2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Multi-robot Search in a 3D Environment with Intersection System Constraints

Yan-Shuo Li[1] and Kuo-Shih Tseng[2]

***Abstract*— Efficient task allocation is a challenge for multi-robot search. The multi-robot search problem is reformulated as submodular maximization subject to intersection system constraints. The objective function is submodular and consists of a coverage function to cover environments and a balancing function to efficiently dispatch robots. The intersection system is composed of routing and clustering constraints. The experiment results show that the proposed approach outperforms state-of-the-art methods in multi-robot search.**

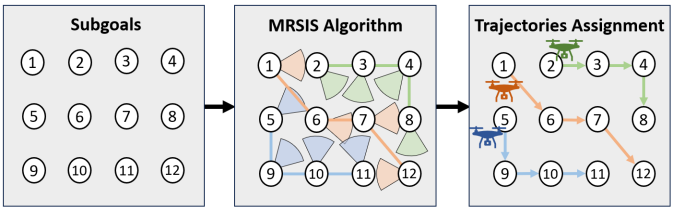
I. Introduction

Multi-robot search problems involve multi-robot task allocation. The challenges of these problems are as follows. First, finding an optimal solution is computational intractability, since the task allocation problem is NP-hard [1]. Second, the workload balance of assigned robots is another issue [2]. Third, each robot in multi-robot search problems needs to solve the traveling salesman problem (TSP) [3].

State-of-the-art approaches formulate task allocation problems as Markov Decision Processes (MDP) over graphs. In [4], the researchers propose a reinforcement learning algorithm based on a graph neural network. However, real-world factors, such as environmental uncertainties and partially observable state spaces, are not considered. Furthermore, since the path planning is not considered, robot trajectories are inefficient. In [5], the researchers propose a probability density function (PDF) as a reward function and generate a sequence of decisions with a reinforcement learning method. Yet, the balance of the robots' workloads is not taken into account, potentially resulting in poor task assignments.

To resolve the aforementioned issues in multi-robot search, this research proposes a Multi-Robot Search with an Intersection System (MRSIS) algorithm, where balanced workloads and minimal routes are considered. MRSIS is to maximize the coverage and balance functions subject to an intersection system. The maximal coverage function is to cover the environment while the maximal balance function is to assign the workloads of robots equally. Besides, a minimum spanning tree (MST) is constructed for each robot routing. Since the resource is limited, the clustering and routing constraints are formulated as an independence system, which finds the largest independent set that satisfies certain conditions. If the independence system has the exchange property, it is a matroid, which leads to better theoretical guarantees for greedy algorithms [6].

[1]Yan-Shuo Li is a graduate student in the Mathematics Department at National Central University, Taiwan. yanshuolee102@gmail.com
[2]Kuo-Shih Tseng is with the Faculty of the Mathematics Department at National Central University, Taiwan. kuoshih@math.ncu.edu.tw



(a) (b) (c)

Fig. 1: Overview of the proposed multi-robot search system with three robots. (a) An example of subgoals (nodes) in a search environment. (b) The trajectories for each robot (orange, blue, and green lines) are obtained by the proposed algorithm (MRSIS). The circular sectors are the coverage areas. (c) The trajectories are assigned to robots to find targets in the environment.

The proposed MRSIS method is illustrated in Fig. 1. Given an environment map with subgoals, the goal is to find all targets with multiple robots as soon as possible. In Fig. 1(a), subgoals are evenly distributed in the space. A complete graph $\mathcal{G}(V, E)$ is constructed, where V and E represent the subgoal set and the Euclidean distance between any two subgoals, respectively. In Fig. 1(b), the MRSIS generates a set of trajectories that maximize the environment coverage and maintain balanced workloads among robots. In Fig. 1(c), trajectories are then assigned to robots to search for targets in the environment.

The contributions of this research are as follows: First, the multi-robot search problem is reformulated as the submodular maximization with an intersection system. To the best of our knowledge, this is the first work to reformulate the multi-robot search problem in this form. Second, the proposed objective function and constraints are proven to be a submodular function and independence systems, respectively. The key novelty of this paper is the integration of clustering and routing into an intersection system. Third, the experiment results show that the proposed method (MRSIS) outperforms state-of-the-art approaches in the multi-robot search problem.

This paper is organized as follows. Section II reviews the relevant works on target search methods, multi-robot task allocations, and routing constraints. Section III describes the background knowledge of this research. Section IV introduces the problem formulation. Section V describes the search algorithm. Section VI describes the experiments and analyzes the results. Finally, Section VII draws conclusions and outlines future work.

II. Related work

In this section, recent works on target search methods, multi-robot task allocations, and routing constraints are reviewed.

A. Target Search

Searching for targets in an environment is a sequential decision-making problem. Therefore, search methods based on the type of decision-making algorithm, such as the next-best-view search, the probabilistic search, and the Partially Observable Markov Decision Processes (POMDP), can be adopted to generate the search strategy.

The next-best-view (NBV) planning determines the next viewpoint that provides the most valuable information to improve search efficiency. Lauri et al. [7] propose a submodular utility function for multi-sensor NBV planning under partition matroid constraints. In addition, the utility function coordinates view selection and prevents overlapping views among multiple sensors.

The probabilistic approach is to estimate target location due to sensor and target motion uncertainty. The robot makes decisions based on the probability distribution. Sheng et al. [5] propose PD-FAC that decomposes the PDF of the multi-robot system into a set of individual value distributions. It is guaranteed that the objective function of the overall system's value distribution can be linearly approximated by the same reliability metric defined over the agent's individual value distribution.

The POMDP further considers the search problem where the state of the target locations and the robot sensor is uncertain. Zhu et al. [8] propose a Dec-POMDP method to find a target in an environment with obstacles. The approach provides a scalable framework for a large number of UAVs. It enables the UAV swarm to cooperate efficiently by sharing limited observations in the mission.

B. Multi-Robot Task Allocation (MRTA)

The goal of MRTA is to optimize an objective function within a given budget for multiple robots. However, finding an optimal solution is NP-hard [1]. Several research studies have employed submodular maximization with matroid constraints to solve the MRTA problem. This problem involves various challenges, such as the orienteering problem [9], the intermittent deployment problem [10], and the capacitated vehicle routing problem [11].

If the objective function is submodular, greedy algorithms can find solutions with theoretical guarantees [6]. In the team surviving orienteers (TSO) problems [12], an independent set of a matroid is selected for maximizing the expected visited nodes at least one robot. These nodes also ensure that each vehicle reaches its destination with probabilities above a specified threshold. In environmental monitoring applications [13], the multi-robot task allocation problem and the multi-robot intermittent deployment problem are formulated as submodular maximization with two matroid constraints.

C. Routing Constraints

Submodular optimization has been explored with extensive applications in various domains. To apply submodular optimization in realistic environments, different constraints (e.g., cardinality [6], additive budget [14], and routing [3]) should be considered.

Zhang et al. [3] propose a generalized cost-benefit (GCB) greedy algorithm to solve two NP-hard problems: submodular maximization and routing path minimization (TSP). Therefore, adopting approximation algorithms (e.g., nearest neighbor) for TSP is a way. GCB finds solutions that achieves $\frac{1}{2}(1-\frac{1}{e})\widetilde{OPT}$ optimum, where $\widetilde{OPT}$ is the approximated optimum. However, there is a gap between $\widetilde{OPT}$ and the optimal solution (OPT). To improve the theoretical guarantee of GCB, Lin et al. [15] propose Tree-Structured Fourier Supports Set (TS-FSS) algorithms that combine the characteristics of submodularity and sparsity of routing trees to boost the theoretical bound.

III. Background

To formulate a multi-robot search problem, the submodularity, independence system, and balancing function are introduced.

Definition 1: (Submodularity) [6] A function $f : 2^N \rightarrow \mathbb{R}^+$ is submodular if and only if $\forall S \subseteq T \subseteq N, \forall e \in N \backslash T,\ f(S \cup \{e\}) - f(S) \geq f(T \cup \{e\}) - f(S)$.

Definition 2: (Independence system) [16] An independence system M is a pair $(E, \mathcal{I})$, where E is a finite set called the ground set and $\mathcal{I}$ is family of subsets of E called independent sets, with the following properties:

1) (non-emptiness) $\emptyset \in \mathcal{I}$;
2) (downward closure) if $A \in \mathcal{I}$ and $B \subseteq A$, then $B \in \mathcal{I}$.

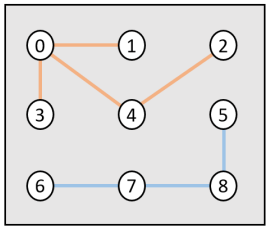
Lemma 1: (Linear combination of submodular functions) [6] If $g(S) = \sum_{i=1}^{n} \alpha_i f_i(S)$, where S is a set, $f_i : 2^S \rightarrow \mathbb{R}^+$ is a submodular function and α_i is a non-negative coefficient, then $g : 2^S \rightarrow \mathbb{R}^+$ is also a submodular function.

Corollary 1: (Submodularity of the balancing function) [17] Given a graph $\mathcal{G} = (V, E)$ and a partition set $S = \{S_1, S_2, ..., S_N\}$, the balancing function $\mathcal{B} : 2^E \rightarrow \mathbb{R}$ is a monotonically increasing submodular function and is defined as follows:

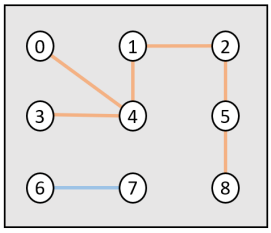
$$\mathcal{B}(S) = -\sum_{i} p_S(i) \log(p_S(i)) - N,$$

where $p_S(i) = \frac{|S_i|}{|V|}, i = \{1, ..., N\}$ and N is the number of connected components in the graph.

The balancing function encourages the clusters to have similar sizes. Fig. 2 shows the balancing function of two topologies. The balancing function shown in Fig. 2(a) has a higher value compared to that in Fig. 2(b). In other words, the higher balancing value has more balanced clustering.

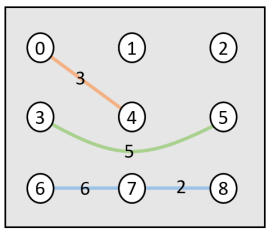


(a) $\mathcal{B}(S_A) = -1.01$

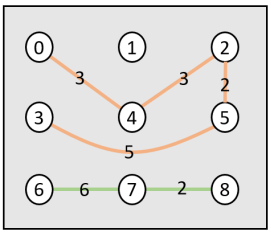


(b) $\mathcal{B}(S_B) = -1.24$

Fig. 2: Illustration of the balancing function. Two clusters are denoted by orange and blue colors. (a) $\mathcal{B}(S_A)$ is the balancing value of S_A, where $S_A = \{(0,3),(0,4),(0,1),(2,4),(6,7),(7,8),(5,8)\}$. (b) $\mathcal{B}(S_B)$ is the balancing value of S_B, where $S_B = \{(3,4),(0,4),(1,4),(1,2),(2,5),(5,8),(6,7)\}$.



(a) $S_A \in \bigcap_{\mathcal{M}\in\mathbb{M}} \mathcal{M}$



(b) $S_B \notin \bigcap_{\mathcal{M}\in\mathbb{M}} \mathcal{M}$

Fig. 3: Illustration of the intersection system. The vertices and lines represent the subgoals and the distance between two vertices, respectively. Each color line denotes a cluster. (a) S_A contains three clusters that satisfy the condition of the intersection system. (b) S_B contains two clusters that violate the condition of the intersection system due to the under-budget of the clustering and the over-budget of routing constraints.

IV. Problem Formulation

Given a complete graph $\mathcal{G} = (V, E)$, and a set of independence systems $\mathbb{M}$, the goal is to find a subset $S \subseteq E$ that maximizes environmental coverage and balances robot workloads under an intersection system. Mathematically,

$$\begin{aligned} \max_{S} \quad & F(S) \\ \text{s.t.} \quad & S \in \bigcap_{\mathcal{M}\in\mathbb{M}} \mathcal{M}, \end{aligned} \tag{1}$$

where $F : 2^E \to \mathbb{R}^+$ is the objective function. The objective function $F(S) = f(S) + \lambda\mathcal{B}(S)$ is defined as a linear combination of a coverage function and a balancing function [17]. In 3D environments, the coverage function f is defined by calculating the ratio of the number of covered voxels to the total number of environmental voxels.

The intersection system $\bigcap_{\mathcal{M}\in\mathbb{M}} \mathcal{M}$ is defined as the intersection of all elements in the set of independence systems. $\mathbb{M} = \{\mathcal{M}_R, \mathcal{M}_C\}$ is defined as a set of independence systems, where $\mathcal{M}_R = (E, \mathcal{I}_R)$ models the routing constraint and $\mathcal{M}_C = (E, \mathcal{I}_C)$ models the clustering constraint.

The routing independence system $\mathcal{M}_R$ is to limit the trajectory length of robots [18]. The independent set $\mathcal{I}_R$ is defined as

$$\mathcal{I}_R = \{S \subseteq E : c(S \cap X_i) \leq l_i, \forall i \in \{1, 2, ..., N\}\},$$

where $c : 2^E \to \mathbb{R}^+$ is a routing cost function, $X_i \subseteq S$ is the set of i^{th} cluster, l_i is the routing constraint in the cluster i, and N is the number of clusters.

The clustering independence system $\mathcal{M}_C$ is to group the ground set into clusters [17]. The independent set $\mathcal{I}_C$ is defined as

$$\mathcal{I}_C = \{S \subseteq E : N \geq k\},$$

where N is the number of clusters and k is the number of robots.

In this research, some assumptions are made. First, a set of subgoals is initially distributed evenly throughout the environment. Second, the perception of robots includes uncertainty and targets may be occluded in the environment. Third, the coverage at every subgoal can be pre-computed before the search.

To illustrate the concept, two cases are explained as follows. Fig. 3 illustrates the intersection system of $\mathcal{M}_R = (E, \mathcal{I}_R)$ and $\mathcal{M}_C = (E, \mathcal{I}_C)$. Let E be the set of edges between any vertex, $(a, b) \in E$ be an undirected edge, $\mathbb{M} = \{\mathcal{M}_R, \mathcal{M}_C\}$ be the set of independence systems, $l_i = 10$ be the routing constraint and $k = 3$ be the minimum number of clusters. In Fig. 3(a), let $S_A = \{S_1, S_2, S_3\}$, where $S_1 = \{(0,4)\}, S_2 = \{(3,5)\}$, and $S_3 = \{(6,7),(7,8)\}$. S_A satisfies the intersection system since $|S_A| \geq k$ and $c(S_A \cap S_i) \leq l_i$, where $i = \{1, 2, 3\}$. In Fig. 3(b), let $S_B = \{S_1, S_2\}$, where $S_1 = \{(0,4),(4,2),(2,5),(3,5)\}$ and $S_2 = \{(6,7),(7,8)\}$. S_B does not satisfy the intersection system since $|S_B| < k$ and $c(S_B \cap S_1) > l_i$.

Theorem 1: The proposed objective function $F(S) = f(S) + \lambda\mathcal{B}(S)$ is submodular.[1]

Proof: Let f be a coverage function and $\mathcal{B}$ be a balancing function. From Def. 1 and Cor. 1, both the coverage function and the balancing function are submodular. By Lem. 1, the linear combination of submodular functions is also submodular. Therefore, the proposed objective function F is a submodular function. ■

Theorem 2: (Routing constraint) The proposed $\mathcal{M}_R = (E, \mathcal{I}_R)$ is an independence system, where $\mathcal{I}_R = \{S \subseteq E : c(S \cap X_i) \leq l_i, \forall i \in \{1, 2, ..., N\}\}$.

Proof: To prove that $\mathcal{M}_R$ is an independence system, two properties described in Def. 2 must hold.

To prove that $\mathcal{M}_R$ satisfies the non-emptiness property, consider a set $B = \emptyset$. Let $m_i = c(B \cap X_i), \forall i \in \{1, 2, ..., N\}$ be the routing costs of the i-th cluster, and X_i be a set of cluster i. Since B is an empty set, it implies that $m_i = 0, \forall i \in \{1, 2, ..., N\}$. Thus, B belongs to the independent

[1]Both f and $\mathcal{B}$ are normalized. λ is between 0 and 1.

set $\mathcal{I}_R$.

To prove that $\mathcal{M}_R$ satisfies the downward closure property, consider a set $B_2 \subseteq E$ and $B_2 \in \mathcal{I}_R$. Let $m_i = c(B_2 \cap X_i), \forall i \in \{1, 2, ..., N\}$ be the routing costs of the i-th cluster, and X_i be a set of cluster i. Since $B_2 \in \mathcal{I}_R$, this implies that $c(B_2 \cap X_i) \leq l_i, \forall i \in \{1, 2, ..., N\}$. Therefore, $m_i \leq l_i, \forall i \in \{1, 2, ..., N\}$. Let $B_1 \subseteq B_2$ and $n_i = c(B_1 \cap X_i), \forall i \in \{1, 2, ..., N\}$. Since B_1 contains edges no more than B_2, the routing costs $n_i \leq m_i$. As a result, $n_i \leq m_i \leq l_i$. Therefore, B_1 belongs to the independent set $\mathcal{I}_R$.

Since the independent set $\mathcal{I}_R$ satisfies the non-emptiness and downward closure properties, $\mathcal{M}_R = (E, \mathcal{I}_R)$ is an independence system.

■

Theorem 3: (Clustering constraint) Let E be the edge set and $S \in E$ be the set of subsets. The clustering constraint $\mathcal{M}_C = (E, \mathcal{I}_C)$ is an independence system, where $\mathcal{I}_C = \{S \subseteq E : N \geq k\}$ and N is the number of clusters.

Proof: Since $\mathcal{M}_C$ is a matroid [17] and a matroid satisfies the properties of the independence system (Def. 2), the clustering constraint $\mathcal{M}_C$ is also an independence system.

■

V. Algorithms

The proposed algorithm is Multi-robot Search with an Intersection System (MRSIS). MRSIS solves Eq. (1) to generate k sets of trajectories for k robots.

MRSIS is presented in Alg. 1. Line 1 is to initialize the selected edge set S. Lines 2-3 define the objective function and the set of independence systems, respectively. Lines 4-8 are the initialization of the k clusters. For each robot i, an edge e_i is randomly selected that does not form connected components when added to S. Lines 9-16 are to generate trajectories for robots by iteratively selecting the edge e^* with the maximal marginal gain. To calculate the routing constraint for $\mathcal{M}_R$, an MST S' is constructed from $S \cup \{e^*\}$ by Prim's algorithm. If S' satisfies the condition of the intersection system, the edge e^* is considered the best current step; otherwise, the edge e^* will not be added to S. By iteratively adding the edge to the selected set S, MRSIS implicitly assigns a path to the robot and ensures that there are at least k clusters. The edge e^* is removed from the set E. The loop continues until there is no edge in E. The computational complexity of MRSIS is $\mathcal{O}(|E|^2 + |E||V|^2)$.

Algorithm 1 Multi-robot Search with Intersection System (MRSIS)

Input: $G = (V, E)$ (graph), $\mathbb{M}$ (set of independence systems), f (coverage function), $\mathcal{B}$ (balancing function), λ (balancing function weight), k (number of robots), l_i (routing budget)

Output: Selected edge set $S \subseteq E$

1: $S \leftarrow \emptyset$
2: $F \triangleq f + \lambda B$
3: $\mathbb{M} \triangleq \{\mathcal{M}_C, \mathcal{M}_R\}$
4: **for** $i \leftarrow 1$ to k **do**
5: $\quad e_i \leftarrow \text{RandomSample}(E, S)$
6: $\quad S \leftarrow S \cup \{e_i\}$
7: $\quad E \leftarrow E \setminus \{e_i\}$
8: **end for**
9: **while** $E \neq \emptyset$ **do**
10: $\quad e^* \leftarrow \arg\max_{e \in E} F(S \cup e) - F(S)$
11: $\quad S' \leftarrow \text{Prim}(S \cup \{e^*\})$
12: $\quad$ **if** $S' \in \bigcap_{\mathcal{M} \in \mathbb{M}} \mathcal{M}$ **then**
13: $\quad\quad S \leftarrow S \cup \{e^*\}$
14: $\quad$ **end if**
15: $\quad E \leftarrow E \setminus \{e^*\}$
16: **end while**

VI. Experiments

The proposed algorithm (MRSIS) and the benchmark algorithms (CapAM [4], PD-FAC [5], and the other 2 methods) are evaluated based on the average number of detected objects and the expected time to detection (ETTD). The ETTD is defined as follows:

$$E[TTD] = \frac{1}{Nn} \sum_{j=1}^{N} \sum_{i=1}^{n} t_i^j,$$

where N, n, and t_i^j represent the number of trials, number of objects in trial $j \in N$, and detection time of object $i \in n$ in trial $j \in N$. Besides, the search time of non-detected objects will be set to the maximal time constraint if it reaches the time constraint.

The search process is as follows: MRSIS generates a set of trajectories composed of K_i edges before the i^{th} drone takes off. As the drone reaches the subgoal, it hovers and detects targets within 3 seconds. The drone continues to traverse the edge and visits the next subgoal. The process repeats until all targets in the environment have been found, or all drones pass through all computed edges. Finally, drones land on the ground when the search task is terminated.

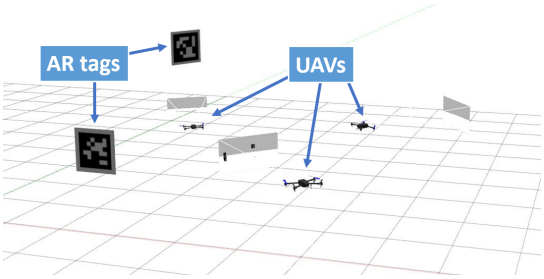


Fig. 4: Gazebo simulation with 3 UAVs and 2 AR Tags in a 3D Environment. The UAV can project a viewing frustum to observe the search space.

A. Experiment Setup

There are two experiments: the target search experiment and the balanced workloads experiment. In the target search experiment (EX1), the performance of the proposed algorithm is evaluated with simulations and a realistic complex environment. In the balanced workloads experiment (EX2), the weight λ of the balancing function is examined.

In EX1, the simulations are conducted with three robots (UAVs) on Gazebo (see Fig. 4). Various environment sizes $E = \{8, 12\}$ and numbers of targets (AR tags) $T = \{2, 4, 6\}$ in the space are evaluated. Each (e, t) pair generates 100 trials with randomly located targets, where $e \in E$ is an $e \times e \times e$ search space, and $t \in T$ is the number of targets that the robots can find in e. The goal of the robots is to find all targets that contain partial occlusion.

In the search process, each robot can take one of the actions from $\mathcal{A} = \{Move(x, y, z, \theta), Detect\}$. The $Move$ action takes the drone to the coordinate (x, y, z) with the heading θ. The $Detect$ action performs an object detection. The time constraint of the search process is 10 minutes. Once all the targets have been found or the search time exceeds the constraint, the task is terminated.

The coverage function (f) is calculated by the ratio of the number of covered voxels to the total number of environmental voxels. The routing cost (c) is measured by the Euclidean distance between two subgoals. For instance, for any two connected nodes, $v_1 = (x_1, y_1, z_1, \theta_1)$ and $v_2 = (x_2, y_2, z_2, \theta_2)$, the routing cost $c(\{(v_1, v_2)\}) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 + (\theta_1 - \theta_2)^2}$.

The real-world experiment is conducted with two drones in a 13×9 m public area on the third floor of the General Education Building at the National Central University. The map and subgoals are shown in Fig. 5(a). The space is divided into voxels whose unit size is $15 \times 15 \times 15$ cm. The goal is to find a sports ball, a chair, and a bottle in the environment, shown in Fig. 5(b). Three targets are randomly located and can be partially occluded due to the complex environment with obstacles. The searchers are the drones developed by Taiwan Drone 100 shown in Fig. 6. The drone is equipped with NVIDIA Jetson Xavier NX and two Intel RealSense cameras. The Intel RealSense T265 camera is used to localize the drone and the D435i camera is to explore the environment. The camera and drone parameters are shown in Table I.

To detect objects, the YOLOv5 [19] is adopted and run on NVIDIA Jetson Xavier NX. The uncertainty of detection is considered due to object occlusion in the environment. To successfully detect the object, the confidence rate of detection must be over a threshold of 0.6.

To further consider the importance of the balancing term, a set of weights $\lambda \in L$ is evaluated in EX2, where $L = \{0, 0.3, 0.6, 0.9, 1\}$. When $\lambda = 0$, no balanced workload is considered. When $\lambda = 1$, the workloads assigned to robots are thoroughly optimized. The experiment setup is the same as the simulation in EX1.

TABLE I: Parameters of EX1 and EX2.

Parameters	Simulation	Real-world Search
Camera Range	2 grids	$3m$
Horizontal FOV	$45°$	$69°$
Vertical FOV	$45°$	$42°$
Transitional Velocity	$1.3\ m/sec$	$0.2\ m/sec$
Angular Velocity	$45\ deg/sec$	$120\ deg/sec$

(a) (b)

Fig. 5: (a) A 13×9 m. public area on the third floor of the General Education Building at the National Central University. (b) An example of UAVs searching for targets (sports ball, bottle, and chair).

B. EX1: Targets Search

Table II and Table III show the performance of the simulated results with the routing budget $l_i = 350$, the number of robots $k = 3$, and the weight of the balancing function $\lambda = 1$ in various sizes of the environment. MRSIS is compared to state-of-the-art baselines (e.g., CapAM [4], AM-RL [20], and PD-FAC [5]).

Table II shows that the drones find more targets in a small environment ($e = 8$) than in a large one ($e = 12$). Besides, the difficulty of searching increases when the number of targets decreases in the environment. However, MRSIS achieves the highest average number of detected objects, except for $e = 8$. The advantage of the proposed approach, in contrast with all benchmarks, is that MRSIS maximizes the coverage function and the workload balancing function. In addition, the robot trajectories are minimized by applying MST.

Table III shows the ETTD of all approaches. All ap-

Fig. 6: A customized UAV developed by Taiwan Drone 100.

TABLE II: Average number of detected objects in various environment sizes. A tuple (e, t) represents a pair of space size and number of targets.

(e,t)	$(8,2)$	$(8,4)$	$(8,6)$	$(12,2)$	$(12,4)$	$(12,6)$
Random	0.86	1.43	2.25	0.12	0.34	0.52
AM-RL [20]	**1.5**	**3.15**	**4.73**	0.23	0.56	0.77
CapAM [4]	0.89	1.69	2.48	0.34	0.81	1.18
PD-FAC [5]	0.49	0.81	1.11	0.15	0.43	0.46
MRSIS	1.03	2.14	3.21	**0.51**	**0.99**	**1.51**

TABLE III: Expected time to detection (ETTD) in various environment sizes. A tuple (e, t) represents a pair of space size and number of targets. The unit is seconds.

(e,t)	$(8,2)$	$(8,4)$	$(8,6)$	$(12,2)$	$(12,4)$	$(12,6)$
Random	479.5	437.7	427.7	600	600	600
AM-RL [20]	600	600	600	564.5	539.5	518.3
CapAM [4]	462.9	417.5	376.6	544.3	499.5	463.2
PD-FAC [5]	525	450.2	438.6	556.1	497.4	485.7
MRSIS	**250.9**	**210.6**	**205**	**532**	**477.6**	**454**

proaches are limited to search within 10 minutes (600 seconds). In $e = 8$ and $e = 12$, MRSIS is the most efficient approach that can scale in a large environment. Compared to AM-RL [20], MRSIS finds fewer average targets in $e = 8$. However, most targets are found around the time constraint (600 seconds) by AM-RL [20]. Although AM-RL [20] detects a higher average number of objects compared to MRSIS, the search efficiency is compromised by the large value of ETTD.

To examine the practicality of the approaches, the algorithms (MRSIS, CapAM [4], and PD-FAC [5]) are evaluated in a real-world environment. Experiments are conducted 4 times for each approach. Since CapAM [4] and PD-FAC [5] do not consider obstacle avoidance, the trial is terminated when two drones collide. The search time of non-detected objects will be set to the maximal time constraint. The ETTD and coverage rate are shown in Table IV. MRSIS is the fastest approach among other approaches. Moreover, the coverage of MRSIS is 69%, which is the largest among these approaches. These experiments demonstrate that MRSIS outperforms benchmark algorithms.

The summaries of these experiments are as follows: The proposed algorithm (MRSIS) outperforms state-of-the-art approaches (e.g., AM-RL [20], CapAM [4], and PD-FAC [5]). MRSIS has the largest coverage rate and finds targets faster with the smallest ETTD. However, it is not guaranteed to find the most targets in the environment.

TABLE IV: The expected time to detection (ETTD) of MRSIS, CapAM [4], and PD-FAC [5].

	Mean (sec.)	Std.	Coverage Rate
MRSIS	**196**	71.8	**69%**
CapAM [4]	238	40.5	42%
PD-FAC [5]	278	102.5	33%

TABLE V: Average number of detected objects with different λ.

(e,t)	$(8,2)$	$(8,4)$	$(8,6)$
$\lambda = 0$	0.91	1.58	2.50
$\lambda = 0.3$	1.02	2.12	**3.27**
$\lambda = 0.6$	1.01	2.13	3.23
$\lambda = 1$	**1.03**	**2.14**	3.21

TABLE VI: Expected time to detection (ETTD) with different λ. The unit is seconds.

(e,t)	$(8,2)$	$(8,4)$	$(8,6)$
$\lambda = 0$	398.2	340	311.7
$\lambda = 0.3$	252.3	244.2	222.6
$\lambda = 0.6$	258.3	245.4	250.7
$\lambda = 1$	**250.9**	**210.6**	**205**

C. EX2: Effect of Balanced Workloads

Table V and Table VI show the results of various λ in simulation. In Table V, robots with a balancing function find more targets than those without one. The magnitude of λ has little impact on the average number of detected objects. In Table VI, robots with the balancing function find targets significantly faster. In addition, the ETTD is the smallest when the balanced workload is considered thoroughly.

To summarize, the assignment of balanced workloads to robots improves the search efficiency in multi-robot search scenarios.

VII. Conclusions

This research proposes a multi-robot search algorithm, MRSIS, which maximizes the objective function under an intersection system. MRSIS further considers the balanced workloads for robots to improve search efficiency. The submodularity of the objective function is proven. Besides, the routing constraint and clustering constraints have been proven to be independence systems. The experiment results show that MRSIS outperforms state-of-the-art approaches in multi-robot search problems.

The future work of this research is as follows. First, the theoretical guarantee of the proposed algorithm (MRSIS) should be provided. Second, MRSIS is based on known environments. The search for targets in unknown environments will be a potential direction. Third, MRSIS generates trajectories for robots offline. Extending the current approach to plan a trajectory online via adaptive submodularity is another direction.

ACKNOWLEDGMENT

This research was completed thanks to the financial support from Taiwan NSTC Grant 112-2221-E-008-075.

REFERENCES

[1] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.

[2] B. Zhou, H. Xu, and S. Shen, "Racer: Rapid collaborative exploration with a decentralized multi-uav system," *IEEE Transactions on Robotics*, 2023.

[3] H. Zhang and Y. Vorobeychik, "Submodular optimization with routing constraints," *Association for the Advancement of Artificial Intelligence (AAAI)*, vol. 30, no. 1, 2016.

[4] S. Paull, P. Ghassemi, and S. Chowdhury, "Learning scalable policies over graphs for multi-robot task allocation using capsule attention networks," *International Conference on Robotics and Automation (ICRA)*, pp. 8815–8822, 2022.

[5] W. Sheng, H. Guo, W.-Y. Yau, and Y. Zhou, "Pd-fac: Probability density factorized multi-agent distributional reinforcement learning for multi-robot reliable search," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8869–8876, 2022.

[6] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—i," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.

[7] M. Lauri, J. Pajarinen, J. Peters, and S. Frintrop, "Multi-sensor next-best-view planning as matroid-constrained submodular maximization," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5323–5330, 2020.

[8] X. Zhu, F. Vanegas, and F. Gonzalez, "An approach for multi-uav system navigation and target finding in cluttered environments," *International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1113–1120, 2020.

[9] A. Gunawan, H. C. Lau, and P. Vansteenwegen, "Orienteering problem: A survey of recent variants, solution approaches and applications," *European Journal of Operational Research*, vol. 255, no. 2, pp. 315–332, 2016.

[10] J. Liu and R. K. Williams, "Optimal intermittent deployment and sensor selection for environmental sensing with multi-robot teams," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1078–1083, 2018.

[11] T. K. Ralphs, L. Kopman, W. R. Pulleyblank, and L. E. Trotter, "On the capacitated vehicle routing problem," *Mathematical programming*, vol. 94, pp. 343–359, 2003.

[12] S. Jorgensen, R. H. Chen, M. B. Milam, and M. Pavone, "The matroid team surviving orienteers problem: Constrained routing of heterogeneous teams with risky traversal," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5622–5629, 2017.

[13] J. Liu and R. K. Williams, "Submodular optimization for coupled task allocation and intermittent deployment problems," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3169–3176, 2019.

[14] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Information processing letters*, vol. 70, no. 1, pp. 39–45, 1999.

[15] P.-T. Lin and K.-S. Tseng, "Improvement of submodular maximization problems with routing constraints via submodularity and fourier sparsity," *IEEE Robotics and Automation Letters*, pp. 1–8, 2023.

[16] B. Korte and D. Hausmann, "An analysis of the greedy heuristic for independence systems," *Annals of Discrete Mathematics*, vol. 2, pp. 65–74, 1978.

[17] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy-rate clustering: Cluster analysis via maximizing a submodular function subject to a matroid constraint," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 99–112, 2013.

[18] H. Zhang, R. Li, Z. Wu, and G. Sun, "Nonmonontone submodular maximization under routing constraints," *arXiv preprint arXiv:2211.17131*, 2022.

[19] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, TaoXie, J. Fang, imyhxy, Lorna, Z. Yifu, C. Wong, A. V, D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, V. Sonck, tkianai, yxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, and M. Jain, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation," Nov. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7347926

[20] W. Kool, H. van Hoof, and M. Welling, "Attention, learn to solve routing problems!" *International Conference on Learning Representations*, 2019.