2023 International Conference on Unmanned Aircraft Systems (ICUAS)
June 6-9, 2023 | Lazarski University, Warsaw, Poland

Solving Vehicle Routing Problem for Unmanned Heterogeneous Vehicle Systems using Asynchronous Multi-Agent Architecture (A-teams)

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

2023 International Conference on Unmanned Aircraft Systems (ICUAS) | 979-8-3503-1037-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICUAS57906.2023.10156585

***Abstract*— Fast moving but power hungry unmanned aerial vehicles (UAVs) can recharge on slow-moving unmanned ground vehicles (UGVs) to cooperatively perform tasks over wide areas. Such a cooperation can be achieved efficiently by solving a path planning problem. On top of solving a path planning problem, the problem of routing an heterogeneous set of vehicles in an optimal fashion is quite challenging. In order to solve the computationally expensive path-planning problem in a reasonable time, we created a two-level optimization approach with heuristics. At the outer level, the UGV route is parameterized by considering which set of locations to visit in the scenario and the UGV wait times to recharge UAVs and at the inner level, the UAV route is solved by formulating and solving a vehicle routing problem with capacity constraints, time windows, and dropped visits. The UGV free parameters need to be optimized judiciously in order to create high quality solutions. We explore two methods for tuning the free UGV parameters: (1) a Genetic Algorithm (GA), and (2) Asynchronous Multi-agent architecture (A-teams). The A-teams uses multiple agents to create, improve, and destroy solutions. The parallel asynchronous architecture enables A-teams to quickly optimize the parameters. Our results on test cases show that the A-teams produces similar solutions as GA but is 2-3 times faster.**

1. INTRODUCTION

There has been a considerable increase in the use of the small Unmanned Aerial Vehicles (UAVs) across diverse fields such as entertainment and logistics [1]. The reason for such a widespread adaption is because they are agile robots that can navigate at high speeds in complex environments otherwise inaccessible to humans [23]. Although UAVs are fast, they are limited by their battery capacity to a relatively small area [26].

To complete tasks over wider areas, UAVs could be provided mobile recharging stations that are hosted by unmanned ground vehicles (UGV). Such cooperative routing of a team of UAV-UGVs have been utilized in tasks such as inspection in cluttered environments [8], congested urban environments [5] and post-disaster relief [13], [6].

[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] 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. [3] James D. Humann is with DEVCOM Army Research Laboratory, Los Angeles, CA, 90094 USA. james.d.humann.civ@army.mil † This work was supported by ARO grant W911NF-14-S-003.

The cooperative routing of UAV and mobile recharging stations is complex and computationally challenging [2]. The formulation of the problem involves minimizing a cost such as the time or fuel consumption while constraining the fuel capacity and speed limits of the UAV and UGV and ensuring that they are able to rendezvous efficiently. Although it is relatively easy to formulate the problem, it is difficulty to solve the formulation using exact methods due to the combinatorial nature of the problem. However, using suitable heuristics, it is possible to achieve high quality solutions relatively quickly.

There has been a considerable work done in the literature related to solving fuel-constrained routing of UAVs. Sundar et. al., [24] worked on Fuel-Constrained UAV Routing Problem where a generalization of the asymmetric Traveling Salesman Problem (TSP) is solved using Approximation algorithm and fast heuristics. A Mixed Integer Programming Problem formulation is also proposed to obtain optimal solutions. Here, a single UAV is used and gets recharged on fixed depots. Venkatachalam et. al., [27] modeled a multiple fuel-constrained UAV routing problem with fixed recharging depots. Here, the authors implemented a two-stage stochastic optimization problem with uncertainties in the fuel consumption of UAVs. Heuristics are used to achieve high quality solutions with faster computing time.

Some extensions in the aspect of heterogeneous vehicle routing were also considered in the literature to overcome some of the limitations existed in such fuel-constrained UAV routing problem with fixed depots. Subramanian et. al. [20] considered the vehicle routing problem of multiple fuel-constrained UAVs and a single UGV that acts as a mobile recharging vehicle. The problem is being solved in a tiered fashion. The authors used K-means clustering and TSP to solve UGV routing problem, and then implemented Vehicle Routing Problem (VRP) with fuel, time and optional node constraints. The aforementioned work is extended in [21] where a more generalized approach is taken to solve several different scenarios and proved the robustness of the algorithm. The above works allows the UGV to move freely on any paths, but there are some works in the literature that considers heterogeneous UAV-UGV vehicle routing with UGV constrained to move on certain prescribed paths. This is a challenging problem because each of these vehicles have different constraints on speed and fuel capacity. Maini et al. [14] considered the problem of routing a single fuel-constrained UAV to a set of task locations while being recharged by stopping at a UGV traveling on a road net-

work. They solved the problem using a two-stage approach. First, using the UAV range constraints, they found a set of recharging depots. Second, they formulated a mixed-integer linear program and solved for the path of both the UAV and UGV. Safwan et. al., [16] performs a bi-level optimization on a road network with prescribed UGV paths and the obtained simulation results are validated by performing a lab-setup experiment, which asserts the ability to map from simulation setup to real-time practical deployment. The work in [19] also allows the UGV to move freely only on prescribed paths and is also followed in this paper.

Since UGV has a fixed route, the heuristics for the UGV could be modeled as a parameter tuning problem as that would provide a better solution by tuning the heuristic parameters. [9] applied Bayesian Optimization to tune the hierarchical decomposition algorithm parameters and thus helped to achieve optimal solutions at a faster rate. Although such algorithms help us to achieve globally optimal solutions, they come at a cost of significant computational time. This was seen from the previous work by the authors [19] where GA and Bayesian Optimization (BO) are implemented for parameter tuning to obtain the UGV route. The results from that work show that particularly in case of GA, higher computational time of about 180 minutes was needed to solve that problem. The reason for high computational time is that these algorithms GA and BO are basically global optimization algorithms. Although such global optimization algorithms perform search over a larger space, this compromises their efficiency. Hence they can be used when the computational time is not critical, such as offline optimization [4]. At the same time, local optimization algorithms provide quicker solutions, but are optimal in a small region of space.

To achieve faster global optimal solutions, Sachdev [22] worked on proposing an architecture called A-Teams, which was originally developed by Talukdar et. al. [25]. In A-teams, global optimization methods search over a larger space to find potentially feasible solutions. These are then improved by the local optimization methods. The author implemented two algorithms, Stochastic Quadratic Programming (SQP), a local optimization method, and GA, a global optimization method, in A-Teams to show how the advantages of local and global optimization algorithms can be tapped to produce a better result in a computationally efficient manner. Jedrzejowicz et. al., [11] proposed A-Teams to solve a Resource Constrained Project Scheduling Problem. The authors perform Reinforcement Learning (RL) along with using other optimization algorithms like local search, tabu search to solve the problem using A-Teams. The RL component in their architecture helps to apply dynamic strategy for interaction between those different optimization algorithms in an A-Team. Those authors use a middleware called JABAT (Java Agent DEveleopment-Based A-Teams), to implement the A-Teams architecture. Kazemi et. al., [12] implemented A-Teams for solving a Production-Distribution Planning Problem where each agent in their architecture uses a GA sub-module to handle its tasks and conclude that the combined multi-agent GA system provides better

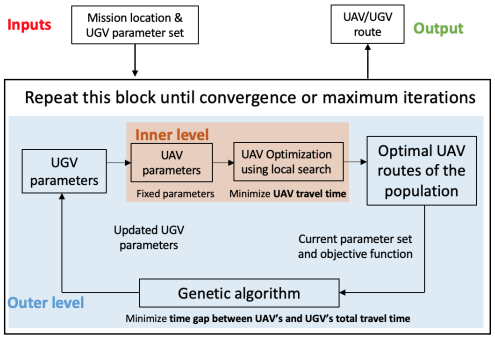

Fig. 1. Overview of the bi-level optimization algorithm. The outer-level block is run in parallel on multiple cores.

solutions than the individual ones for their problem. Recent works by Jedrzejowicz et. al., [10] involves implementing this architecture to solve a Resource Investment Problem in which different agents use Local search, Lagrangian relaxation, Path relinking algorithms, Crossover operators and cooperate together to solve such a problem.

The usage of A-Teams is also found amongst the routing problems in the literature. Rabak et. al. [17] presented the A-Teams framework to optimize the automatic electronic component insertion process on an inserting machine. They implemented a combination of Quadratic Assignment Problem and Traveling Salesman Problem (TSP) in the framework to perform optimization. The above work shows the framework's ability to handle multiple algorithms simultaneously. Such a realization becomes helpful in this work where the A-Teams come in hand for utilizing the local and global optimization algorithms, whose advantages and disadvantages were talked about a few lines before. Barbucha et. al., [3] worked on investigating the effects and impact of a Team of A-Teams working in parallel to solve difficult combinatorial optimization problems. In their work, different algorithms cooperate together within an A-Team, and several similar A-Teams are made to work in parallel. The computational efficiency of their architecture is demonstrated by solving benchmark instances in different problems like Euclidean Planar Traveling Salesman Problem (TSP), Vehicle Routing Problem (VRP), Clustering Problem (CP), and Resource Constrained Project Scheduling Problem (RCPSP). Rachlin et. al., [18] implemented the A-Teams to solve a Traveling Salesman Problem (TSP) by using Farthest Insertion and Arbitrary Insertion heuristic algorithms in their architecture.

The A-teams has been limited to solving basic routing problems (e.g., TSP). Thus, the main contribution of this work is that we use the A-teams architecture to solve a heterogeneous and co-operative vehicle routing problem involving a UAV and a UGV. We also compare the A-teams architecture with results obtained using genetic algorithms in several scenarios. The flow of the paper is as follows. We present details about the optimization method in Sec. 2. The results are in Sec. 3, followed by the Discussion in Sec. 4. Finally, the conclusion and future work is in Sec. 5

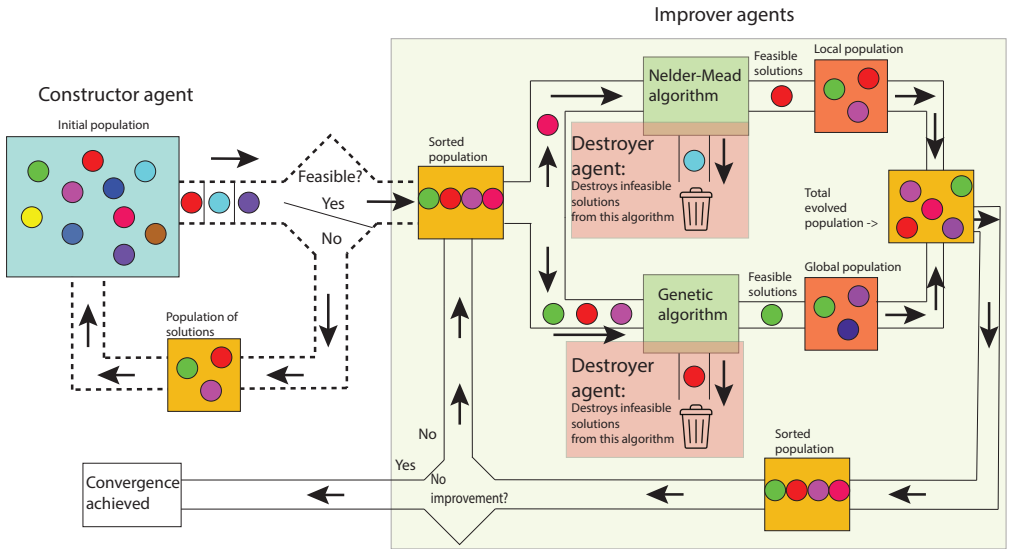

Fig. 2. Implementation of A-Teams architecture for this cooperative routing problem.

2. METHODS

The authors developed a two-level optimization framework [19] that uses Genetic Algorithm and Bayesian optimization and is described in Sec. 2-A. The main contribution of this paper is the A-teams architecture which uses the two-level optimization framework as its basis and is described in Sec. 2-B.

A. Conventional Two-level optimization

The Figure 1 shows the conventional two-level optimization [19]. The outer level block shown in blue are the heuristics to choose a UGV route. The UGV route heuristics has a few free parameters. Once these parameters are set, the inner-level block shown in orange performs the UAV route optimization using Google's OR-Tools^{TM}[7]. The UAV route optimization is heavily dependent on the free parameters of the UGV route. These parameters are optimized using a genetic algorithm. The complete block (inner/outer loop) runs several times till the genetic algorithm can no longer improve the solution or when the maximum iteration limit is reached.

B. Description of the proposed architecture - A-Teams

A-Teams is an architecture that uses a team of autonomous agents to perform optimization on a given problem. The agents have a common set of potential solutions. Each agent works asynchronously on the potential solutions to find better solutions which are then updated as the new potential solutions. There are three main agents that constitute this architecture and are described next.

1. Constructor Agent is used to develop an initial pool of solutions using the user inputs.

2. Improver Agent is used to improve on the pool of solutions using different optimization methods. It is important to choose complementary optimization methods (e.g., global and local optimizers) to help improve the quality of the solutions.

3. Destroyer Agent is used to discard non-optimal and bad solutions. It does this by ranking the solutions based on the cost and constraints satisfaction.

Populations are shared repositories for storing solutions computed and evaluated by different agents. These are accessible by all agents.

The architecture is modular and distributed which enables each optimizer to work independently. However, the architecture also has mechanisms to combine solutions generated by individual optimization to realize further improvements of the solutions. This makes the framework very powerful producing optimal solutions in a computationally efficient manner.

Figure 2 shows the implementation of A-Teams to solve this problem. Note that the A-teams operates over the two-level optimization block shown in Figure 1. The Constructor agent utilizes a 'randomized' initial UGV parameter set to construct an initial population of solutions. The algorithm is used until the feasibility of a solution in the population is achieved. Every time the constructor agent pulls solution from that initial population, it fills up a current population, that was initially empty (represented in orange box in Figure 2), until feasibility. These solutions are usually sub-optimal, but are then passed to the Improver agent. The Improver agents improve the solutions by using two algorithms: (1) the Nelder-Mead, a gradient free direct search method, that is good for local optimization and (2) the Genetic Algorithm that is inspired from natural selection and is good global optimization method. These two algorithms are complementary in nature; Genetic algorithm searches big regions of the parameters set (exploration) while Nelder-Mead improves on the solution in the vicinity of the current solution (exploitation).

We now describe Algorithm 1 used in A-teams. On lines 1 and 2 the constructor agent role is to generate feasible solutions for the improver agent. A random initial parameter

Algorithm 1 A-Teams architecture

Input: Population size, n; Initial population
Output: Global best solution

1: **Constructor agent:** Generate random initial population with population size n;
2: **Constructor agent:** Perform the UAV optimization for corresponding UGV parameter set until feasibility;
3: **while** Convergence is not achieved **do**
4: The following two improver agents work in parallel
5: **Improver agent 1:** Perform Nelder-Mead optimization for local improvement on the current best solution;
6: **Improver agent 2:** Perform Genetic Algorithm optimization for global improvement on the current population;
7: **Destroyer agents 1 and 2:** Destroy the infeasible or already existing solutions on the fly;
8: Replace initial or old population with newly generated population;
9: Compute the fitness value for each population member and sort them in ascending order;
10: **end while**

set for UGV is generated and checked if it leads to a feasible UAV solution. After a sufficient number of good feasible solutions are generated, the algorithm proceeds to the main while loop that uses the constructor and destroyer agent. When the constructor agent produces a feasible solution, the current population which has been updated so far from the initial population is sorted, and the role is handed over to the Improver agents. The solutions are sorted based on the cost and the improver agent uses the global optimizer (GA) shown on line 5 or local optimizer (Nelder-mead) shown on line 6. These improver agents work in parallel. Thus, both, the exploration and the exploitation happens simultaneously and independently. Next, on line 7, the destroyer agent looks at all the solutions and discards the infeasible solutions and those that are already existing in the pool of solutions. Finally, on line 8, all the good solutions are pooled together and sorted in order to get ready for the next iteration. This process repeats until convergence is achieved, i.e., when there is no improvement in the population.

C. Heuristics for UGV (Outer-level)

Our heuristics for UGV route are based on maximum fuel range of the UAV described earlier (range is shown as a blue circle in Figure 3). Figure 4 shows the heuristics for the UGV route. The UGV starts at the depot and travels along the task locations. Next, the UGV is allowed to stop anywhere in the ellipse with dashed red lines for a prescribed time. The rationale is that in choosing a stop and wait time is to give the UAV enough time to land and recharge on the UGV. Next, the UGV moves to to the bottom right side and can take another stop anywhere inside the blue ellipse with blue dash-dot lines. We have shown two random stop locations in each ellipse with a blue hollow circle. There are 7 parameters for the UGV heuristics; the starting location of the UGV/UAV, the x- and y-coordinate of each of the two stop locations, and the wait times at the stops.

D. Optimizing UAV route (Inner-level)

We formulate a Vehicle Routing Problem (VRP) with capacity constraints to account for fuel limits, time windows to allow for rendezvous, and dropped visits to allow the UAV to visit some of the many vertices on the UGV path. We constrain the UAV to a fixed speed, pre-specify the battery capacity and service time as the UAV lands and waits on the UGV. Constrained Programming approach is being used to solve this VRP using OR-Tools solver.

The mathematical details are not included here because of space constraint, but can be found in [19].

3. RESULTS

We used Python 3 for all the computations: a custom-written genetic algorithm and Nelder Mead from Scipy package for UGV parameter optimization, and OR-tools for UAV optimization. All computations were done on a 3.7 GHz Intel Core i9 processor with 32 GB RAM on a 64-bit operating system.

Figure 3 shows the problem scenarios considered in this paper. The task locations are shown with black dots. There are 3 recharging depots shown with a black dot that is bigger than the one used for task locations. The UAV can travel on the UGV or fly by itself. The UAV may be charged by the UGV or at the depot. Both, UAV-UGV start and end at the depot.

The blue circles represent the range of the UAV on a full charge; the distance that the UAV can cover is the diameter of the circle. For example, consider Figure 3 (a). If the UAV starts from the center of the circle on a full charge, it can return back to the center of the circle with an empty charge if it travels straight out and back. We have drawn two circles which are centered approximately at $(1, 12.5)$ km and $(5, 12)$ km. It can be observed that from the start location, the UAV cannot travel to the set of task locations approximately from $(7.5, 10)$ km to $(10, 8)$ km. However, if the UAV starts from the point $(5, 12)$ km with a full charge, it can cover those sets of task locations and return back. To enable this solution, the UAV would need to ride with the UGV along the UGV till $(5, 12)$ km, then visit the task locations within that radius and get refueled. Meanwhile, the UGV stops at $(5, 12)$ km location and waits for some time. Although, such a UGV stop helps to cover additional task locations, there are some task locations along the bottom right region that are left out. Hence, in such case, either the UGV has to have another stop along that region so that UAV can cover those task locations and utilize that UGV stop to recharge or the UGV itself should travel along that path to cover all of those task locations. From this Figure, you can see that all 3 branches intersect at a common point $(6.1, 10.8)$ km. Other scenarios are considered similar to the distribution of this scenario where three different branches meet at a point. This illustrates some of the intricacies of choosing an appropriate

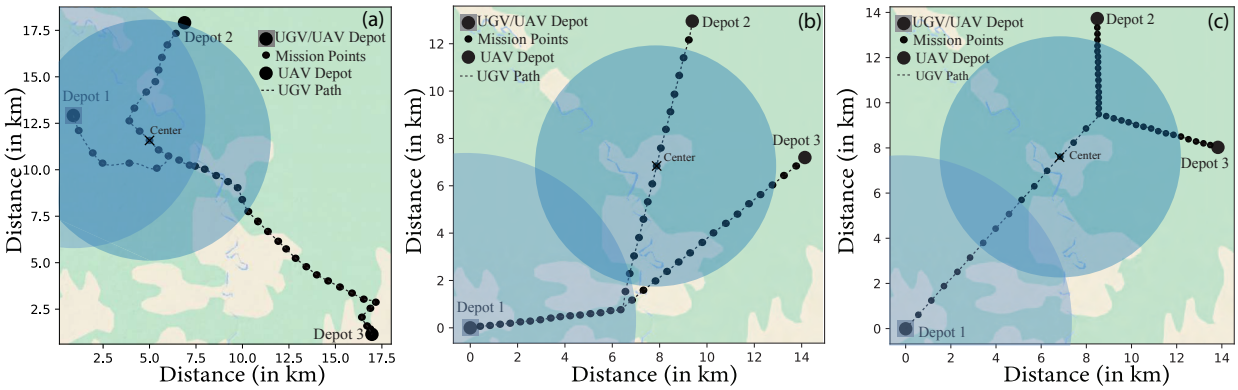

Fig. 3. Description of different scenarios. The UAV and UGV, both start from one of the recharging depots. The task locations are shown with black dots. The UAV range is shown with a blue circle. a) Scenario 1 b) Scenario 2 c) Scenario 3.

Parameter	Range		
	Scenario 1	Scenario 2	Scenario 3
UGV stop 1 (km,km)	(6.02,16.82) to (4.99, 11.65)	(11.36,4.86) to (14.34, 7.31)	(8.95,9.48) to (9.29, 9.38)
UGV stop 2 (km,km)	(14.70, 4.02) to (16.96,1.45)	(7.72,6.13) to (9.46, 13.02)	(8.61,9.82) to (8.61, 10.07)
UGV wait 1,2 (min)	2 to 50	2 to 50	2 to 50
Starting point	1,2, or 3	1,2, or 3	1,2, or 3

TABLE I

UGV PARAMETERS AND THEIR RANGES (OUTER LOOP)

Scenario type	Computational time (in minutes)		Objective (in minutes)	
	A-Teams	Two-level optimization	A-Teams	Two-level optimization
Scenario 1	37 ± 1	47 ± 2	163	166
Scenario 2	28 ± 9	82 ± 10	12	9
Scenario 3	13 ± 3	44 ± 2	18	13

TABLE II

COMPARISON OF TOTAL COST BETWEEN A-TEAMS AND CONVENTIONAL TWO-LEVEL OPTIMIZATION FOR DIFFERENT SCENARIOS

path for the UGV such that the UAV can successfully cover the task locations at the extreme ends. This is an optimization problem where optimal routes are to be found for both UGV and UAV. In case of UGV, its route is modeled as a parameter set consisting of two UGV stop locations to recharge the UAVs, the wait time of UGV at those corresponding stops, and the starting or ending point of the entire route plan. The optimal solution corresponds to a UGV routes parameter set and its subjected UAV route for which the overall objective function is minimized.

In order to prove the computational efficiency, we present the results on three different scenarios shown in Figure 3. The scenarios under consideration have three branches that intersect at a single point. Each scenario has three depots. At each of the these depots, the UAV or the UGV may be recharged. The UAV may also recharged when it lands on the UGV. The UGV/UAV start their route execution from one of the three depots. This location is one of the free UGV parameter. All these scenarios consider the optimization problem for 1 UGV and 1 UAV. The UAV is a custom quadrotor with a battery capacity of 4000 mAh. The UAV and UGV velocities when moving are fixed at 10 m/s and 4 m/s respectively. The UAV and UGV fuel capacity are 287.7 kJ and 25.01 MJ respectively.

Figure 4 shows the UGV parameters for the three scenarios. The black dot on the gray rectangle represents the depot where both UGV and UAV can recharge. The large black circles represents the locations where only the UAV can recharge. Either of those depots represent the potential starting location for the UAV and UGV and is an optimization parameter. The small black circles represent the task locations that need to be visited either by the UGV or the UAV. The stopping locations for the UGV can be either in the red ellipse or the blue ellipse. In each of this ellipse, the x- and y-coordinate is a parameter (2 parameters per ellipse). For each stop location, the wait time is also a free parameter (1 parameter per ellipse). The UAV/UGV may start at Depot 1, 2, or 3 (1 parameter). Table I shows the UGV parameter range for the outer level. The objective function is to minimize the time gap between completion of UAV's and UGV's routes after visiting all their task locations in the respective scenario. This kind of objective function helps to minimize the waiting time between the heterogeneous system after the route execution cycle.

In order to compare the two methods, an initial population size of 30 was chosen and both algorithms were run 3 times. Table II compares the cost and the computational time. It can be seen that the computational time is reduced by a factor of 2 to 3 by the A-teams architecture in comparison to two-level optimization, but the cost is within 25%.

Table III compares the A-teams solution with two-level optimization for the same initial population for the three different scenarios. From the value of the objective in the table it can be seen that the results are mixed: A-teams is better than two-level optimization for scenario 1 but not for scenario 2 and 3. However, the difference between the two is not substantially large. The other metrics such as the UAV/UGV travel time, energy consumed, recharging stops,

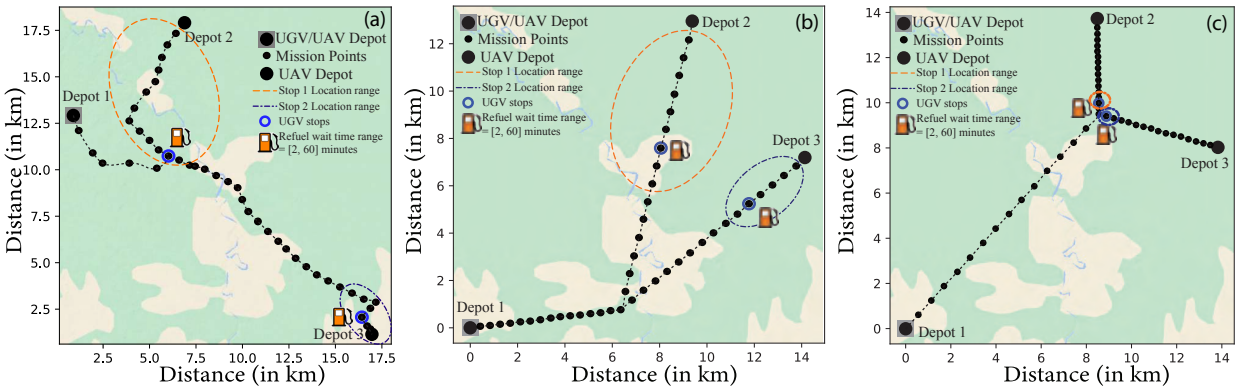

Fig. 4. Description of different scenarios with parameters to be optimized. a) Scenario 1 b) Scenario 2 c) Scenario 3.

Parameter	Optimal parameter values					
	Scenario 1		Scenario 2		Scenario 3	
	A-Teams	Two-level	A-Teams	Two-level	A-Teams	Two-level
UGV stop 1 location (km,km)	(4.99,11.65)	(4.99,11.65)	(7.92,6.90)	(11.36,4.86)	(8.61,10.08)	(8.95,9.48)
UGV stop 2 location (km,km)	(16.96,1.45)	(16.96,1.45)	(12.36,5.68)	(8.30,8.43)	(9.29,9.38)	(8.61,10.08)
UGV stop 1 wait time (min)	20	20	50	21	20	22
UGV stop 2 wait time (min)	20	21	20	21	20	20
Route starting and ending point	Depot 1	Depot 1	Depot 3	Depot 3	Depot 3	Depot 2
Metrics	Scenario 1		Scenario 2		Scenario 3	
	A-Teams	Two-level	A-Teams	Two-level	A-Teams	Two-level
Objective function (min)	163	166	12	9	18	13
Total time (min)	228	231	216	201	145	131
UGV results						
Travel time (minutes)	228	231	216	201	145	131
Energy consumed (MJ)	23.16	23.19	19.50	20.89	8.16	6.31
# Locations visited	34	34	23	25	17	17
UAV results						
Travel time (minutes)	65	65	204	192	127	118
Energy consumed (kJ)	460.65	460.65	1082.92	1301.78	841.72	874.73
Recharging stops on UGV	1	1	2	3	2	2
Recharging stops on Depot	0	0	2	2	1	1
# Locations visited	10	10	23	21	29	29

TABLE III

COMPARISON BETWEEN A-TEAMS AND CONVENTIONAL TWO-LEVEL OPTIMIZATION ON METRICS FROM THE OPTIMAL SOLUTION FOR DIFFERENT SCENARIOS. THESE RESULTS ARE SHOWN FOR A SPECIFIC INITIALIZATION OF POPULATION BY CONSTRUCTOR AGENT.

locations visited are also shown. There are minor differences between the two.

Figure 5 shows the final solution for scenario 1 at three different time ranges (in min): $1-35$, $36-65$, and $66-166$. Since both two-level and A-teams produce very similar solution, only one solution, the two-level optimization, is shown here. The total routing time for A-teams is less than that of the two-level optimization by about 3 min. The UAV/UGV start at Depot 1 then they move together to the first stop location. Here the UAV flies to cover the locations on the top portion returning to Depot 1 to recharge. Then the UGV travels to all the task locations and returns back to the Depot 1. Figure 6 shows the coverage of task locations and the recharging stops used by the UAV-UGV for the A-teams for scenario 1. The task locations in red are those that are covered by the UAV while those in blue are the ones covered by the UGV. The red cross shows the stopping locations for the UAV on the UGV for recharging. The overlaid light blue circles indicate the range of the UAV. It can be seen that the recharging stops are chosen strategically to enable maximum fuel coverage for the UAV on a single charge.

4. DISCUSSION

This paper presented the A-teams framework for optimizing the routes of a UAV-UGV pair subject to fuel and speed constraints. The A-teams framework uses asynchronous agents to create an initial pool of solutions, improve the pool, and then destroy the infeasible solutions. These agent exploit parallel architecture to produce fast solutions. When compared with conventional optimization method, A-teams produces the solution $2-3$ times faster while achieving similar quality of solutions.

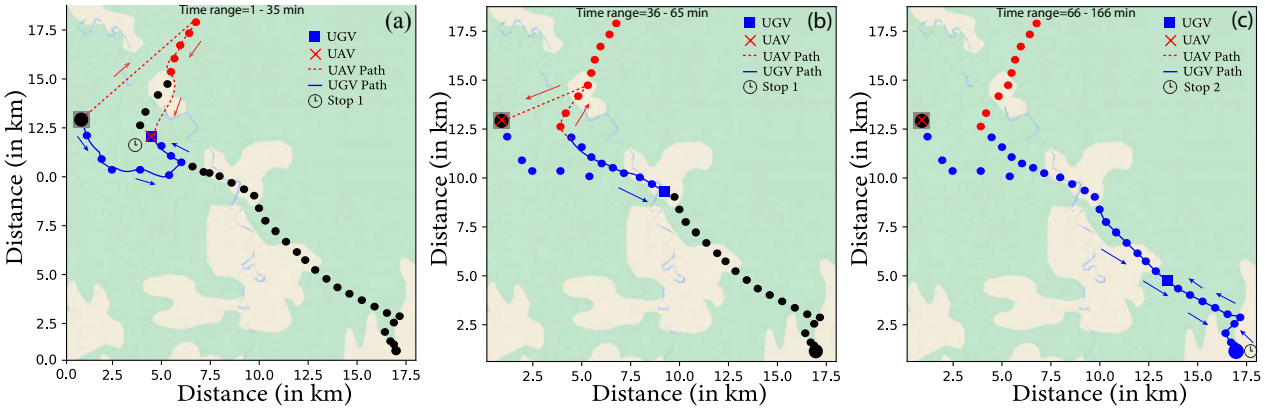

Fig. 5. Solution produced by conventional two-level optimization and A-teams on Scenario 1 are indistinguishable and are shown here. The different plot shows the UAV and UGV route at various time-steps.

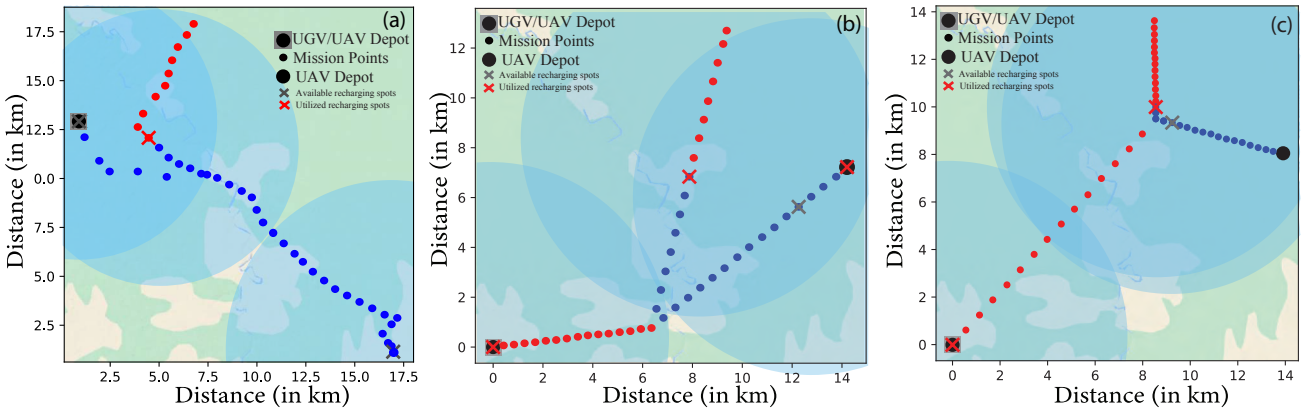

Fig. 6. Optimal parameter results of respective scenarios obtained using the A-Teams architecture. (a) Scenario 1 (b) Scenario 2 (c) Scenario 3

One advantage of A-teams is the use of asynchronous agents to improve the solutions. These agents work in parallel and hence they can be deployed independent of each other. When the algorithm is deployed either on multiple core or parallel computing machines, they are able to speed up computations. Apart from that, A-Teams has the capability to scale to multi-cores where each core could be used for performing specific parallelized A-Teams computation with threads in that core. This way, a combined effort of searching for the optimal solution can be made efficiently.

Another advantage of A-teams is that the architecture seamlessly exploits the advantages of multiple algorithms to improve the solution. In our case, the genetic algorithm is used to explore the search space while the Nelder-Mead is used to locally improve the solution. Thus, we have combined a global search with local search to improve the solution quality. However, genetic algorithm is not sample efficient. One could use a sample efficient method like Bayesian Optimization if sample efficiency is important [19].

The proposed works has some disadvantages. The UGV heuristics, the stop location and wait times, were manually determined by hand tuning. This may be overcome by using minimum set cover algorithm [15]. The quality of the initial pool of solutions created by the constructor agent is critical to ensure that the improver agent is able to improve the solution. Thus, we had to play with a few random initial guesses till we got a feasible solution as a starting base. Our results indicate that the A-teams produce superior solutions for complex scenarios (Scenario 1), but was unable to produce better solutions that our baseline method of using genetic algorithms in simple scenarios (Scenario 2 and 3) were able to. This might indicate that the more complex A-teams architecture might not be ideal for certain scenarios whose task space distribution is simple, or the number and combination of free parameters (such as stop locations, waiting times) used for the optimization problem is very large.

5. CONCLUSIONS AND FUTURE WORK

We conclude that Asynchronous multi-agent architecture (A-teams) is a competitive tool for solving the cooperative heterogeneous Vehicle Routing Problem. A-teams is able to produce good quality solutions with less computation time. It is able to do so by using specialized agents: agents to create solutions, agents to improve solutions globally and locally, and agents to destroy bad solutions.

Our future work will explore methods to automate the choosing of UGV parameters, testing the scalability of the approach by adding more UGV parameters, more UAVS and UGVs, and testing other algorithms such as Bayesian or

reinforcement learning to improve the quality of the solutions as well as the solution time.

REFERENCES

[1] Yaniv Altshuler, Alex Pentland, and Alfred M Bruckstein. Optimal dynamic coverage infrastructure for large-scale fleets of reconnaissance uavs. In *Swarms and Network Intelligence in Search*, pages 207–238. Springer, 2018.

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

[3] Dariusz Barbucha, Ireneusz Czarnowski, Piotr Jkedrzejowicz, Ewa Ratajczak-Ropel, and Izabela Wierzbowska. Team of a-teams-a study of the cooperation between program agents solving difficult optimization problems. In *Agent-based optimization*, pages 123–141. Springer, 2013.

[4] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[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] Abenezer Girma, Niloofar Bahadori, Mrinmoy Sarkar, Tadewos G Tadewos, Mohammad R Behnia, M Nabil Mahmoud, Ali Karimoddini, and Abdollah Homaifar. Iot-enabled autonomous system collaboration for disaster-area management. *IEEE/CAA Journal of Automatica Sinica*, 7(5):1249–1262, 2020.

[7] Google. Google OR-tools. `https://developers.google.com/optimization`, 2021. Online; accessed Feb 2, 2021.

[8] Difeng Hu, Vincent JL Gan, Tao Wang, and Ling Ma. Multi-agent robotic system (mars) for uav-ugv path planning and automatic sensory data collection in cluttered environments. *Building and Environment*, 221:109349, 2022.

[9] Changwu Huang, Bo Yuan, Yuanxiang Li, and Xin Yao. Automatic parameter tuning using bayesian optimization method. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 2090–2097. IEEE, 2019.

[10] Piotr Jedrzejowicz and Ewa Ratajczak-Ropel. Experimental evaluation of a-teams solving resource availability cost problem. In *Intelligent Decision Technologies 2019*, pages 213–223. Springer, 2020.

[11] Piotr Jkedrzejowicz and Ewa Ratajczak-Ropel. Reinforcement learning strategy for solving the resource-constrained project scheduling problem by a team of a-teams. In *Asian Conference on Intelligent Information and Database Systems*, pages 197–206. Springer, 2014.

[12] A Kazemi, MH Fazel Zarandi, and SM Moattar Husseini. A multi-agent system to solve the production–distribution planning problem for a supply chain: a genetic algorithm approach. *The International Journal of Advanced Manufacturing Technology*, 44(1):180–193, 2009.

[13] Abderrahmane Lakas, Boumediene Belkhouche, Omar Benkraouda, Amin Shuaib, and Hussain Jaffar Alasmawi. A framework for a cooperative uav-ugv system for path discovery and planning. In *2018 International Conference on Innovations in Information Technology (IIT)*, pages 42–46. IEEE, 2018.

[14] Parikshit Maini, Kaarthik Sundar, Sivakumar Rathinam, and PB Sujit. Cooperative planning for fuel-constrained aerial vehicles and ground-based refueling vehicles for large-scale coverage. *arXiv preprint arXiv:1805.04417*, 2018.

[15] Parikshit Maini, Kaarthik Sundar, Mandeep Singh, Sivakumar Rathinam, and PB Sujit. Cooperative aerial–ground vehicle route planning with fuel constraints for coverage applications. *IEEE Transactions on Aerospace and Electronic Systems*, 55(6):3016–3028, 2019.

[16] Md Safwan Mondal, Subramanian Ramasamy, and Pranav Bhounsule. A bilevel optimization framework for fuel-constrained uav-ugv cooperative routing: Planning and experimental validation. *arXiv preprint arXiv:2303.02315*, 2023.

[17] César Scarpini Rabak and Jaime Simão Sichman. Using a-teams to optimize automatic insertion of electronic components. *Advanced Engineering Informatics*, 17(2):95–106, 2003.

[19] 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.

[20] Subramanian Ramasamy, Jean-Paul F Reddinger, James M Dotterweich, Marshal A Childers, and Pranav A Bhounsule. 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)*, pages 155–164. IEEE, 2021.

[21] 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):1–18, 2022.

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

[23] Yunlong Song and Davide Scaramuzza. Policy search for model predictive control with application to agile drone flight. *IEEE Transactions on Robotics*, 2022.

[24] Kaarthik Sundar and Sivakumar Rathinam. Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots. *IEEE Transactions on Automation Science and Engineering*, 11(1):287–294, 2013.

[25] Sarosh Talukdar, VC Ramesh, et al. Cooperative methods for security planning. 1992.

[26] Mirco Theile, Harald Bayerlein, Richard Nai, David Gesbert, and Marco Caccamo. Uav coverage path planning under varying power constraints using deep reinforcement learning. *arXiv preprint arXiv:2003.02609*, 2020.

[27] Saravanan Venkatachalam, Kaarthik Sundar, and Sivakumar Rathinam. A two-stage approach for routing multiple unmanned aerial vehicles with stochastic fuel consumption. *Sensors*, 18(11):3756, 2018.