm rapid prototyping testbed. In: AIAA Infotech@ Aerospace Conference and AIAA Unmanned... Unlimited Conference; 2009. p. 1824
31. How, J.P.: Multi-vehicle flight experiments: Recent results and future directions. In: Proceedings of AVT-146 Symposium on Platform Innovations and System Integration for Unmanned Air, Land and Sea Vehicles (2007)
32. Wu, N., Chacon, C., Hakl, Z., Petty, K., Smith, D.: Design and implementation of an unmanned aerial and ground vehicle recharging system. In: IEEE National Aerospace and Electronics Conference (NAECON). IEEE **2019**, 163–168 (2019)
33. Yu, K., Budhiraja, A.K., Buebel, S., Tokekar, P.: Algorithms and experiments on routing of unmanned aerial vehicles with mobile recharging stations. Journal of Field Robotics. **36**(3), 602–616 (2019)
34. Karapetyan N, Asghar AB, Bhaskar A, Shi G, Manocha D, Tokekar P.:Ag-cvg: Coverage planning with a mobile recharging ugv and an energy constrained uav. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2024. p. 2617–2623
35. Ramasamy, S., Reddinger, J.P.F., Dotterweich, J.M., Childers, M.A., Bhounsule, P.A.: Cooperative route planning of multiple fuel-constrained Unmanned Aerial Vehicles with recharging on an Unmanned Ground Vehicle. In: 2021 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE; 2021. p. 155–164
36. Ramasamy, S., Reddinger, J.P.F., Dotterweich, J.M., Childers, M.A., Bhounsule, P.A.: Coordinated Route Planning of Multiple Fuel-constrained Unmanned Aerial Systems with Recharging on an Unmanned Ground Vehicle for Mission Coverage. Journal of Intelligent & Robotic Systems. **106**(1), 1–18 (2022)
37. Mondal, M.S., Ramasamy, S., Humann, J.D., Reddinger, J.P.F., Dotterweich, J.M., Childers, M.A.,: Optimizing Fuel-Constrained UAV-UGV Routes for Large Scale Coverage: Bilevel Planning in Heterogeneous Multi-Agent Systems. In, et al.: International Symposium on Multi-Robot and Multi-Agent Systems (MRS). IEEE **2023**, 114–120 (2023)
38. Ramasamy, S., Mondal, M.S., Reddinger, J.P.F., Dotterweich, J.M., Humann, J.D., Childers, M.A., et al.: Heterogenous vehicle routing: comparing parameter tuning using genetic algorithm and bayesian optimization. In: 2022 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE; 2022. p. 104–113
39. Ramasamy, S., Mondal, M.S., Reddinger, J.P.F., Dotterweich, J.M., Humann, J.D., Childers, M.A., et al.: Solving Vehicle Routing Problem for Unmanned Heterogeneous Vehicle Systems using Asynchronous Multi-Agent Architecture (A-teams). In: 2023 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE; 2023. p. 95–102
40. Chour, K., Reddinger, J.P., Dotterweich, J., Childers, M., Humann, J., Rathinam, S., et al.: A reactive energy-aware rendezvous planning approach for multi-vehicle teams. In: 2022 IEEE 18th

International Conference on Automation Science and Engineering (CASE). IEEE; 2022. p. 537–542
41. Hurwitz, A.M., Dotterweich, J.M., Rocks, T.A.: Mobile robot battery life estimation: battery energy use of an unmanned ground vehicle. In: Energy Harvesting and Storage: Materials, Devices, and Applications XI. vol. 11722. SPIE; 2021. p. 24–40
42. Gao, W., Luo, J., Zhang, W., Yuan, W., Liao, Z.: Commanding cooperative ugv-uav with nested vehicle routing for emergency resource delivery. IEEE Access. **8**, 215691–215704 (2020)
43. Maini, P., Sundar, K., Singh, M., Rathinam, S., Sujit, P.: Cooperative aerial-ground vehicle route planning with fuel constraints for coverage applications. IEEE Trans. Aerosp. Electron. Syst. **55**(6), 3016–3028 (2019)
44. Gens, G., Levner, E.: Complexity of approximation algorithms for combinatorial problems: a survey. ACM SIGACT News **12**(3), 52–65 (1980)
45. Lin, B.: A simple gap-producing reduction for the parameterized set cover problem. arXiv:1902.03702. (2019)
46. Slavík, P.: A tight analysis of the greedy algorithm for set cover. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing; 1996. p. 435–441
47. Google.: Google OR-tools. Online; accessed Feb 2, 2021. https://developers.google.com/optimization
48. Miller, C.E., Tucker, A.W., Zemlin, R.A.: Integer programming formulation of traveling salesman problems. Journal of the ACM (JACM). **7**(4), 326–329 (1960)

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

**Md Safwan Mondal** is a PhD student in the Department of Mechanical and Industrial Engineering at the University of Illinois Chicago. He received his BE from Jadavpur University, Kolkata. Prior to his PhD, he interned at the Indian Institute of Technology Mandi, India. His research focuses on robotics, multi-robot systems, optimization, and artificial intelligence. He has published in leading conferences such as IROS, CASE, MRS, and ICUAS, as well as in journals like JIRS, with an emphasis on multi-robot planning and optimization.

**Subramanian Ramasamy** is currently pursuing a Ph.D. in the Department of Mechanical and Industrial Engineering at the University of Illinois Chicago, IL, USA. He received his B.Eng. degree from Sri Sivasubramaniya Nadar College of Engineering, Anna University, Chennai, India, in 2019. He was an Operations Research Intern at Amtrak in 2022. His research interests include operations research, deep learning, optimization, and path planning for autonomous vehicles. He has published papers in leading robotics conferences (IROS, CASE, ICUAS) and journals (JIRS) focusing on automation and route planning for unmanned vehicle systems.

**James D. Humann** received the B.S. degree in mechanical engineering from The University of Oklahoma and the M.S. and Ph.D. degrees in mechanical engineering from the University of Southern California. He has been with the DEVCOM Army Research Laboratory since 2017. He is currently a Mechanical Engineer with the Vehicle Applied Research Branch. His research interests include design, path planning, and simulation for multi-robot systems.

**James M. Dotterweich** is a robotics research engineer with the Combat Capabilities Development Command Army Research Laboratory DEVCOM ARL in Aberdeen Maryland, USA. He received his Master of Science in Mechanical Engineering with a robotics focus from The University of Utah, Salt Lake City, UT, USA in 2015. He received his Bachelor's degree in Aerospace Engineering from The University of Colorado at Boulder, CO, USA in 2008. He has been working on robotic systems and software development for unmanned aerial and ground vehicles for ARL for the past eight years.

**Jean-Paul F. Reddinger** is an Aerospace Research Engineer for the Vehicle Applied Research Branch of the DEVCOM Army Research Laboratory. He received his B.S. in 2008 and his Ph.D. in 2012, both from Rensselaer Polytechnic Institute. His primary research focus is at the intersection of vertical take-of and landing (VTOL) aircraft aeromechanics, dynamics, and autonomy.

**Marshal A. Childers** achieved a Masters Degree in Mechanical Engineering from the University of Maryland, Baltimore County (UMBC) in 2001. He attained a Bachelors Degree in Mechanical Engineering from the same university, graduating with honors in 2000. Mr. Childers is a Mechanical Engineer for the Combat Capability Development Command (CCDC)-Army Research Laboratory at Aberdeen Proving Ground (APG), MD. He leads eforts in integration, prototyping, and experimentation of autonomous robotics technologies, and has collaborated on numerous experiments that were conducted to evaluate technologies developed by ARL-funded research programs.

**Pranav A. Bhounsule** is an Assistant Professor in the Department of Mechanical and Industrial Engineering at the University of Illinois Chicago. He received his BE from Goa University in 2004, MTech from the Indian Institute of Technology Madras in 2006, and PhD from Cornell University in 2012. His research focuses on mathematical optimization, estimation, and control of robotic systems, with applications in unmanned systems. He has authored over 50 peer-reviewed publications and received several awards, including Best Paper in Biologically Inspired Robotics (Climbing and Walking Conference, 2012) and Best Paper Award (ASME Computers and Information in Engineering Conference, 2019).