Computationally Efficient Multi-Agent Optimization Framework for Online Routing of UAV-UGV System

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

***Abstract*— Unmanned Aerial Vehicles (UAVs) have the ability to monitor vast areas but are limited in their battery capacity. The collaboration with Unmanned Ground Vehicles (UGVs) can significantly enhance the endurance and potential of UAVs by utilizing them as mobile recharging vehicles. However, such collaboration increases the complexity of planning the routes of the vehicles. For practical applications, it's crucial to efficiently develop high-quality UGV-UAV routing solutions within a reasonable time frame and quickly adapt those routes to changing conditions, if any. In this paper, we propose an improved method for multi-agent optimization framework that provides computationally efficient online UGV-UAV route solutions. The effectiveness of the optimization framework is validated with several scenarios by comparing it across standard meta-heuristic baselines like the Genetic Algorithm in simulation. The proposed framework produces near-optimal solutions that are computed 40% faster than the baseline while maintaining the solution quality within 2%. Additionally, the framework's computational efficiency is validated using hardware (`http://tiny.cc/ngjkxz`), demonstrating its ability to do online planning when mission points are dynamically added or removed from the scenario.**

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have rapidly evolved across diverse fields such as entertainment, logistics, surveillance, and disaster relief management due to their small size, relatively low cost, agility, and ease of usage [1]. However, the major limitation of using such UAVs is their restricted operational time due to limited battery capacity, which does not allow them to perform critical tasks such as surveillance and disaster relief that either demand large-scale usage or longer duration.

To perform tasks of high endurance over wider areas, UAVs could be paired up with Unmanned Ground Vehicles (UGVs) to provide them with mobile recharging platforms [2]. In this setup, due to their relatively fast speeds and high altitude, UAVs can provide a bird's eye view of the ground and return to UGVs for recharging before taking off again. The collaborative routing of Unmanned Aerial and Unmanned Ground Vehicle Systems is an NP-Hard combinatorial optimization problem [3]. The traditional route planning algorithms often struggle with online planning due to the challenges of combinatorial optimization. Our approach overcomes these issues by developing an optimization framework that can provide online planning for UAV-UGV collaborative routing.

2. RELATED WORK

Several works in the literature have addressed the cooperative UGV-UAV routing problem. Liu et. al. [4] solved a cooperative Ground Vehicle(GV)-UAV routing problem for surveillance and reconnaissance missions. They proposed a method that first created a large UAV route without considering battery limits. Then, they use a Split Heuristic to split this route into feasible parts connected by GVs to complete the route. Gao et. al. [5] formulated a cooperative UGV-UAV routing problem for emergency resource delivery during COVID-19, using a system that accepts operation orders through an intelligent module and plans routes with a Mixed Integer Linear Programming (MILP) method, followed by an iterative improvement algorithm. Seyedi et. al. [6] addressed the problem of persistent surveillance by using energy-constrained UAVs and the UGVs. The UGVs acted as mobile recharging stations to recharge UAVs. The UGVs and UAVs were organized as a set of UGV-UAV teams. The environment was optimally partitioned into several segments, where each team performed persistent surveillance of task points in those respective partitions in a cyclic fashion. The task points of each partition were visited optimally by the UGV-UAV team using the Dantzig-Fulkerson-Johnson (DFJ) linear programming algorithm. These studies are limited in utilizing specific optimization algorithms for UGV and UAV routing, missing out on the potential benefits of combining different algorithms. The combination could enhance solution quality and computational efficiency by leveraging the strengths of each algorithm together. [7].

The proposed research work deals with improving a multi-agent optimization framework called Asynchronous Teams (A-Teams) [8]. A-Teams is a multi-algorithm framework that uses a team of optimization agents such as Constructor, Improver, and Destroyer agents to evolve a solution pool towards near-optimal results. In this context,"agents" refer to the role of algorithms in the framework. Jedrzejowicz et al. [9] implemented this framework to solve a Resource Investment Problem in which the Improver agents use different optimization algorithms like Local search, Lagrangian relaxation, Path relinking algorithms, Crossover operators

[1] Subramanian Ramasamy, Md Safwan Mondal, and Pranav A. Bhounsule are with the Department of Mechanical and Industrial Engineering, University of Illinois Chicago, IL, 60607 USA. `sramas21@uic.edu` `mmonda4@uic.edu` `pranav@uic.edu` [2] James D. Humann is with DEVCOM Army Research Laboratory, Los Angeles, CA, 90094 USA.james.d.humann.civ@army.mil [3] Jean-Paul F. Reddinger, James M. Dotterweich, Marshal A. Childers are with DEVCOM Army Research Laboratory, Aberdeen Proving Grounds, Aberdeen, MD 21005 USA. jean-paul.f.reddinger.civ@army.mil, james.m.dotterweich.civ@army.mil, marshal.a.childers.civ@army.mil † This work was supported by ARO grant W911NF-14-S-003.

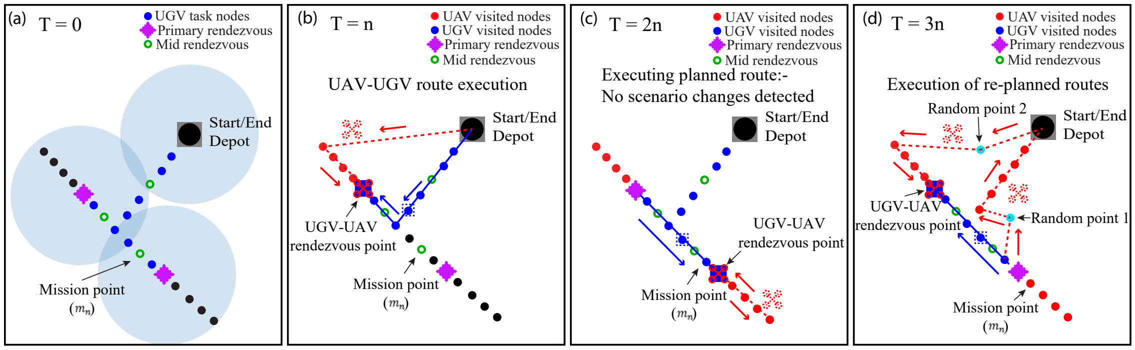

Fig. 1: Persistent surveillance illustration for a collaborative UGV-UAV routing problem. For a single Persistent Surveillance mission, steps a) through d) occur sequentially, where 'n' represents the frame rate. a) A candidate UGV-UAV task visit points. b), c), d) Execution of the planned/re-planned route. Steps (c) and (d) are repeated for persistent node visits. Re-planning is done online if dynamic changes are observed.

and cooperate together to solve such a problem. Other multi-agent optimization frameworks are used across the literature to solve combinatorial optimization problems. For instance, Milano and Roli et al. [10] introduced MultiA-Gent Metaheuristic Architecture (MAGMA), an optimization framework that facilitated cooperation between Memetic algorithms and GRASP (Greedy Randomized Adaptive Search Procedures). In MAGMA, agents were operated at various levels: Solution Builder Agents applied constructive heuristic algorithms, Solution Enhancer Agents carried out local searches, Strategy Agents devised strategies to avoid local optima, and Coordination Agents oversaw the search process and agent coordination.

On the experimental front, limited work has been done to implement the different routing and path planning algorithms on the UAV-UGV hardware system. Nigam et al. [11] solved the persistent surveillance problem of multiple UAVs by developing a optimal control policy structure to navigate on a gridded target space and validated it on their hardware testbed for up to 4 UAVs. The policy dictates the UAVs' action to move front, back, or side based on the age period of each grid cell. Karapetyan et al. [12] demonstrated an approach to solve a coverage path planning problem for a UGV-UAV system using a two-step algorithm. Their two-step algorithm first ignored UAV energy limits to plan the overall coverage path. In the first step, both UGV and UAV optimal coverage trajectories across an area are planned using the Boustrophedon Cellular Decomposition (BCD) algorithm. Then, they divided the area into clusters considering the UAV energy limits, using bipartite graph matching to link UAV and UGV clusters efficiently. They tested their approach with an outdoor UAV-UGV system to prove its effectiveness.

Some works in the literature have also considered re-planning UAV-UGV paths owing to dynamic changes in the environment. Ni et al. [13] introduced a online path planning method for mixed UAV-UGV systems using a Dragonfly algorithm inspired by nature to optimize movements in 3D space, and tested in 3D simulations. Ma et al. [14] developed a fast re-planning method for obtaining optimal UAV paths between the start and goal. The work presented an improved A* algorithm to perform real-time path re-planning. Martinez et. al. [15] tackled a real-time navigation problem for urban firefighting with UAVs and UGVs in GNSS-denied areas. They used Monte-Carlo localization to accurately determine the robots' positions in a pre-mapped environment, a Lazy Theta* algorithm for global path planning, and 3D-LIDAR mapping to enable the local path planner to quickly adjust plans based on new obstacles or changes, ensuring efficient navigation. Both UGV and UAV uses the same set of algorithms, but UGV planning is done in 2D whereas the UAV planning is done in 3D.

In this paper, we propose an enhanced multi-agent optimization framework (A-Teams) with a new **Predictor Agent** for achieving computationally efficient, near-optimal solutions. The Predictor Agent predicts the feasibility of UGV route parameters, reducing unnecessary computations. We validated the framework on actual hardware, showcasing online planning amid dynamic changes. Our novel contributions are: 1) A computationally efficient framework for online heterogeneous UGV-UAV routing. 2) Implementation and testing of autonomous online route re-planning on hardware under dynamic conditions. 3) A faster approach for the Dynamic Vehicle Routing Problem (DVRP) applied to a heterogeneous UGV-UAV system. This approach stands out because it shifts away from the conventional application of DVRP, which primarily concentrates on logistic trucks and vehicles.

3. METHODS

A. Problem formulation

To address the collaborative UGV-UAV routing problem for remote scenarios that are less accessible to humans, we define a set of task points, $\mathcal{M} = \{m_0, m_1, \ldots, m_n\}$, located within a guided road network. The UGV and UAV start from the depot D. These points are considered for persistent surveillance by both UAV and UGV. The UAV can either get recharged on UGV or at D. The vehicle system is of a heterogeneous nature where the UAV travels with higher velocity v_a but is limited in its energy capacity. Thus, it can be paired with a slower-moving UGV with velocity v_g,

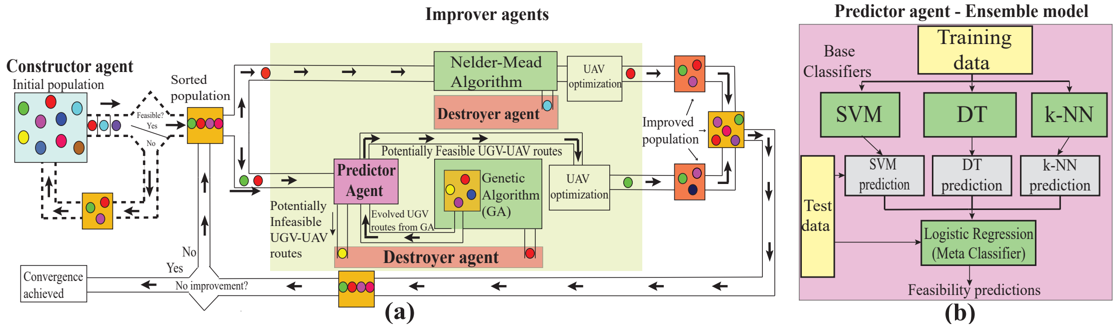

Fig. 2: Description of the proposed A-Teams optimization framework used for UGV-UAV routing. a) This proposed A-Teams has Constructor, Improver, Destroyer and Predictor agents. Agents strategically choose UGV routes, which are fed into UAV optimization to get collaborative UGV-UAV route outputs. b) Training of Ensemble model in Predictor Agent

which has a larger energy capacity E_g to recharge the UAV and power its components. The operational range of the UAV with full energy E_a is visualized as blue circles in Figure 1 a), demonstrating that a single UAV cannot adequately cover all task points, thus substantiating the need for strategic UGV routing to ensure full UAV task coverage and efficient rendezvous. This interdependence between UAV and UGV routes necessitates a bi-level optimization approach, where the UGV route is optimized at the outer level, and the UAV route is optimized at the inner level. This is based on the idea of "UGV first, UAV second" approach.

The optimization of the UGV route is formulated as a set of free parameters, denoting the rendezvous locations that the UAV makes with the UGV. The complete parameter set for a candidate UGV route, X_s, is made up of two subsets: primary rendezvous locations $X_p = \{x_1, x_2, \ldots, x_j\}$ aimed at recharging as well as directing the UAV to new task regions, and mid-rendezvous locations $X_m = \{x_1, x_2, \ldots, x_k\}$ for UAV recharging in middle of the UGV's travel. To identify primary rendezvous points, the Minimum Set Cover (MSC) Problem is solved using Constrained Programming, as detailed by Mondal et al. [16]. This approach tackles the NP-Hard nature of the MSC problem, allowing for multiple X_p sets depending on the scenario's scale and complexity. Equations 3.1-3.3 represents the condition required to obtain a single X_p set such that each location in X_p should have radial coverage R whose union encompasses all the mission points $\mathcal{M}$. R is obtained from half the time of flight T_f^a and velocity v_a of the UAV. Mid-rendezvous points (X_m), used for UAV recharging, are randomly selected between each location in X_p. The total number of route parameters is represented as $\mathcal{S} = |X_s| = j + k$. For each candidate UGV route, the inner-level UAV optimization is solved for near-optimal UAV routes. The UAV optimization is formulated as an Energy-constrained Vehicle Routing Problem (E-VRP) subjected to satisfying UGV route and UAV fuel constraints. The UAV route is given as feedback to the outer level for UGV optimization. The feedback indicates whether the UAV provides feasible or infeasible results for a UGV route. The feasibility criteria consider that the UAV should visit all its task points at least once.

$$min \quad j \tag{3.1}$$

s.t.,

$$\bigcup_{i=1}^{j} \{x_i \mid |x_i - m_n| \le R,\, x_i \in X_p\}, \forall m_n \in \mathcal{M} \backslash x_i \tag{3.2}$$

where

$$j = |X_p|, k = |X_m| = j + 1, R = 0.5 \cdot T_f^a \cdot v_a \tag{3.3}$$

The overall objective of this problem is minimizing the total time T to perform persistent surveillance of UGV-UAV task points subjected to dynamic changes until the UGV runs out of fuel. For instance, Figure 1 a) through d) shows the UGV-UAV persistent surveillance nature for a toy scenario. For Figure a), the initial route planning would happen by solving the optimization problem, and the planned route gets executed as shown in Figure b). When the UGV and UAV are at an X_p location, the scenario is assessed for any dynamically changed or newly appeared task points, represented as $\mathcal{R} = \{r_0, r_1, \ldots, r_n\}$. If there's no dynamic change, the route gets executed per the initial plan as illustrated in Figure c). If such changes are identified, the optimizer performs re-planning, and the UGV and UAV then execute the updated route sequence as illustrated in Figure d). Irrespective of those changes, Figures c) and d) happen in a loop until the UGV runs out of fuel. Due to the specificity of the problem nature and the scenarios considered, the persistent surveillance is ensured by adding a penalty P_t of at least one task point in $\mathcal{M}$ to be visited more than once to the objective function. If all points are visited only once or some are not visited, a large penalty $P_t = L$ is added to the objective. This discourages infeasible or one-time surveillance solutions and encourages solutions with persistent surveillance of the task points. The objective function for the UGV-UAV routing is denoted mathematically as follows. Here $|v(m_n)|$ denotes the number of visits of a point m_n where, $m_n \in \mathcal{M}$. Once the near-optimal UGV-UAV routes are obtained, the set $\mathcal{M}$ gets divided into $\mathcal{M}_g$ for UGV visits, and $\mathcal{M}_a$ for UAV visits, such that $\mathcal{M}_g \cup \mathcal{M}_a = \mathcal{M}$, thus allocating distinct tasks between UGV and UAV.

$$min \quad T + P_t \tag{3.4}$$

where,

$$P = \begin{cases} 0, & \text{if } |v(m_n)| > 1 \text{ for atleast one } m_n, \\ L, & \text{otherwise, where } L \text{ is a large number} \end{cases} \tag{3.5}$$

The different optimization levels for UGV and UAV routes are described in the following subsections.

B. Proposed optimization framework

*1) **Solving outer-level UGV routing using Asynchronous Teams (A-Teams) framework**:* Figure 2 a) represents the proposed framework to perform UGV-UAV optimization. A-Teams is a multi-agent framework that uses a team of algorithms to optimize a given problem. The agents in the framework possess distinct functionalities and solve a problem through cooperation and achieve better solutions than their individual counterparts. There are four agents: **Constructor Agent** is used to develop an initial pool of candidate UGV routes through randomization. **Improver Agent** is used to improve the pool of UGV routes obtained from constructor agent using different optimization methods. **Destroyer Agent** is used to discard non-optimal or infeasible or redundant UGV routes once it gets feedback from UAV optimization. **Predictor Agent** predicts the infeasible UGV routes before being evaluated by UAV optimization using Machine Learning algorithms. **Populations** are shared repositories for storing computed solutions and assessed by different agents. More details about the existing A-Teams framework can be found in [17].

To solve the collaborative UGV-UAV routing problem, the Constructor Agent initializes candidate UGV routes using Latin Hypercube Sampling (LHS) with a sample size of $\mathcal{N}=$ 40. These routes are sent to the inner-level block for UAV optimization using OR-Tools. Feedback on route feasibility is used by Improver Agents, employing Nelder-Mead and Genetic Algorithms, to refine the UGV routes until near-optimal solutions are achieved. The novelty involves the inclusion of an additional Predictor agent in the framework. The primary role of the Predictor agent is to forecast the feasibility of combined UGV-UAV routes, reducing computational effort and improving efficiency. In the proposed system, the Predictor agent uses an Ensemble model of several base classifiers such as Support Vector Machines (SVM), Decision Trees (DT), and k-nearest Neighbors (k-NN) stacked together and uses Logistic Regression as the meta-classifier. Figure 2 b) shows the training process, where base classifiers are trained first, followed by the meta-classifier. The Constructor Agent randomly initializes UGV routes, which are then optimized for UAVs, creating a dataset of combined route solutions. This dataset, with a training batch size of $\mathcal{N}$ includes UGV route parameters as input features and binary feasibility labels $1/0$ as output. Feasibility is determined by whether all task points are visited at least once. The Ensemble model, tested with UGV routes generated by a Genetic Algorithm, filters out infeasible routes before UAV optimization, thereby enhancing process efficiency.

*2) **Solving E-VRP for inner-level UAV routing**:* The inner-level UAV route optimization is defined as an Energy-Constrained Vehicle Routing Problem (E-VRP) to obtain a near-optimal path for the UAV subjected to satisfying its fuel and time window constraints. The MILP formulation is as follows. Consider a directed graph $G = (V, E)$ where V is the set of UAV task points $V = \{0, 1, 2, ..., m, D\}$ and E is the set of edges that gives the arc costs between two consecutive nodes i and j and $E = \{(i, j)|i, j \in V, i \neq j\}$. Here D denotes the potential UGV points that UAV could utilize to get recharged. Let c_{ij} be the non-negative arc cost between a particular i and j. Let t_{ij} be the time travel cost between a particular i and j. Let x_{ij} be the binary variable where the value of x_{ij} will be 1 if a vehicle travels from i to j, and 0 otherwise. We formulate the VRP problem with fuel constraints, time windows, and dropped visits. The mathematical formulation for EVRP is presented here, but more details may be found in [18].

Objective:

$$min \sum_{i \in V} \sum_{j \in V} t_{ij} x_{ij} \tag{3.6}$$

Major constraints:

$$f_j \leq f_i - (P^a t_{ij} x_{ij}) + L_1(1 - x_{ij}), \forall i \in V, j \in V \setminus D \tag{3.7}$$

$$f_j = Q, \quad \forall j \in D \tag{3.8}$$

$$0 \leq f_j \leq Q, \quad \forall j \in V \tag{3.9}$$

$$t_j \geq t_i + ((t_{ij} x_{ij})) - L_2(1 - x_{ij}), \forall i \in V, j \in V \tag{3.10}$$

$$t_j^l \leq t_j \leq t_j^u, \quad \forall j \in V \tag{3.11}$$

$$x_{ij} = 1 \rightarrow \sum_{i \in V \setminus D} x_{ji} = 1, \forall j \in D, \forall i \in V \setminus D \tag{3.12}$$

The objective is Eq. 3.6 is to minimize the time to complete UAV routing. The constraint in Eq. 3.7 is the Miller-Tucker-Zemlin (MTZ) formulation [19] for sub-tour elimination which enables that none of the UAVs are fully drained out while eliminating loops. P^a in this equation represents the UAV power consumption curve, which will be provided in Section 4. Constraint Eq. 3.8 states that if the vertex is a recharging UGV stop, UGV must refuel the UAV to its full capacity Q. Constraint Eq. 3.9 is the condition that the UAV's fuel at any vertex in V should be between 0 and maximum fuel capacity. Constraint Eq. 3.10 denotes that the cumulative arrival time at j^{th} node is equal to the sum of cumulative time at the node i, t_i and the travel time between nodes i and j, $t_{ij} x_{ij}$. Constraint Eq. 3.11 is the time window constraint that tells the vehicle to visit a certain vertex in the specified time window for that node. The constraint in Eq. 3.12 ensures that if any UAV comes to the refuel vertex to recharge, an arc must exist between that refuel node and a task node to maintain the flow conservation. The above MILP formulation takes significant time to solve with the increased number of task points, which becomes unrealistic for practical hardware implementation. Ramasamy et al. [20] conducted research that tested the MILP method on various toy scenarios, finding that solving a relatively simpler UGV-UAV routing problem involving 25 task points demanded considerable computational effort, with computation times being up to 30 times slower than the Constrained Programming (CP) approach while the optimality gap is up to 8% on average. Hence, CP with local search heuristics is used to solve this E-VRP problem quickly. More details about this can be found in [20].

C. Hardware setup

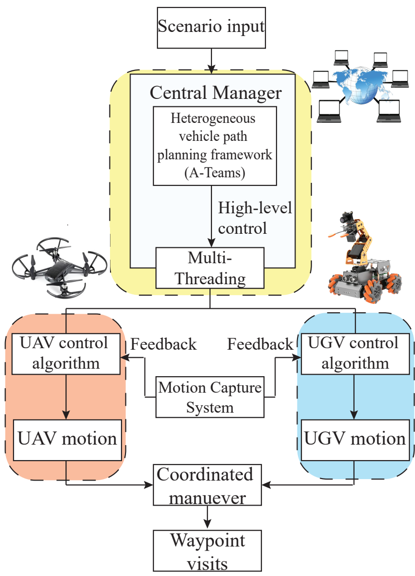

Fig. 3: Hardware experimental framework

Figure 3 displays the lab-based architecture used to perform the hardware experiment. We utilized a DJI Tello drone (80g) and a HiWonder MasterPi robot, controlled via a 2.4GHz WiFi connection and socket communication, respectively. The central manager, integral to the hardware framework, uses A-Teams to determine optimized waypoints for UGV-UAV operations. It generates route sequences in the form of YAML files for each vehicle, transmits pre-planned routes in static scenarios, and adapts with re-planning for dynamic changes. This system allows for parallel processing and online communication with the vehicles. Additionally, a motion capture system monitors the vehicles' trajectories, providing feedback to ensure adherence to the designated paths and coordinating UAV rendezvous with the UGV.

4. RESULTS

A. Evaluation of the proposed framework

The proposed framework is evaluated on different scenarios to test its generalizability, followed by hardware implementation on one scenario. We used Python 3 for all the computations: the Ensemble of different classification algorithms (kNN, SVM, Decision Trees) used in the Predictor agent is from the Scikit-learn package, a custom-written Genetic Algorithm, and Nelder-Mead from the Scipy package for performing UGV free parameter optimization; and OR-Tools for UAV optimization. All computations are done on a 3.7 GHz Intel Core i9 processor with 32 GB RAM on a 64-bit operating system.

Figure 4 depicts three distinct scenarios for simulation, designed to facilitate comprehensive computational analysis through varied spatial settings. To ensure robustness, the initial population for simulation is randomized multiple times for each scenario. The scenarios consider a single UAV and UGV, with the UAV having a battery capacity of 4000 mAh (total energy of $E_a = 287.7kJ$) and the UGV having an energy capacity of $E_g = 25.01MJ$. The UAV and UGV are set to travel at velocities of $v_a = 10m/s$ and $v_g = 4.5m/s$, respectively. The UAV follows the power consumption curve of $P_a = 0.0461(v_a)^3 - 0.5834(v_a)^2 - 1.8761v_a + 229.6$ and UGV follows $P_g = 464.8v_g + 356.3$ (referred from [21]). For the considered scenarios, a candidate UGV route X_s has two primary rendezvous locations ($j = 2$) and three mid UGV-UAV rendezvous locations ($k = 3$), adding up to a total of 5 parameters ($\mathcal{S}$=5) for the UGV route.

The proposed method is compared against existing methods including the standard A-Teams framework without a Predictor agent and a parallelized version of the traditional GA, with GA's stopping condition set at either population convergence or a maximum of $\mathcal{G}$= 20 generations. For A-Teams as well as GA population initialization, the sample size for performing UGV free parameter optimization is considered to be $\mathcal{N}$= 40, according to [22]. Table I shows the optimization metrics and Table II shows the Predictor agent's prediction quality. The simulation results in Table I indicate that A-Teams equipped with the Predictor agent achieve objective results that are comparable to other methods across scenarios, while reducing computational time up to 30% compared to conventional A-Teams and by up to 70% compared to the GA-only meta-heuristics. This demonstrates the framework's generalizability and computational efficiency without sacrificing solution quality compared to standard meta-heuristics. Table II evaluates the prediction quality between two Machine Learning models (proposed Ensemble vs simpler k-NN model) used in the Predictor agent by comparing them across four metrics. Eq. 4.1 outlines those metrics. In the equations, **True Feasible (TF)** represents the number of accurately predicted feasible UGV routes, while **True Infeasible (TI)** correctly identifies the number of infeasible ones. Conversely, **False Infeasible (FI)** occurs when feasible routes are mistakenly labeled infeasible, and **False Feasible (FF)** happens when infeasible routes are wrongly labeled feasible. The table reveals that the Ensemble model outperforms k-NN across all the metrics considered, demonstrating the Predictor agent's capability to differentiate feasible from infeasible UGV-UAV routes accurately.

$$\begin{aligned} \text{Accuracy} &= \frac{TF+TI}{TF+TI+FF+FI}(\%) \\ \text{Precision} &= \frac{TF}{TF+FF}(\%) \\ \text{Recall} &= \frac{TF}{TF+FI}(\%) \\ F\text{-score} &= 2\cdot\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}(\%) \end{aligned} \tag{4.1}$$

B. Hardware re-planning with dynamic changes

Figure 5 shows the scaled scenario considered for performing the hardware experiments with real-time dynamic changes. The chosen scenario for the hardware is scaled down to a 250×250 cm area in the lab. The experiments took place in the well-equipped Robotics and Motion Laboratory

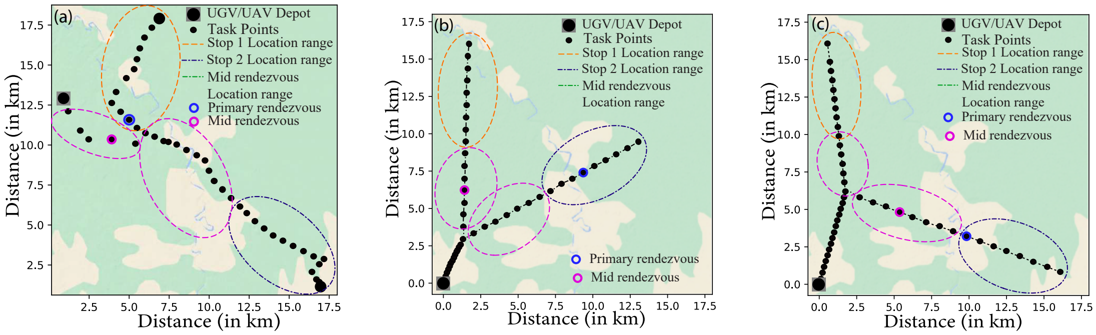

Fig. 4: Scenario descriptions containing the UGV free route parameters for the optimization process. Primary rendezvous location parameter set (orange and blue ellipse) are solved from Minimum Set Cover problem a) Scenario 1 b) Scenario 2 c) Scenario 3

TABLE I: Optimized solution and computational comparison between proposed framework and standard meta-heuristics

Scenario type	**A-Teams with Predictor agent for UGV routing & CP for UAV routing**		**A-Teams without Predictor agent for UGV routing & CP for UAV routing**		**Genetic Algorithm for UGV routing & CP for UAV routing**	
	Objective value (min.)	**Computing Time (min.)**	**Objective value (min.)**	**Computing Time (min.)**	**Objective value (min.)**	**Computing Time (min.)**
Scenario 1	212.3 ± 3.8	9.5 ± 0.6	210.8 ± 3.5	12.8 ± 0.5	212.3 ± 3.8	47 ± 4.4
Scenario 2	233.3 ± 6.9	7.8 ± 1.9	233.8 ± 7.2	11.3 ± 1	228 ± 8.1	39 ± 12
Scenario 3	191 ± 7.8	4 ± 1	190.6 ± 8.1	6.3 ± 1.5	190 ± 8.4	45 ± 7.5

TABLE II: Predictor agent: Assessment of classification performance with proposed Ensemble vs simpler k-NN model

Scenario #	**Accuracy (%)**		**Precision (%)**		**Recall (%)**		**F-score (%)**	
	Ensemble	**k-NN**	**Ensemble**	**k-NN**	**Ensemble**	**k-NN**	**Ensemble**	**k-NN**
Scenario 1	99.3 ± 0.6	99.3 ± 0.6	100 ± 0	100 ± 0	92 ± 6.9	92 ± 6.9	95 ± 4	95 ± 4
Scenario 2	91 ± 11.4	81 ± 1.53	85 ± 26.5	68.3 ± 11	94 ± 7.2	90 ± 7.4	86.6 ± 15	73.3 ± 2.1
Scenario 3	94.3 ± 2.6	90 ± 5.2	83 ± 11.8	80 ± 14.3	89.8 ± 7.2	73.5 ± 12.5	85.5 ± 11.5	76 ± 8.1

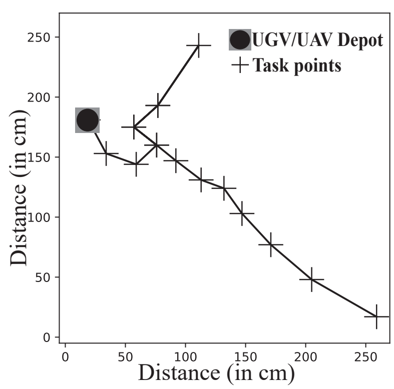

Fig. 5: 2D plot of experimental scenario

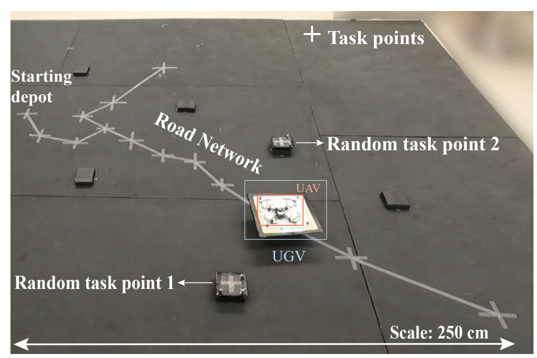

Fig. 6: Experiment instance with dynamic changes

at the University of Illinois Chicago, which includes a high-fidelity motion capture system. To perform the route optimization, the scenario is scaled up to match the required UAV-UGV power consumption and recharging specifications mentioned in subsection 4-A, as those equations pertain to the large-scale scenarios seen in Fig. 4. Once the optimized route plan is obtained, the UAV and UGV task point visit sequence is scaled down accordingly. The vehicle speed for UAV hardware is considered to be 0.15 m/s, and for UGV it is 0.4 m/s. When the UAV lands on UGV, it waits for 'X' seconds to model the recharging time.

Multiple trials of the experiment were conducted. The central manager uses the proposed framework to perform offline route planning in a scenario with no dynamic changes. Once the pre-planned routes are obtained for the system, those routes are fed as commands to the UAV and UGV respectively. For the scenario with dynamic changes, a few random dynamic targets would appear during the middle of the route execution, and the central manager would re-plan the UGV and UAV routes to accommodate those changes. Figure 6 shows the dynamic UAV targets appearing randomly whenever a rendezvous between UAV and UGV is about to happen. Once the re-planning is done, the updated route commands are fed to the vehicle system. The experiment takes only about **1.5 minutes** to provide optimized routes for a planning horizon of **8 minutes**. More details can be seen from the experimental video here: `http://tiny.cc/ngjkxz`. Since the proposed framework delivers near-optimal solutions quickly, the process of re-planning happens online.

In the lab, we scaled down the UAV's recharge and flight times by a factor of 60 (e.g., recharging time was scaled from 15 minutes to 15 seconds). Also, a buffer time

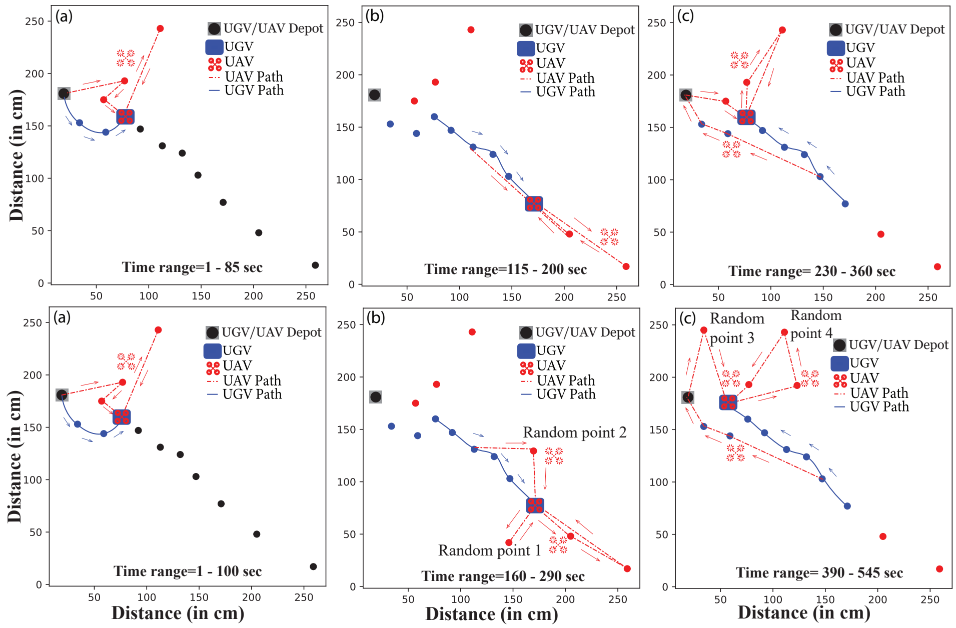

Fig. 7: UAV-UGV simulated route plan obtained from the multi-agent optimization framework (A-Teams). Top Row shows the initial planned route without any dynamic changes. Bottom Row shows re-planned routes considering the dynamic changes in the scenario. Animation of simulated routes can be viewed here: **http://tiny.cc/mgjkxz**

is added to account for the drone's takeoff and landing time in the simulation. After multiple trials, the buffer time to be added in simulation results for takeoff and landing was obtained at 10 and 15 seconds on average. Thus, the maximum flight time for a single charge is replicated to be 50 seconds, denoted as the Maximum endurance limit for a single flight. The time of flight is scaled down from 25 minutes to 25 seconds and an additional time of 25 seconds to accommodate the buffer time for takeoff and landing.

Figure 7 illustrates the simulated route patterns for the case study scenario. The top row shows the static scenario without dynamic changes, where the UAV and UGV follow pre-planned routes. The bottom row depicts adaptive routing under dynamic conditions, with updated route commands adjusting the patterns when changes are detected. For instance, seeing the Figures 7 c) of the top and bottom row, the UAV's route includes additional task points, and the UGV's route is updated accordingly to meet the objective of the problem. These results are validated with hardware experiments, and Figure 8 compares UAV flight and rest durations between simulation and actual hardware. Discrepancies in the figure are noted due to hardware uncertainties. The positive slope in the figure represents UAV takeoff and flight phases, while the negative slope indicates the recharging phase. The figure demonstrates that actual landing and takeoff times may differ from simulations, contributing to overall discrepancies.

5. DISCUSSION

This research work develops a computationally efficient optimization framework that has the ability to plan the cooperative UGV-UAV routing online. The key to computational efficiency comes from the algorithms used in the proposed A-Teams framework, where the Predictor agent used an En-

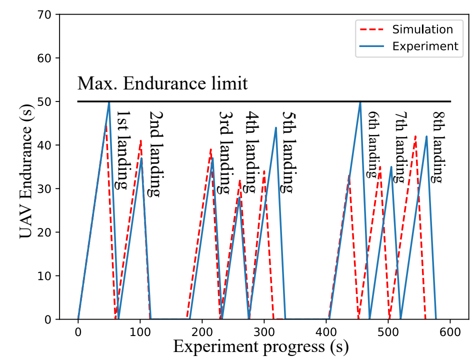

Fig. 8: Hardware vs simulation Time-of-Flight comparison of lab setup experiment

semble Machine Learning model to discard infeasible route solutions. Compared to conventional A-Teams, the proposed framework improves computational efficiency by up to 36% while maintaining solution quality within 1%. Against the Genetic Algorithm, it achieves up to 78% efficiency with solution quality within 2%.

The A-Teams multi-agent framework enhances solutions by integrating different algorithms, each with unique strengths. For instance, pairing a global search like Genetic Algorithm with a local search such as Nelder-Mead, streamlines the search and quickly yields near-optimal solutions, unlike the GA-only method. Also, CP for UAV routing leads to inexpensive optimization using heuristics, which helps achieve an efficient computation. For the Predictor agent, an Ensemble model that integrates several base classifiers such as k-NN, SVM, and Decision trees is chosen for making a robust prediction about discarding the UGV-UAV route solutions that are potentially infeasible. This is because one base classifier might give a contradicting prediction compared to the other. Hence, the Ensemble meta-classifier tries to make a balanced prediction with a more accurate output. In Table II both the Ensemble and k-NN classifiers give the same prediction results for Scenario 1. This is partly due to the distinct spatial distribution of task points in Scenario 1, with drastically different branch lengths connecting the midpoint. This distribution made it easier for the classifiers to discern between feasible and infeasible solutions, resulting in identical predictions.

The proposed work has some limitations. In the context of the optimization framework, the Predictor agent's quality depends upon the base classifiers used, which depend on several hyperparameters. Currently, the Predictor agent uses default hyperparameter settings from Scikit-Learn, such as the 'number of neighbors' parameter N in k-NN set to 5, SVM kernel set to Radial Basis Function (RBF), and 'gini' index for tree splitting in Decision Trees. Poor Predictor Agent may discard good solutions and result in an optimal solution worse than conventional A-Teams. Future work would address this by performing hyperparameter optimization to have the classifiers tailored to this UGV-UAV prediction and thus have a better predictive model. The closeness of hardware results to simulation depends upon the type of hardware used. Since we performed the experiments with a scaled hardware setup in the lab, there are considerable discrepancies between the simulation and hardware results regarding delays in takeoff/landing times of the UAV from/on the UGV. Future efforts will focus on improved hardware, developing a simulation pipeline between routing and hardware deployment, and outdoor testing where there is a large variability compared to indoor testing.

6. CONCLUSION

In this study, a computationally efficient A-Teams framework integrated with a Predictor Agent has been developed. The developed framework is demonstrated to generate near-optimal solutions faster than the existing methods. Furthermore, it was validated in actual setting by conducting a hardware experiment on a case study scenario subjected to dynamic changes.

Our future work will focus on improving the Predictor agent's ensemble model through hyperparameter optimization. Additionally, we will explore alternative objective functions for persistent surveillance, such as minimizing the maximum age period, and extend these improvements to outdoor experiments.

REFERENCES

[1] Pratap Tokekar, Joshua Vander Hook, David Mulla, and Volkan Isler. Sensor planning for a symbiotic uav and ugv system for precision agriculture. *IEEE transactions on robotics*, 32(6):1498–1511, 2016.

[2] Mingjia Zhang, Huawei Liang, and PengFei Zhou. Cooperative route planning for fuel-constrained ugv-uav exploration. In *2022 IEEE International Conference on Unmanned Systems (ICUS)*, pages 1047–1052. IEEE, 2022.

[3] Roberto Baldacci, Maria Battarra, and Daniele Vigo. Routing a heterogeneous fleet of vehicles. *The vehicle routing problem: latest advances and new challenges*, pages 3–27, 2008.

[4] Yao Liu, Zhihao Luo, Zhong Liu, Jianmai Shi, and Guangquan Cheng. Cooperative routing problem for ground vehicle and unmanned aerial vehicle: The application on intelligence, surveillance, and reconnaissance missions. *IEEE Access*, 7:63504–63518, 2019.

[5] Wei Gao, Junren Luo, Wanpeng Zhang, Weilin Yuan, and Zhiyong Liao. Commanding cooperative ugv-uav with nested vehicle routing for emergency resource delivery. *IEEE Access*, 8:215691–215704, 2020.

[6] Sepehr Seyedi, Yasin Yazicioğlu, and Derya Aksaray. Persistent surveillance with energy-constrained uavs and mobile charging stations. *IFAC-PapersOnLine*, 52(20):193–198, 2019.

[7] Christian Blum, Jakob Puchinger, Günther R. Raidl, and Andrea Roli. Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing*, 11(6):4135–4151, 2011.

[8] Sanjay Sachdev. *Explorations in asynchronous teams*. PhD thesis, Carnegie Mellon University, 1998.

[9] Piotr Jedrzejowicz and Ewa Ratajczak-Ropel. Experimental evaluation of a-teams solving resource availability cost problem. In *Intelligent Decision Technologies 2019: Proceedings of the 11th KES International Conference on Intelligent Decision Technologies (KES-IDT 2019), Volume 1*, pages 213–223. Springer, 2019.

[10] Michela Milano and Andrea Roli. Magma: a multiagent architecture for metaheuristics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(2):925–941, 2004.

[11] Nikhil Nigam, Stefan Bieniawski, Ilan Kroo, and John Vian. Control of multiple uavs for persistent surveillance: Algorithm description and hardware demonstration. In *AIAA Infotech@ Aerospace Conference and AIAA Unmanned... Unlimited Conference*, page 1852, 2009.

[12] Nare Karapetyan, Ahmad Bilal Asghar, Amisha Bhaskar, Guangyao Shi, Dinesh Manocha, and Pratap Tokekar. Ag-cvg: Coverage planning with a mobile recharging ugv and an energy-constrained uav. *arXiv preprint arXiv:2310.07621*, 2023.

[13] Jianjun Ni, Xiaotian Wang, Min Tang, Weidong Cao, Pengfei Shi, and Simon X. Yang. An improved real-time path planning method based on dragonfly algorithm for heterogeneous multi-robot system. *IEEE Access*, 8:140558–140568, 2020.

[14] Ning Ma, Yunfeng Cao, Xinyao Wang, Zhaoyang Wang, and Houjun Sun. A fast path re-planning method for uav based on improved a* algorithm. In *2020 3rd International Conference on Unmanned Systems (ICUS)*, pages 462–467, 2020.

[15] Simón Martínez-Rozas, Rafael Rey, David Alejo, Domingo Acedo, José Antonio Cobano, Alejandro Rodríguez-Ramos, Pascual Campoy, Luis Merino, and Fernando Caballero. An aerial/ground robot team for autonomous firefighting in urban gnss-denied scenarios. 2022.

[16] Md Safwan Mondal, Subramanian Ramasamy, James D. Humann, Jean-Paul F. Reddinger, James M. Dotterweich, Marshal A. Childers, and Pranav A. Bhounsule. Cooperative multi-agent planning framework for fuel constrained uav-ugv routing problem, 2023.

[17] Subramanian Ramasamy, Md Safwan Mondal, Jean-Paul F Reddinger, James M Dotterweich, James D Humann, Marshal A Childers, and Pranav A Bhounsule. 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)*, pages 95–102. IEEE, 2023.

[18] Subramanian Ramasamy, Md Safwan Mondal, Jean-Paul F Reddinger, James M Dotterweich, James D Humann, Marshal A Childers, and Pranav A Bhounsule. Heterogenous vehicle routing: comparing parameter tuning using genetic algorithm and bayesian optimization. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 104–113. IEEE, 2022.

[19] C. E. Miller, A. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. *J. ACM*, 7:326–329, 1960.

[20] Subramanian Ramasamy, Jean-Paul F Reddinger, James M Dotterweich, Marshal A Childers, and Pranav A Bhounsule. 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):30, 2022.

[21] Arnon M Hurwitz, James M Dotterweich, and Trevor A Rocks. Mobile robot battery life estimation: battery energy use of an unmanned ground vehicle. In *Energy Harvesting and Storage: Materials, Devices, and Applications XI*, volume 11722, pages 24–40. SPIE, 2021.

[22] Katharine Mullen, David Ardia, David L Gil, Donald Windover, and James Cline. Deoptim: An r package for global optimization by differential evolution. *Journal of Statistical Software*, 40(6):1–26, 2011.