2024 International Conference on Unmanned Aircraft Systems (ICUAS)
June 4-7, 2024. Chania, Crete, Greece

Optimizing Routes of Heterogenous Unmanned Systems using Supervised Learning in a Multi-Agent Framework: A computational study

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

2024 International Conference on Unmanned Aircraft Systems (ICUAS) | 979-8-3503-5788-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICUAS60882.2024.10557087

Abstract— **Fast-paced but power-hungry Unmanned Aerial Vehicles (UAV) may collaborate with slow-paced Unmanned Ground Vehicles (UGV) acting as mobile recharging depots to perform large-scale and long-duration tasks such as in disaster relief management. It is important to be able to create high-quality vehicle routes in a short span of time to enable in-field, real-time deployment. A two-level optimization enables a tractable approach to solving such NP-hard combinatorial optimization problems, and it consists of an outer-level UGV route optimization to compute recharge locations and an inner-level UAV optimization to compute a sequence of nodes and recharging nodes to be visited. We consider various approaches to solve this two-level optimization. Method 1: Genetic algorithm for global search of UGV routes and a constraint programming solver for the UAV routes. Method 2: A-Teams framework that uses a combination of Genetic Algorithm (GA) for global search and Nelder-Mead for local search for UGV routes and constraint programming for inner routes, Method 3: Our proposed A-Teams by adding a supervised learning prediction step to veto out suboptimal evolutions from GA to the UGV route and constraint programming for the inner level, Method 4: Our proposed A-Teams for outer level but with a mixed integer programming solver for the inner-level. Our results on test cases show that the proposed Method 3 produces an optimal solution 30% faster than Method 2, 79% faster than Method 1, and 83% faster than Method 4 while being within 2% of the solution optimality across these methods.**

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.

This research work focuses on developing optimization algorithms that apply to tasks such as surveillance and disaster relief. To perform such tasks of high endurance over wider areas, UAVs could be paired up with Unmanned Ground Vehicles (UGVs) to provide them with a mobile recharging base [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 a combinatorial optimization problem and is deemed to be NP-hard [3]. In a practical setup, where the environment dynamics are constantly changing, the UAV and UGV routes need to be re-planned, which necessitates relatively fast solution methods. There is a critical need to develop methods that can compute optimal solutions in real time. In this paper, we provide a multi-agent framework to be able to solve this computationally challenging problem relatively fast, reaching near-optimal solutions.

[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] James M. Dotterweich, Jean-Paul F. Reddinger, 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.

2. RELATED WORK

1) System models: Liu et al. [4] solved a cooperative Ground Vehicle (GV)-UAV routing problem for surveillance and reconnaissance missions. The authors proposed a Two-Echelon Ground vehicle and UAV Cooperative Routing Problem (2E-GUCRP) and solved using Split Heuristic. A big UAV tour is first constructed without considering the constraint on the UAV's battery capacity; then the UAV route is split into feasible segments to construct the UAV route. The open ends of those feasible segments are then closed with GV to form a complete GV-UAV route. As a preliminary study, Ramasamy et al. [5] considered the vehicle routing problem of multiple fuel-constrained UAVs and a single UGV that acts as a mobile recharging vehicle. The problem was solved in a tiered fashion. K-means clustering and TSP were used to solve the UGV routing problem, and the UAV route was formulated as a Vehicle Routing Problem (VRP) with fuel, time, and optional node constraints. Mondal et al. [6] performed a fuel-constrained cooperative UGV-UAV routing problem using a bi-level planner. The Minimum Set Cover (MSC) algorithm and Traveling Salesman Problem (TSP) were used to solve the UGV route in the outer loop, and a Vehicle Routing Problem (VRP) was solved for inner UAV routing. A related work is on persistent surveillance, which is the problem of continuous monitoring of nodes using UGV-UAV systems. Yu Wu et al. [7] have performed a cooperative path planning of UGV-UAV in urban envi-

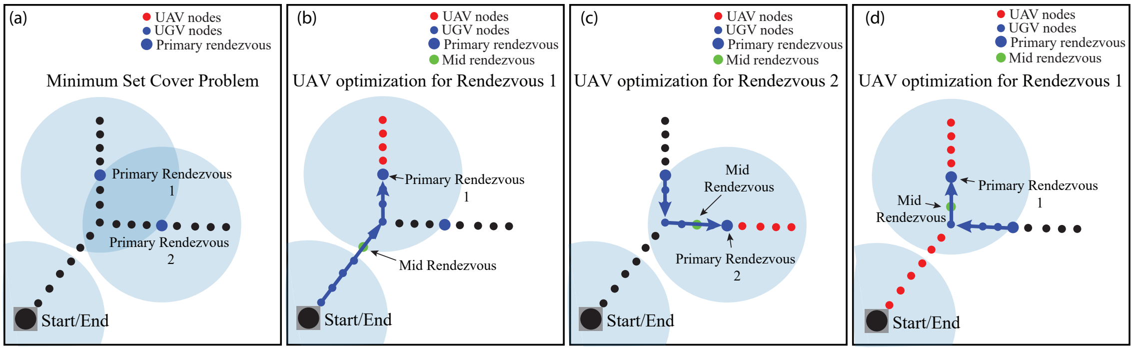

Fig. 1: Illustration of performing persistent surveillance for a collaborative UGV-UAV routing problem. Steps (c) and (d) are repeated for persistent node visits until UGV runs out of fuel.

ronments using a customized hybrid EDA-GA algorithm (Estimation of Distribution Algorithm-Genetic Algorithm). The UAV and UGV have to visit a set of respective waypoints in a circular fashion persistently over an urban setting. Wang et al. [8] proposed a novel path-planning framework using a path-planning algorithm and a set of decision rules to perform persistent surveillance using an Unmanned Ground Vehicle. The process begins by determining the shortest path between two target viewpoints using the Floyd-Warshall algorithm. Then, a set of decision rules was incorporated to adjust the path based on the data from that target viewpoint pair, iterating this method for each pair to cover all points.

2) Asynchronous Teams (A-Teams) optimization: The proposed research work deals with improving a multi-agent optimization framework called Asynchronous Teams (A-Teams) [9]. A-Teams is a multi-algorithm framework that uses a team of different agents such as Constructor, Improver, and Destroyer agents together to evolve a solution pool towards optimal results. Jedrzejowicz et al., [10] implemented this framework 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. Kazemi et al., [11] 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 solutions than the individual ones for their problem. Ramasamy et al. [12] used the A-Teams for solving a cooperative UGV-UAV routing problem, and the authors showed the superior performance of A-Teams compared to the Genetic Algorithm (GA). Thelasingha et. al. [13] solved a problem of cooperative UGV-UAV route planning by developing an iterative planning framework capable of handling several optimization solvers to solve the route optimization problem. The framework employed a 'Transition System' for problem formulation, defining nodes, edges, a combination of node connections, and output. The solvers within the framework utilized those data to improve and provide better solutions iteratively. The authors pointed out the usage of different solvers in the framework to guarantee recursive feasibility.

3) Other multi-agent optimization approaches: Other multi-agent optimization frameworks are used across the literature for solving various combinatorial optimization problems. Milano and Roli et al. [14] introduced MultiAGent 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 heuristics, 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. Similarly, Fernandes et al. [15] and Silva et al. [16] presented a multi-agent framework called A Multi-agent Architecture for Metaheuristic (AMAM) in which each agent implemented a metaheuristic to solve a certain problem. Those agents interacted with the environment, where the agents exist and act with other agents cooperatively, exchanging and sharing information about their condition and about the environment. The main agents of this framework were Constructor, Local search, Metaheuristic and Coordinator agents.

Out of all these multi-agent optimization frameworks, there is a lack of exploration in utilizing machine learning-based approaches that cooperate with agents in the framework. This research addresses these issues by integrating a novel **Predictor Agent** within the A-Teams framework to improve computational efficiency without sacrificing the quality of the solution. The contribution of this research work is as follows: 1. Solving a UGV-UAV routing problem for Persistent surveillance using a variant of the multi-agent A-Teams framework. To this end, this proposed variant of A-Teams is a novel method to solve such persistent surveillance problems; 2. Augmenting a unique agent (Predictor agent) into the framework.

3. METHODOLOGY

A. Problem statement

Figure 1 gives an example scenario of the type of problem considered in this paper. The goal of this research work is to perform collaborative UGV-UAV routing where UGV and UAV survey a set of designated task points persistently $\mathcal{M} = \{m_0, m_1, ..., m_n\}$ in remote areas where UGV acts as a

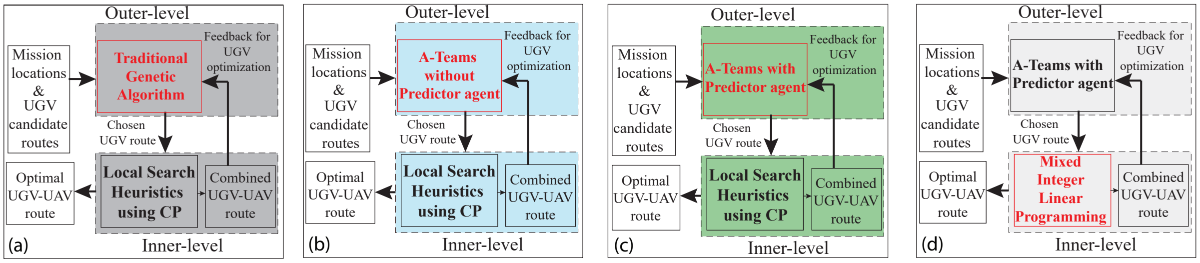

Fig. 2: Overview of different methods of bi-level optimization algorithm. The red highlighted blocks show the difference across the methods involved. a) Method 1: GA for UGV optimization and Local search for UAV optimization. b) Method 2: Conventional A-Teams for UGV optimization and Local search for UAV optimization. c) Method 3: Proposed A-Teams for UGV optimization and Local search for UAV optimization. d) Method 4: Proposed A-Teams for UGV optimization and MILP for UAV optimization

mobile recharging vehicle for UAV along with task visits. Those task points are shown with black dots in Figure 1. The UGV and UAV start from the depot D. The UAV can either get recharged on UGV or at D. Once the 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. 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, which has a larger energy capacity E^g to recharge the UAV and power its components. The bigger blue circles represent the range of the UAV with full energy capacity E^a. If the UAV starts from the center of the circle at a full charge, it can return to the center of the circle with an empty charge. As it can be seen from Figure 1 a), a single circle is unable to cover the task point space completely, and hence the need for a UGV becomes important. We have to determine strategic UGV locations so that the UAV coverage covers all UAV task points at least once and can make a rendezvous with UGV. 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. A complete parameter set of 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 towards the primary rendezvous locations. The total number of route parameters is represented as $\mathcal{S} = j + k$. To identify primary rendezvous points, the Minimum Set Cover (MSC) Problem is solved using Constrained Programming, as detailed by Mondal et al. [6]. This approach tackles the NP-Hard nature of the MSC problem, allowing for multiple local optima for X_p depending on the scenario's scale and complexity such that $|X_p| = j$. Mid-rendezvous points (X_m), used for UAV recharging, are randomly selected between the X_p locations such that $|X_m| = k$. The k is considered to be $k = j + 1$, accounting for an extra point between the starting depot and the first X_p location, leading to $X_s = X_p \cup X_m$ as the complete parameter set for each candidate UGV route. A bi-level optimization framework is proposed to solve the collaborative UGV-UAV routing problem for persistent surveillance because the UAV routes depend upon the UGV routes. Thus, if the UGV route is formulated at the outer level, the UAV route can be formulated at the inner level and be subjected to satisfying the UGV constraints. Blue arrows in Figures 1 b)-d) shows one candidate UGV route. For each candidate UGV route, the inner-level UAV route optimization is performed by formulating them as Energy Constrained Vehicle Routing Problem (E-VRP). Red dots in Figures 1 b)-d) shows the UAV visit nodes for that corresponding UGV route. The obtained UAV route might be feasible or infeasible for that corresponding candidate UGV route. This process is repeated for all candidate UGV routes. By giving the UAV's route output as feedback to a certain UGV route, the overall process is repeated to perform overall UGV-UAV optimization. Figure 1 from a)-d) constitutes one UGV-UAV route solution.

The overall objective of this optimization problem is minimizing the total time T to perform persistent surveillance of UGV-UAV task points until the UGV runs out of fuel. Thus, a collaborative UGV-UAV routing solution that persistently surveys all task points is optimized via UGV optimization until the optimal objective value is obtained. Due to the specificity of the problem nature and the scenarios considered, the assumption is that the persistent surveillance is made sure 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. This penalty will make sure to perform persistent surveillance. The objective function for the UGV-UAV routing is denoted mathematically as follows. Here $|v(m_i)|$ denotes the number of visits of a point m_i where, $m_i \in \mathcal{M}$.

$$min \quad T + P_t \tag{3.1}$$

where,

$$P = \begin{cases} 0, & \text{if } |v(m_i)| > 1 \text{ for atleast one } m_i, \\ L, & \text{otherwise, where } L \text{ is a large number} \end{cases} \tag{3.2}$$

This paper implements four distinct methods to optimize the respective UGV and UAV routes, including our proposed method amongst them. Details about those methods are given in the coming subsections.

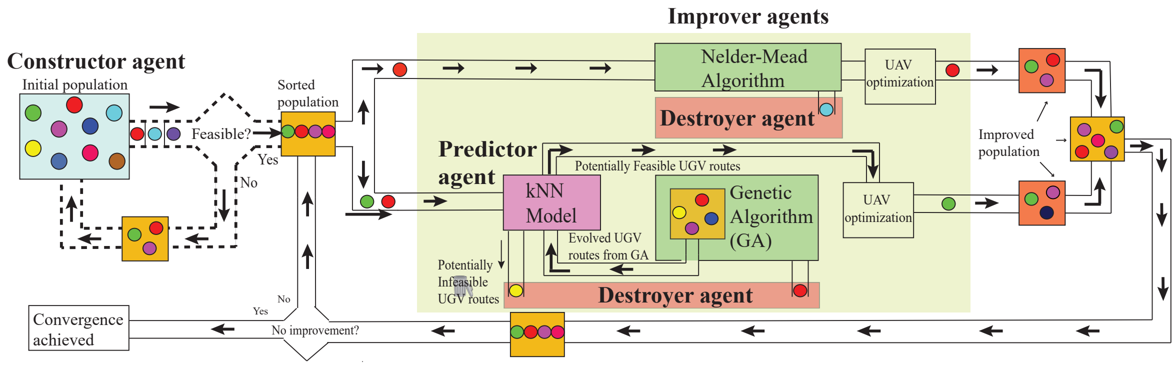

Fig. 3: Description of the proposed A-Teams optimization framework used for UGV-UAV routing. This proposed A-Teams has Constructor, Improver, Destroyer and Predictor agents. Agents strategically choose UGV routes, which are fed into UAV optimization to get collaborative UGV-UAV route outputs.

B. Method 1: GA at outer-level and Constraint Programming (CP) at inner-level

Genetic algorithms (GAs) are meta-heuristic techniques inspired by natural selection, effectively solving global optimization problems that may contain multiple local optima. Figure 2 a) shows the two-level optimization with Traditional GA for UGV optimization and CP for UAV optimization. In this case, the outer-level block at the top implements GA to choose a UGV route. The UGV route heuristics is formulated as a set of free parameters depicting candidate UGV routes. For each UGV route sent, the inner-level block optimizes the UAV route using Google's OR-Tools solver [17]. OR-Tools uses local search heuristics and solves using the CP approach. Some of the constraints, such as time constraints on the UAV, come from the UGV route.UAV route optimization is closely linked to UGV route optimization, which utilizes genetic algorithms (GAs) based on the feasibility of the combined UGV-UAV route. A route is deemed feasible if it covers all mission points during persistent surveillance. The process iterates until the GA can no longer improve the solution or reaches the maximum iteration limit. The sample size of the population is considered to be $\mathcal{N} = 40$, which is chosen as per [18].

C. Method 2: Conventional A-Teams at outer-level and CP at inner-level

A-Teams is a multi-agent framework that uses a team of autonomous agents to optimize a given problem. The agents in the framework possess distinct functionalities, solve problems through cooperation, and achieve solutions that are better than their individual counterparts. There are main core agents that constitute this framework: **Constructor Agent** is used to develop an initial pool of solutions using the construction algorithms or randomization. **Improver Agent** is used to improve the pool of solutions obtained from Constructor Agent using different optimization methods. **Destroyer Agent** is used to discard non-optimal, bad, and redundant solutions. **Populations** are shared repositories for storing solutions that are computed and evaluated by different agents. Those solutions are also accessible to all the agents involved in the framework.

Figure 2 b) shows the two-level optimization using A-Teams and CP. For solving this collaborative UGV-UAV routing problem, our previous work [12] has implemented this framework for similar kind of scenarios where the Constructor Agent creates an initial population by random initialization of candidate UGV routes using Latin Hypercube Sampling (LHS). The sample size of the Constructor agent is considered to be $\mathcal{N} = 40$. Those UGV route parameters are sent to the inner-level block and UAV optimization is solved using OR-Tools. Based upon sending the feasibility of the obtained UGV-UAV route as feedback, the Improver Agents improve the UGV route until convergence is obtained, therefore giving an optimal collaborative route solution. Here, Nelder-Mead and GA are used as Improver agents.

D. Method 3: Proposed A-Teams Variant - A-Teams with Predictor agent at outer-level and CP at inner-level

This subsection explains the novelty that is being imparted in this multi-agent framework. Figure 3 illustrates the A-Teams variant with the additional agent. The previous works using A-Teams so far [12] had Constructor, Improver, and Destroyer agents, which use algorithms suitable for this problem. Practical implementation on hardware demands the ability of an optimization process to be fast enough to implement. Considering that, the fusion of Machine Learning (ML) adds a layer of intelligence, particularly through the introduction of the Predictor agent, thus augmenting an additional component to the existing array of A-Team agents in the proposed framework (Figure 3). The primary role of the Predictor agent is to forecast whether a given UGV route parameter is likely to lead to an infeasible UGV-UAV routing solution, thus enabling us to avoid unnecessary computational efforts. This enhancement significantly enables the computational efficiency of the optimization process. Within the Predictor agent, we employ a k-Nearest Neighbors (k-NN) classification approach to make these predictions. Since the UAV route depends upon the UGV route, each UGV route from the Constructor Agent is fed into UAV optimization to get optimal UAV routes for the corresponding UGV route. So a single combined UGV-UAV route is treated as a solution. The k-NN model is trained on a dataset

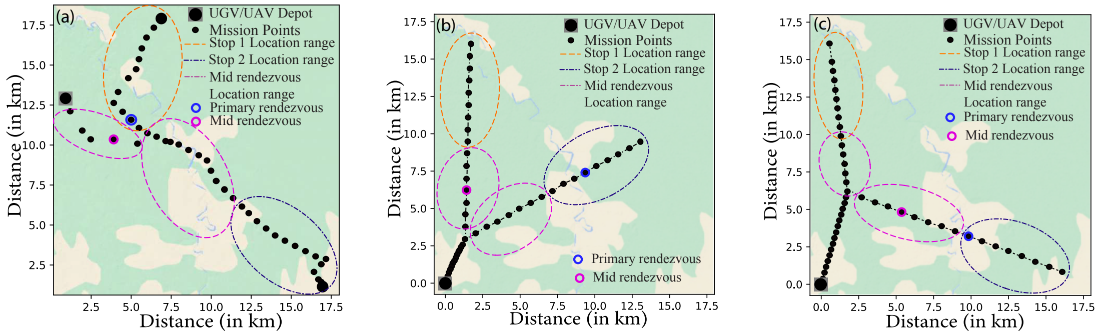

Fig. 4: Scenario descriptions containing the UGV free route parameters for the optimization process. Primary rendezvous location parameter set (red and blue ellipse) are solved from Minimum Set Cover problem a) Scenario 1 b) Scenario 2 c) Scenario 3

comprising such solutions evaluated from the Constructor agent. Similar to the conventional A-Teams method described before, the Constructor Agent creates the initial population by random initialization of candidate UGV routes using Latin Hypercube Sampling (LHS) for the problem considered, the Improver Agents consist of Nelder-Mead and Genetic Algorithm, and the Destroyer Agent destroys bad solutions. The batch size for **training** is $\mathcal{N}= 40$, which comes from the sample size of the Constructor Agent considered in this problem. The dataset's inputs consist of UGV route parameters, while the outputs are UGV-UAV collaborative route solutions. We assign binary class labels (1/0) to indicate the feasibility of the UGV-UAV solution. Similar to previous methods, a solution is labeled as feasible if it ensures that all task points $\mathcal{M}$ within the scenario are visited at least once. Once the k-NN model is trained on the Constructor agent's data, it is put to **testing** using UGV route parameters generated by the GA of sample size N, as illustrated in Figure 3 b). In each iteration, the UGV route parameters produced by GA are fed into the k-NN model. The model identifies and discards the UGV route that might lead to potentially infeasible solutions for UAVs. This selective process ensures that only UGV route parameters with the potential to yield an overall feasible route solution are considered for further UAV optimization. The algorithm for k-NN classification is represented in Algorithm 1. In this ML model, the number of neighbors is set to the default parameter value from the Scikit-learn package($k = 5$). Such k-NN-selected UGV route parameters are fed into the inner-level block where the UAV optimization is carried out through the CP method using the OR-Tools solver. Figure 2 c) shows the two-level optimization using this Proposed method.

The A-Teams framework is modular and distributed, enabling each optimizer to work independently; hence, parallel computing is enabled. Additionally, the proposed framework incorporates various optimization algorithms and machine learning models to enhance its search efficiency, leading to computationally effective optimal solutions that are more efficient than the existing A-Teams methods. This integration significantly boosts the framework's capability to deliver high-quality solutions efficiently.

Algorithm 1 k-NN classification

Input: X: Initial UGV routes of $\mathcal{P}$ dimension, Y: Feasibility labels, k:Number of neighbors
Output: Predict feasibility of GA-evolved UGV route 'x'
Train k-NN model using X and Y;
for each evolved UGV route from GA 'x' **do**
  for each UGV route X **do**
    Calculate the distance: $d(x, X) \leftarrow \sqrt{\sum_{i=1}^{n}(x_i - X_i)^2}$;
    Aggregate distances of all X from x
  end for
  Obtain k nearest neighbors to x based on distances from all X obtained;
  Assign the most frequent feasibility label to 'x';
end for

E. Method 4: Proposed A-Teams with Predictor agent at outer-level and MILP at inner-level

This method uses the same outer-level UGV optimization algorithm as explained in section 3-D but the inner-level transitions from CP to MILP, as highlighted in the bottom block of Figure 2 d). The inner-level UAV route optimization is defined as Energy Constrained Vehicle Routing Problem (E-VRP) and formulated as a MILP problem. Consider a directed graph $G = (V, E)$ where V is the entire set of vertices or nodes $V = \{0, 1, 2, ..., m, D\}$ and E is the set of edges that gives the arc costs between i and j and $E = \{(i, j)|i, j \in V, i \neq j\}$. Here, D denotes the potential rechargeing points that the 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. This problem is solved repeatedly between consecutive UGV-UAV rendezvous. Due to the interest of space, some of the important constraints considered in this problem are represented as follows. The detailed representation of the MILP formulation for this E-VRP can be found in [19].

UAV Objective:

$$min \sum_{i \in V} \sum_{j \in V} t_{ij} x_{ij} \tag{3.3}$$

Major constraints:

$$f_j \leq f_i - (P^a t_{ij} x_{ij}) + L_1(1 - x_{ij}), \forall i \in V, j \in V \setminus D \tag{3.4}$$

$$f_j = Q, \quad \forall j \in D \tag{3.5}$$

$$0 \leq f_j \leq Q, \quad \forall j \in V \tag{3.6}$$

$$t_j \geq t_i + ((t_{ij} x_{ij})) - L_2(1 - x_{ij}), \forall i \in V, j \in V \tag{3.7}$$

$$t^l_j \leq t_j \leq t^u_j, \quad \forall j \in V \tag{3.8}$$

$$x_{ij} = 1 \rightarrow \sum_{i \in V \setminus D} x_{ji} = 1, \forall j \in D, \forall i \in V \setminus D \tag{3.9}$$

The objective in Eq. 3.3 is to minimize the time to complete a single leg of UAV routing. A single leg includes the UAV's takeoff, visits to specific nodes, and subsequent landing on the UGV. The constraint in Eq. 3.4 is the Miller-Tucker-Zemlin (MTZ) formulation [20] for sub-tour elimination. This constraint represents the energy consumption of the UAV, which ensures that the UAV's energy is not fully drained out during its visits. P^a in this equation represents the power consumption curve of UAV, which will be provided in the Results section. Constraint Eq. 3.5 states that if the node is a recharging UGV stop, then UGV has to recharge the UAV to its full capacity Q. Constraint Eq. 3.6 is the condition that the UAV's fuel at any node in V should be between 0 and maximum fuel capacity. Constraint Eq. 3.7 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. 3.8 is the time window constraint that tells the UAV to visit a certain node in the specified time window for that node. While recharging, the UGV travels to the recharging nodes, allowing the UAV to take off or land on those recharge nodes whenever the UGV reaches within the corresponding time window defined by this constraint. The constraint in Eq. 3.9 ensures that if any UAV comes to the recharge node, an arc must exist between that recharge node and a task node to maintain the flow conservation. Once the UAV comes to the recharge node, it gets recharged on the UGV. In our problem, a realistic consideration is kept where the recharging time depends on the existing charge present in the UAV. If the fuel level is low, recharging will take longer. Hence, the profile of the power transfer from UGV to UAV should be considered based on the fuel level present in the UAV. A first-order approximation of the battery recharge rate, P_r is given by:

$$P_r = \begin{cases} 310.8; \quad \text{E} \leq 270.4 \text{ kJ}, \\ 17.9(287.7 - E); 270.4 < \text{E} \leq 287.7\text{kJ}. \end{cases} \tag{3.10}$$

where E represents the existing energy level on the UAV when it docks to recharge. The power transfer uses constant current until 94% of the battery capacity, with a 3.5C charge rate. After 94% capacity, it switches to a constant voltage charge. From the existing fuel level on the UAV and computing this battery recharge rate, when the UAV is charged to its maximum capacity, we would get the recharging time of the UAV.

The above MILP formulation takes a lot of time to solve a UAV route, which becomes unrealistic for practical hardware implementation. This will also be analyzed and compared in the next section to understand the computation time taken by such MILP method.

4. RESULTS

The proposed framework is applied to different scenarios to test its robustness and generalizability. We used Python 3 for all the computations: the k-NN classification used in Predictor agent is from Scikit-learn package [21], a custom-written GA, and Nelder-Mead from Scipy package [22] for performing UGV free parameter optimization; and OR-Tools (CP with Local Search Heuristics), Gurobi optimizer [23] (MILP) 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 represents 3 scenarios considered, and each of them is spatially different to make an extensive computational analysis. The UAV, as well as the UGV, can visit the task points. The UGV-UAV optimal route visit starts and ends at the **Depot**. At times, when UGV performs its route, the UAV rendezvous with the UGV to get recharged, and the UGV and UAV move together while recharging. The time of recharge of the UAV is translated into the distance they move together on the road network.

UGV-UAV optimization deals with optimizing the UGV route from a population of UGV routes and optimizing the UAV route corresponding to a candidate UGV route. 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 each UGV route. Those primary rendezvous locations are represented in red and blue ellipse regions (1 parameter per ellipse) in Figure 4 and the remaining 3 rendezvous locations happen anywhere bounded by the primary rendezvous locations (1 parameter between each of the 3 branches). Thus, for these scenarios considered, the total number of UGV parameters $\mathcal{P}$ is 5. All these scenarios consider the optimization problem for 1 UGV and 1 UAV. For simulation purposes, these scenarios handle a UAV with a battery capacity of 4000 mAh which has a total energy of $E^a = 287.7kJ$, a UGV with an energy capacity of $E^g = 25.01MJ$, and the UAV and UGV velocities when moving are fixed at $v^a = 10m/s$ 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 $P^g = 464.8v^g + 356.3$.

The objective of this problem is to minimize the time to perform persistent routing by the UGV-UAV system to visit all the task points in the space until the UGV runs out of its fuel. The four methods discussed in Sec. 3 are compared. The traditional GA-only method is parallelized to have a fair comparison with the proposed method, and the termination criteria for the GA-only method is based on from work [24] with a maximum generation being $\mathcal{G}$= 20. For A-Teams as well as GA population initialization, the sample size for performing UGV free parameter optimization is considered

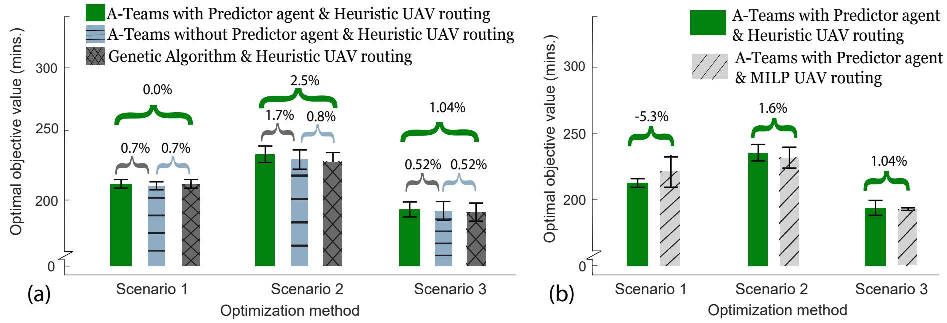

Fig. 5: Comparison of optimal objective value obtained through different optimization methods implemented across different scenarios. The percentage values show the optimality gap. a) Comparison results across different outer-level UGV optimization methods with same inner-level UAV routing. b) Comparison results across different inner-level UAV optimization methods with same outer-level UGV routing.

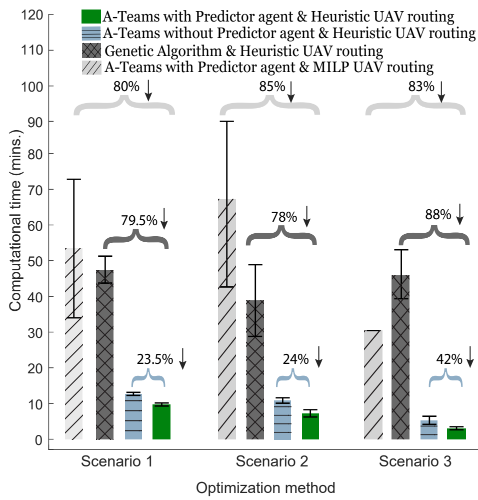

Fig. 6: Computational time analysis of different optimization methods across different scenarios. The percentage values and arrows show the reduction in simulation time of the proposed method compared to existing ones.

TABLE I: UGV and UAV metrics obtained for Persistent Surveillance with proposed method

Parameter	Optimal parameter values		
	Scenario 1	Scenario 2	Scenario 3
UGV primary rendezvous 1 location (km,km)	(6.1,10.8)	(7.16, 6.2)	(9.82,3.19)
UGV primary rendezvous 2 location (km,km)	(14.69, 4.02)	(1.51,10.28)	(1.09, 11.75)
UGV mid rendezvous 1 location (km,km)	(2.48, 10.43)	(0.32,0.74)	(0.22, 0.77)
UGV mid rendezvous 2 location (km,km)	(10.9, 7.29)	(1.38,5.39)	(3.56, 5.52)
UGV mid rendezvous 3 location (km,km)	(10.01, 8.45)	(3.51,4.17)	-
Metrics	**Scenario 1**	**Scenario 2**	**Scenario 3**
Objective function (min)	209	223	187
Total time (min)	209	223	187
UGV results			
Travel time (minutes)	209	223	187
Energy consumed (MJ)	21.92	21.80	23.91
# Locations visited	28	33	34
UAV results			
Travel time (minutes)	209	223	187
Energy consumed (MJ)	1.12	1.13	0.80
Recharging stops on UGV	5	5	3
Recharging stops on Depot	1	0	0
# Locations visited	25	30	29

to be $\mathcal{N}=$ 40, according to [18]. In order to test the framework's robustness, the initial population is randomized to 4 different seeds for each scenario, hence amounting to a total of 12 overall simulations across all scenarios. Figure 6 compares the computational time across 4 above-mentioned methods for the scenarios considered. In this case, the percentage reduction is calculated by evaluating the difference in computational time or value between the proposed method and other methods, relative to the value of the proposed method. Figure 5 compares the optimality gap of those methods. Figure 5 a) compares different UGV optimization methods with a heuristic method for UAV optimization and Figure 5 b) compares the MILP (exact) solution method with a heuristic method for UAV optimization. In this case, the percentage values are similarly calculated by finding the difference in optimal solution between the analyzed method and others relative to the value of the analyzed method. From the results, it can be seen that the proposed A-Teams with Predictor agent and inner-level heuristics help to perform efficient optimization by producing similar near-optimal solutions comparable to other methods, but is 30% faster than conventional A-Teams, 79% faster than GA and 83% faster than the current method with MILP at inner-level. Figures 5 a) & b) validate the above statement because the optimality gap is very low with the current method, which suggests the computational efficiency is achieved without compromising on the solution quality. For instance, comparing the A-Teams with the Predictor agent against those without the Predictor agent, it can be seen that the difference in solution quality is within 3% on average, whilst the computational time reduction is around 30% on average, suggesting a 90% improvement. The efficiency comes from the Predictor agent used, which uses the Machine Learning model and hence plays a crucial role. Figure 7 represents the confusion matrix, which assesses the quality of the k-NN model being used in this work. The Y-axis represents the feasibility of actual candidate UGV-UAV route solutions upon evaluation, and the X-axis represents the feasibility of predicted candidate solutions by k-NN. Zero (0) represents infeasible and one (1) represents feasible solutions. The accuracy of the k-NN classification model varies between 83% to 99% across these

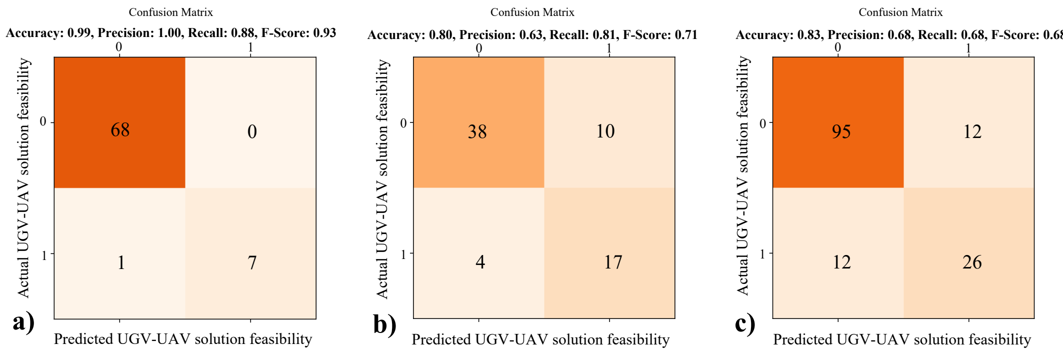

Fig. 7: Confusion matrix assessing the Predictor agent's accuracy by comparing its predictions against actual feasibility results of UGV-UAV routing for each Scenario. The prediction process happens within the GA part of the A-Teams framework as mentioned in Figure 3 a) Scenario 1 b) Scenario 2 c) Scenario 3

scenarios, which shows that the k-NN model is good at distinguishing between the feasible and infeasible UGV-UAV routes proactively. The accuracy of this result indicates the extent to which a predicted feasible solution deviates from becoming infeasible upon actual evaluation through inner-level feedback. Having good accuracy shows the benefit of implementing a Predictor agent in the existing framework. Table I shows a particular optimal solution output obtained with the proposed method for the scenarios considered.

5. DISCUSSION

This paper contributes a novel update to the A-Teams framework by proposing a Predictor Agent. The predictor agent uses supervised learning to cluster the best solutions and use this trained data to seed the global search method. The efficacy of the approach is demonstrated for optimizing the routes of the UGV-UAV system subject to fuel, time, and other flow constraints. The proposed algorithm is able to achieve solutions that are within 2% of the optimal solution produced using the exact method but takes about $30 - 80\%$ less time.

From Figure 6, the conventional A-Teams framework has greater computational performance than GA, but is not fast enough to perform real-time computation on hardware. The main motive is to produce the optimal solution result more efficiently without compromising on the solution quality, and this proposed method does that by reducing the computational time up to 42% from conventional A-Teams, making it viable for hardware implementation. The capability of the proposed method to perform efficient computation can help to perform re-planning of the UGV-UAV routes in real-time when there are dynamic changes in the environment like obstacles or additional task visits.

The A-Teams multi-agent framework enhances solutions by integrating different algorithms, each with unique strengths. For instance, pairing a global search like GA with a local search such as Nelder-Mead, as demonstrated in this work, streamlines the search and quickly yields optimal solutions, unlike using the GA-only method. Also, MILP for UAV optimization leads to intensive computations and CP reduces it by using heuristics, which helps to achieve an efficient computation overall. In terms of the Predictor agent, the k-NN model is chosen for its simplicity and interpretability. It makes no assumptions about the underlying data distribution, making its use for real world data. The training of the k-NN depends upon the population size and it was quicker in this work due to a moderate population size.

The proposed work has some limitations. Although the inclusion of a Predictor agent leads to optimal solutions in a computationally efficient manner, from Figure 7, it can be seen that the precision and recall are slightly lower on such predictions across all the scenarios. Low precision means there are some UGV route parameters that are predicted to give feasible UGV-UAV routes, but turned out to become infeasible after the actual UAV optimization evaluation for that corresponding UGV route. Also, in the case of Scenario 3, the recall is lower as well, which compromises on neglecting feasible UGV-UAV route solutions that are potentially optimal. High accuracy alone can be misleading, especially with imbalanced classes, making precision and recall essential for a thorough assessment of the ML model. Another limitation is that the solution quality relies on the Constructor agent's initial UGV routes. The kNN model's performance depends on the appropriate 'number of neighbors (N)' parameter, which was set to the Python package's default in this work (N=5). Fine-tuning N will be addressed in future research.

6. CONCLUSION AND FUTURE WORK

We conclude that integrating a novel Predictor Agent into the existing A-Teams optimization framework improves the computational efficiency by observing a 30% to 83% reduction in computing time for solving the UGV-UAV persistent surveillance problem, with UGV routing using A-Teams and Predictor at the outer level and UAV routing using local search heuristics through CP at the inner-level. This method greatly helps in practical settings to perform real-time tasks that aid humans, such as surveillance or disaster relief management.

Our future work will address the present issue in Predictor agents by exploring more advanced ML models, such as Ensemble Learning to improve the overall quality of the ML model used in it. This helps to achieve significant improvement in optimization and thereby subsequently extending the framework to hardware experiments to harness its potential for realistic practical applications.

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] 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.

[6] Md Safwan Mondal, Subramanian Ramasamy, James D Humann, Jean-Paul F Reddinger, James M Dotterweich, Marshal A Childers, and Pranav Bhounsule. Optimizing fuel-constrained uav-ugv routes for large scale coverage: Bilevel planning in heterogeneous multi-agent systems.

[7] Yu Wu, Shaobo Wu, and Xinting Hu. Cooperative path planning of uavs & ugvs for a persistent surveillance task in urban environments. *IEEE Internet of Things Journal*, 8(6):4906–4919, 2020.

[8] Tong Wang, Panfeng Huang, and Gangqi Dong. Modeling and path planning for persistent surveillance by unmanned ground vehicle. *IEEE Transactions on Automation Science and Engineering*, 18(4):1615–1625, 2020.

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

[10] 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.

[11] 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*, 44:180–193, 2009.

[12] 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.

[13] Neelanga Thelasingha, Agung Julius, James Humann, Jean-Paul Reddinger, James Dotterweich, and Marshal Childers. Iterative planning for multi-agent systems: An application in energy-aware uav-ugv cooperative task site assignments. *arXiv preprint arXiv:2401.08846*, 2024.

[14] 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.

[15] Filipe Costa Fernandes, Sérgio Ricardo de Souza, Maria Amélia Lopes Silva, Henrique Elias Borges, and Fábio Fernandes Ribeiro. A multiagent architecture for solving combinatorial optimization problems through metaheuristics. In *2009 IEEE International Conference on Systems, Man and Cybernetics*, pages 3071–3076. IEEE, 2009.

[16] Maria Amélia Lopes Silva. Modelagem de uma arquitetura multiagente para a solução, via metaheurísticas, de problemas de otimização combinatória. *Dissertacção de mestrado, Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG), Belo Horizonte, Brazil*, 2007.

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

[18] 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.

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

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[22] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[23] Gurobi. Gurobi Optimization LLC. `https://www.gurobi.com/`, 2021. Online; accessed Sep 19, 2021.

[24] Martín Safe, Jessica Carballido, Ignacio Ponzoni, and Nélida Brignole. On stopping criteria for genetic algorithms. In *Advances in Artificial Intelligence–SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings 17*, pages 405–413. Springer, 2004.