KC-TSS: An Algorithm for Heterogeneous Robot Teams Performing Resilient Target Search

Minkyu Kim¹,², Ryan Gupta¹,³, and Luis Sentis¹,³

Abstract—This paper proposes KC-TSS: K-Clustered-Traveling Salesman Based Search, a failure resilient path planning algorithm for heterogeneous robot teams performing target search in human environments. We separate the sample path generation problem into Heterogeneous Clustering and multiple Traveling salesman Problems. This allows us to provide high quality candidate paths (i.e. minimal backtracking, overlap) to an Information-Theoretic utility function for each agent. First, we generate waypoint candidates from map knowledge and a target prediction model. All of these candidates are clustered according to the number of agents and their ability to cover space, or coverage competency. Each agent solves a Traveling salesman problem (TSP) instance over their assigned cluster and then candidates are fed to a utility function for path selection. We perform extensive Gazebo simulations and preliminary deployment of real robots in indoor search and simulated rescue scenarios with static targets. We compare our proposed method against a state-of-the-art algorithm and show that ours is able to outperform it in mission time. Our method provides resilience in the event of single or multi teammate failure by recomputing global team plans online.

I. INTRODUCTION

This study attempts to solve the problem of target search with multiple heterogeneous robots by generating informative global paths and executing search plans. We interpret the target search problem as a variant of Coverage Path Planning (CPP) or exploration in which a team must additionally detect a missing or injured person or object in situations where mission speed is critical. In traditional CPP a robot or a team of robots must find the optimal path such that the sensor footprint, or FoV, passes over the entire region, referred to as the search polygon. Exploration strategy is typically to find informative next view-point or path using sampling based approaches. We take heterogeneous to mean different sizes of Field of View (FoV) and movement speeds. The target may be considered as static and may be given with some prior target knowledge (e.g. a search and rescue mission searching near buildings while looking for people) and we assume static map information is known a priori. When the target is static the proposed search algorithm simplifies to a CPP algorithm due to the uniform target prediction model over unseen space in the basic problem setting.

CPP, exploration and target search are receiving significant attention from the robotics community due to the relevance in many important real-world scenarios including map building [1], [2], [3], [4], cleaning/covering indoor areas [5], [6], security, surveillance and information gathering [7], [8], [9], [10], reconnaissance or search and rescue [11], [12], [13].

The goal of the presented algorithm is to generate high quality search plans for robot teams that attempt to utilize each team member to its full capacity. We define a metric, coverage competency, which captures each member of the team’s ability to effectively cover ground during search. Our algorithm uses coverage competency to assign a search region to each agent through heterogeneous clustering (i.e. the spread of the waypoints reflects that agent’s FoV size and movement speed). We also provide a new metric, Waypoint Allocation Factor (WAF), for measuring how evenly the waypoints were divided amongst agents based on their coverage competency score.

We demonstrate that our algorithm has proportional complexity in n and validate with n=50 agents. Further, we demonstrate that our algorithm is robust to failure of team members. If one or more agents fail, the remaining member(s) will complete the search. Because the server has access to robot plans over the entire search time via the TSP solution and current robot state, when one or more teammates fail, a re-planning step is invoked and nearby agents are assigned the failed members waypoints, as demonstrated in Fig. 2.

Fig. 1. Sampling paths along the TSP trajectory to calculate the Expected Information Gain. The three agents each have two candidate paths, A and B. Shaded rectangles denote the FoV of each agent at points along the path.

In the static target case, our method is equivalent to CPP (uniform target prediction) and we show it to outperform the state-of-the-art CPP method [7] in mission
problem and describe our methods. In section IV we exper-
related works. In section III we formally introduce our
areas. They propose the robot team conducts auction and
generating path plans with up to 150 agents in non-convex
frontier and sampling based exploration strategies. [18] focuses on
mission speed with MA Vs by taking a hybrid approach to
gathering missions like signal monitoring. [17], which typically give global guarantees with respect t o
mixed integer linear programming.

This paper is organized as follows. Section II discusses
related works. In section III we formally introduce our
problem and describe our methods. In section IV we exper-
mentally validate our methods in simulation finally section
V concludes our work.

II. BACKGROUND AND RELATED WORK

For decades CPP has received great attention for its
relevance in navigation tasks. Early solutions offer an offline
planner given a static map. A traditional method is to de-
compose the coverage region into cells using Boutroushedon
decomposition [14]. We employ a similar cell decomposition
to generate the set of waypoints for complete coverage. A
common strategy in this form of the problem is to break
down the map and perform simple back-and-forth behaviors.
One alternative to such behavior is to generate the set of
points to visit and solve a Traveling Salesman Problem [6],
[11]. [15] instead find the optimal order of visitation using
mixed integer linear programming.

There are also a number of geometric approaches [5],
[17], which typically give global guarantees with respect to
distance traveled during execution. However, such methods
often make assumptions about perfect omni-directional sens-
ing. Traditional methods for path planning are insufficient,
however, when the goal is to operate in an online fashion to
handle robot failure or dynamic environments.

Sampling based approaches are popular in mapping and
exploration [2], [3], [4], [9], [18]. Such methods account for
sensor noise and environment complexity. Early work used
frontiers [19] to explore spaces. Frontiers are the boundaries
between explored and unexplored regions of the search
polygon. The critical factor in information-theoretic based
strategies is how to generate subsequent observation paths;
that is, to provide quality candidates to the utility function.
Early work [2] uses a laser range scanner to map a previously
unknown small environment. In [3], [4] they leverage a
utility function based on Cauchy-Schwartz quadratic mutual
information to more efficiently generate plans to map 3D
spaces. [9] proposes three planning structures for information
gathering missions like signal monitoring. [18] focuses on
mission speed with MAVs by taking a hybrid approach to
frontier and sampling based exploration strategies.

[7] addresses the problem of scalable CPP by efficiently
generating path plans with up to 150 agents in non-convex
areas. They propose the robot team conducts auction and
conflict resolution steps to determine the region of space
they will cover. We benchmark our method against this
path planner, which is state-of-the-art in terms of coverage
time. Our method features agent autonomy for search execution, a target prediction model, and automatic re-
planning; however, when the target is static the path planning
algorithm is equivalently a CPP problem, and therefore we
use [7] as a state-of-the-art comparison in terms of mission
time and computation time (see Table I).

There are an abundance of real-time algorithms for multi-
robot teams in coverage and exploration tasks. Groups
have emphasized resilience to robot failure [20], [21], [22],
management of energy [23], [24], [25] or communications
[26], [27], [28] constraints, and heterogeneous teaming [29],
[30]. [24] also considers an information gathering mission,
however their future work lists increasing to n agent systems.
[25] performs search and exploration with UAVs, which
are limited in both communication and battery life. They
use a state machine to help team members decide between
exploring, meeting, sacrificing and relaying. They leverage
a frontier based method. In [26] they consider a model of
communication strength between the agents and a central
control PC and cleverly use serial connection of the robots
to maximize their exploration area. Instead of constraining
the team to constant connectivity, [28] proposes a method
of periodic communications at fixed intervals to update the
full team. In [13] an algorithm for solving the Multi-robot
Efficient Search Path Planning (MESPP) problem for find-
a non-adversarial moving target. This method provides
theoretical bounds on search performance however it only
scales up to five agents and does not provide resilience to
robot failure.

Our method, KC-TSS, builds on previous work [30] where
the team was not robust to agent failure because it did not
provide complete search plans at any given moment. As
a result of both this fact and the greedy planner, some of
the generated paths were overlapping and lacked efficiency.
The additional clustering and TSP steps allow us to provide
better candidate paths to the utility function and improve
team cooperation.

III. METHODS

In this section we define a cooperative multi-agent target
search algorithm for solving the problem setting described
above. Given a search map (entropy map), a search poly-
genon, the number of robots and initial robot position, this
method first uses cell decomposition to generate a set of
waypoints, which, if visited, provide complete map coverage
over the unknown (high entropy) regions. Then we perform
a weighted clustering over all points and assign sub-regions
(clusters) to each agent based on their coverage competency.
The next step solves instances of the single agent TSP in
parallel to maintain the computational efficiency needed for
generating global paths online if re-planning is necessary.
The result is two high quality candidate paths for each agent,
that is, along both directions of the TSP solution (see Fig.
I). The final step is for these candidates for each agent to be
Algorithm 1 Multi-Agent Search()

Input: \( M, \mu, \Sigma \)
Entrophy Map, Robot Pose, Coverage Competency

Output: \( P^* = \{p_1^*, p_2^*, \cdots, p_N^*\} \) (A set of paths)

\[
W \leftarrow \text{ExtractionUnknownRegions}() \\
C = \text{HeterogenousClustering}(W) \\
\text{for } n \leftarrow 1 \text{ to } N \text{ do } \quad \triangleright \text{for each cluster in } C \\
\quad r_n = \text{TravelingSalesmanProblem}(c_n) \\
\text{end for} \\
\text{for } n \leftarrow 1 \text{ to } N \text{ do } \quad \triangleright \text{for each agent} \\
\quad \hat{p}_n = \text{SamplingPaths}(r_n) \\
\quad \text{for } k \leftarrow 1 \text{ to } |\hat{p}_n| \text{ do } \quad \triangleright \text{Equation (7)} \\
\quad \quad U(p_k) = IG(p_k) - c(p_k) \\
\quad \text{end for} \\
\quad p^* = p_k \leftarrow \arg\max U(p_k) \quad \triangleright \text{Get the best path} \\
\quad P^*.\text{append}(p^*) \\
\text{end for} \\
\text{return } p^*
\]

considered by the IG based utility function (see Eq. 6). This IG is computed as entropy reduction along the path, where entropy is modeled by the target prediction model and unseen regions. Because of the computational efficiency and global knowledge of this algorithm, re-planning is performed in event of agent failure or if some agents finish their assigned waypoints.

A. Overall Framework

Our search, presented in Algorithm 1
1) Update search map using Bayesian filtering,
2) Use Algorithm 2 to Generate waypoints from map,
3) Heterogeneous Clustering using coverage competency, See Algorithm 3
4) Solve Traveling Salesman Problem for each agent,
5) Select optimal path using information-theoretic approach \\
6) If Re-plan conditions are met, repeat steps 2) - 5).

B. Target Estimation: Bayesian Filtering

We use Bayesian Inference to recursively estimate target state \( x \) through sequential observations \( y \). Bayesian inference is commonly used to estimate target state in a probabilistic manner. This inference model aims to predict the posterior distribution of target position at time \( k \), namely, \( p(x_k) \). Bayesian filtering uses a prediction stage and a correction stage with incoming sensing information. Assuming that the prior distribution \( p(x_{k-1}) \) is available at time \( k-1 \), the prediction stage attempts to estimate \( P(x_k|y_1^{kn}) \) from previous observations as follows.

\[
p(x_k|y_1^{kn}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_1^{kn})dx_{k-1}, \quad (1)
\]

where \( p(x_k|x_{k-1}) \) is the target’s motion model based on a first order Markov process. Then, when the measurement \( y_k \) is available, the estimated state can be updated as

\[
p(x_k|y_1^{kn}) = \frac{p(y_k^{1:n}|x_k)p(x_k|y_1^{kn-1})}{p(y_k^{1:n}|y_1^{kn-1})} \quad (2)
\]

where \( p(y_k^{1:n}|x_k) = \int p(y_k^{1:n}|x_k)p(x_k|y_1^{kn-1})dx_k \) and \( p(y_k^{1:n}|x_k) \) is a sensing model for multi agent system, which can also be decomposed to each agent’s sensing model \( p(y_k|x) \). For the correction stage, the measurement of all agents are used to modify the prior estimate, leading to the target belief. If a static target is assumed the target prediction can be described as \( p(x_{k}|x_{k-1}) = N(x_{k-1}; x_k, \Sigma) \), only containing a noise term with the previous target state. If it is a target motion model assumed to have some constant velocity, we can represent the model as \( p(x_{k}|x_{k-1}) = N(x_{k-1}; x_k + V\Delta, \Sigma) \).

C. Target Prediction

The action plan our algorithm generates incorporates information from the target prediction model. The prediction model is made based on various prior knowledge or experience. For example, in a rescue mission, it might be useful to take advantage of the fact that people are more likely to be near collapsed buildings.In this paper, those are called context \( c \) and can be used to estimate a target location. A Gaussian Mixture Model (GMM) can be constructed using a finite number of contexts at time \( k \) using the following equation

\[
p(x_k|c_k) = \sum_{i=1}^{N} p(x_k|c_k) \pi_i G(x; \mu_i, \Sigma_i)
\]

where \( \pi_i, \mu_i, \Sigma_i \) are a mixing coefficient, mean vector, and covariance for \( i \)-th distribution, respectively. We use a particle filter to implement this prediction model.

D. Search Map

The search map is updated using sensor data of each robot at each instance. The exploration begins under the assumption that the target exists in the search region. We use a 2D occupancy grid map representation in which the search region is discretized into cells. More details of the search map cells are presented in [30]. A cell’s occupancy status is used to measure the uncertainty of the target over the total search space (i.e. the entropy map). Entropy is defined here as

\[
H(M_t) = -\sum_{i=1}^{N} (m_t^i \log(m_t^i) + (1 - m_t^i) \log(1 - m_t^i))
\]

where \( m_t^i \) is the occupancy variable at time step \( t \) and \( N \) denotes the total number of cells. The search map is also maintained in this way. Furthermore, it can be filtered such that the degree of entropy increases over time in previously searched regions to account for a target prediction model.
Algorithm 2 ExtractionUnknownRegions()

Input: $M_{pos}$ $\triangleright$ (Entropy Map, Current Pose)
Output: $\mathcal{W} = \{w_1, w_2, \cdots, w_n\}$ NewUnknownSet

- $queue_m \leftarrow \emptyset$
- $\mathcal{W} \leftarrow \emptyset$
- Initialize $Visited[M_{m=0.5}] = False$
- $c_0 \approx M_{m=0.5}$ take the unknown cell
- $\text{Enqueue}(\text{queue}_m, c_0)$
- $Visited[c_0] = True$

while $\text{queue}_m$ is not empty() do
  $c_0 \leftarrow \text{DEQUEUE}(\text{queue}_m)$
  for each $n_c$ $\leftarrow \text{neighborhood}(c_0)$ do
    if $\text{IsNewUnknown}(n_c)$ and $\text{visited}[\text{cell}] == False$
      $w, \text{Visited}, l_c \leftarrow \text{buildNewUnknowns}(n_c)$
      $\mathcal{W}.\text{append}(w)$
      $\text{Enqueue}(\text{queue}_m, l_c)$
    else if $Visited[n_c] == False$
      $\text{Enqueue}(\text{queue}_m, n_c)$
      $Visited[n_c] = True$
  end if
  end for
end while

return $\mathcal{W}$(NewUnknownSet)

Algorithm 3 HeterogeneousClustering()

Input: $\mathcal{W}, \mu_1, \mu_2, \cdots, \mu_n$, $SR_{1:n}$ (normalized) (Waypoints, Centroids(Robot Poses), Sensing Capabilities)

Output: $\mathcal{C} = \{C_1, C_2, \cdots, C_n\}$

- $cost = 0$
- Initialize $\mu$

repeat
  $\hat{C}_i = \{w_j : \eta_j d(w_j, \mu_i) \leq \eta_j d(w_j, \mu_h) \text{ for all } h = 1, \ldots, n\}$
  $\triangleright$ assign all datapoints to the nearest cluster
  $\mu = \frac{1}{\Delta \text{cost}} \sum_{w_j \in \hat{C}_i} w_j$
  $\triangleright$ update centroids if required
  $\Delta \text{cost} = ||\mu - \text{cost}||$
  if $\Delta \text{cost} < \epsilon$
    $\mathcal{C} \leftarrow \hat{C}$
    break
  end if
  $\text{end if}$
until $\text{MAXLOOP}$
return $\mathcal{C}$

\[
\min_{c^* \in \cdots \in c^{K'}} \sum_{c_j \in \mathcal{C}} \sum_{j=1}^{K} ||\eta_j d_j(w, c_j)||
\]  

where $\eta_j d_j(\cdot, \cdot)$ is the weighted distance, which reflects the $j$-th robot’s sensing capability using the normalized coefficient $\eta_j$. We define coverage competency $\eta_j$ as $\frac{SR_{\min} \cdot \eta_j}{SR_j}$ where $SR_j$ is the sensing capacity of $j$-th agent, which is the product of a moving speed and sensing range of agent $j$. Because we normalize this coefficient with $SR_{\min}$, the minimum value of sensing capacity among all agents, we can use it to consider relative proximity between centroids and points. In this way we ensure that robots which are fast or have a large FoV are allocated more waypoints than robots which are not.

Furthermore, this formulation has the advantage of being flexible in varying situations. When the robots are evenly distributed in the search region, computed waypoints will easily be distributed among the agents. However, some situations arise in which robots start next to one another. If two agents with competency gaps begin next to each other, then all of the nearby points will be assigned to the more competent teammate. To avoid this, we compute a new centroid when any agent is assigned fewer than some minimum defined number of points. If, on the other hand, two agents of equal skill begin next to each other we compute the distance to other agents and determine which member is given the new centroid and in addition those nearby agents are given a new centroid. As a result of clustering, we receive the waypoints assigned to each agent (centroid) in $\mathcal{C}$.

G. Traveling Salesman Problem

After constructing and assigning a cluster to each robot we solve an instance of the Traveling Salesman Problem (TSP) separately for each agent in parallel. In this study we adopted the Genetic Algorithm for solving TSP [31]. In this setting the start point is the cluster’s centroid (i.e. the robot’s
The result is the optimal route along the cluster. While this generates a set of \(k\) optimal paths when considered independently, the solution does not consider the other agents. To overcome this lack of consideration of the team’s combined effort, we generate a set of sample paths for each agent then use the information-theoretic framework using the aforementioned constraints. We exploit the utility function which maximizes the acquisition of information for the full team in a limited time while penalizing traveling costs. In this way we have provided higher quality samples to the utility function when compared to traditional or frontier-based methods.

**H. Path Selection**

In this stage, our framework determines which direction the robots will travel along the given route from TSP (See Fig. 1). Given the routes from TSP, candidate paths are sampled using several points which are close to robot positions. An A* planner is used to generate obstacle-free paths and all paths are re-parameterized with respect to agent’s moving speed. Based on the number of sampled points along a path, we compute the Information Gain, denoted \(IG(s)\), with the following equation:

\[
IG(s) \approx \sum_{i=1}^{N_c} H(FOV(s^i))
\]

\[
= \sum_{i=1}^{N_c} \left[ \sum_{j=1}^{N_s} \left[ m_{i,j}\log(m_{i,j}) + (1 - m_{i,j})\log(1 - m_{i,j}) \right] \right]
\]

where \(s^i\) denotes the \(i\)-th sampled point and \(m\) is the occupancy probability, while \(N_c\) and \(N_s\) are the number of sampling points and the number of cells in the FOV given the sampled points, respectively. Thus, The \(IG(s)\) is calculated by summing over the FOV regions defined by sampled points through the path. The overall expected utility, \(\mathbb{E}[\bar{U}]\), is then computed for the full team as

\[
\mathbb{E}[\bar{U}(x_1, y^1, \cdots, y^n)] = \sum_{i} (IG(s_i) - c(s_i)) \quad (7)
\]

where \(c(s_i)\) denotes the traveling cost to move along the path \(s_i\).

**I. Re-planning**

In the event of robot failure or if some of the agents finish covering their sub-region without finding the target, it is desirable to re-plan for the team. When a preset percentage of the team covers their region we generate a new set of waypoints over the full unexplored search map and perform the full process again. Similarly, if a robot loses communication with the server or fails to move for an extended period of time, we perform the same process of resetting to generating waypoints (step 2) in section III-A.

We explicitly define re-planning conditions.

- Failure (stop operating or lost communication)
- Another prediction prediction model applied
- All way points assigned are visited by the corresponding agent

These re-planning conditions allow our team to be adaptable to changes in the environment including robot failure. One re-planning scenario is shown in Fig. 2.

**J. WAF**

We introduce a new metric called \(WAF\), *Waypoint Allocation Factor* to evaluate the contribution of the task according to the coverage competency. In order to compute how evenly the area coverage is allocated, we take the total swept area of each agent and divide by the coverage competency. Specifically, we apply the following equation,

\[
WAF = std \left( \frac{\lambda A(|r_i|)}{\eta_i} \right) \quad (8)
\]

where \(A(|r_i|)\) denotes total swept area for agent \(i\), \(\eta_i\) is the coverage competency, and \(\lambda\) means a normalizing constant. A value close to zero indicates that the waypoints were evenly distributed among the team after competency considerations.


**K. Computational Complexity**

Achieving computational efficiency is critical for achieving online planning as the number of agents increase. We analyze here the computational efficiency of the proposed algorithm using the following parameters; Number of cells \( N_g \) (map size / resolution), the number of trajectories to sample \( N_t \), the number of agents \( k \), and other parameters for other algorithm.

In detail, the complexity of the waypoint generation algorithm 2 is \( O(N_g) \). The clustering has a time complexity of \( O(i kmd) \) where \( i \) denotes the fixed number of iterations (max iteration), \( k \) is the number of clusters which is equal to the number of agents, number of \( m \) data and \( d \) dimension of the data. It can be computed as \( \frac{N_g}{ENF\_FOV} \) and represents the worst case of decomposition (i.e. that the search region is totally unknown and divided by the product of number of agents and unit cell size of the FOV). The solution of the TSP with the genetic algorithm is of order \( O(jn^2n^2) \) where \( j \) is number of outer iterations of genetic algorithm and \( n_0 \) is the initial size of population, and \( n \) is number of locations. Sampling based path selection algorithm takes \( O(N_t \log N_t) \) complexity. Overall, the complexity of our algorithm is \( O(N_g + 2ik \frac{N_g}{ENF\_FOV} + kjn_0^2N_t \log N_t) \).

IV. Experiments and Results

A. Numerical Simulation Results

We present python-based simulation results of our proposed approach to demonstrate its scalability. Agent state is represented as \( s = [x, y, \theta] \) and has two control inputs \( u = [v, \omega] \) as per the equations of motion for a non-holonomic mobile agent. Each agent is equipped with a ray sensor which has square-type of FoV with limited range. It is assumed that the simulation environment (search region) and all the static obstacles have a rectangular shape and obstacles can not be known in advance. To achieve robust collision avoidance, we use dynamic window approach [32] to generate each agent’s control input. We tested different initial conditions with varying number of agents and we show some example scenarios in Fig 3.

![Fig. 3. Simulation results with different number of agents in different size search regions. (a) 3 agent case (40×40) (b) 10 agents (80×80) (c) 50 agents (200×200).](image)

B. Comparative Results

We benchmark our method against a state-of-the-art algorithm with the results shown in Table I. KC-TSS is shown to outperform the offline algorithm SCoPP [7] when using the same search setting. That is the simulation trials were performed in the python-based numerical simulation in a region of the same size using the same number of agents which are given equal skill. We generate this data using 13 agents in the same size rectangular search polygon as presented by [7] and we generate this data from 10 comparative trials.

C. Effectiveness of Coverage Competency

In simulation we tested six cases. Because of the random distribution of robot starting place, it will not approach
Age nt 1
Age nt 2
Age nt 3
Target Location
Target Object
Local Costmap
Clusters

Fig. 5. (a) Experimental validation for a three robot search in the Aerospace Building 4th floor. In the top left figure, the yellow regions are those which have been explored, while the white areas are regions of uncertainty. The target location is described as a red box and is unknown to the robots. The red, green and blue markers represent the clusters for each agent. (b) The completion of the experimental validation with Agent 1 converging to the target of interest. The object recognition is performed using the well-known YOLO algorithm to detect people (shown in the cyan box).

Fig. 6. Waypoint Allocation Factor versus number of agents

zero because we cannot perfectly account for coverage skill. However, our results suggest that this value is safely maintained below 0.4 in all scenarios. It is natural that WAF increases with the number of agents due to the random initial conditions. The data from these trials is in Fig. 6.

D. Resilience to Failure

One significant advantage of our framework is that it can quickly re-plan navigation behavior when one or more team members fail via loss of communication or navigational error (e.g. a computer loses internet connection, a legged robots falls). When faced with one of these events, we regenerate the set of waypoints over the unexplored region, cluster and allocate them, and then continue the method as before. Fig. 2 demonstrates this resilience to failure.

E. Gazebo-ROS Simulation

We validate our framework’s ability to transfer to robotic systems in human environments with high fidelity Gazebo simulations. The first is a 10mx20m simulation of the Anna Hiss Gymnasium apartment, the second is a 40mx50m region simulating the outside of a home environment (see Fig. 4 upper right corner) and the final is a 100mx100m town that has been struck by natural disaster. We validate the static target search capability in all three environments. In the town map, the agents search for an injured person and we use this information to guide the target prediction model by assuming the injured person will be near a building as opposed to in open space. We also demonstrate the resilience of the search method by performing re-planning under agent failure. This paper’s accompanying video features the described scenarios.

To validate the inclusion of coverage competency, we present II to compare mission time for the agents when coverage competency is used versus when it is not used. Additionally, for the trials using coverage competency we include the WAF score. This table uses teams of $n$ heterogeneous robots and is computed over 10 trials on each map with varying initial conditions. In all cases, it was confirmed that the search time was reduced and the WAF value was also low. There is a trend that the WAF increases slightly as the size of the map increases. The results clearly demonstrates that the addition of coverage competency impacts the search speed of the team and that the search area was more properly distributed.

F. Real Robot Experiment

We implement search for a static target with a heterogeneous three robot team. This preliminary experiment demonstrates that our proposed method is viable in real world environments, however, execution of the search required human intervention due to localization errors. The team is comprised of two Unitree A1 quadrupeds with different sen-

| Map (## robots) | solo CC | WAF | w CC (proposed) | WAF |
|----------------|--------|-----|-----------------|-----|
| Apartment (2)  | 123.8 ± 13.2 | 0.23 | **83.8 ± 22.4** | 0.07 |
| Home Outdoor (3)| 72.8 ± 8.5  | 0.31 | **58.1 ± 12.4** | 0.12 |
| Disaster (4)   | 252.2 ± 21.2 | 0.37 | **189.8 ± 29.6** | 0.18 |

TABLE II
SEARCH TIME (S) WITH AND WITHOUT COVERAGE COMPETENCY
sor suites and a Toyota HSR. One A1 quadruped is equipped with a Velodyne VLP-16 3D Lidar and a Realsense D435. The other is equipped with a RPLidar A3 2D Lidar and a Velodyne VLP-16 3D Lidar and a RealSense D435. Both quadrupeds have Intel NUC Mini PCs onboard which communicate with the robot hardware. The HSR is equipped with Hokuyo 2D Lidar, RGB-D camera sensing and has an on board Jetson TK1. The central search server is run on a remote laptop and we use Robotfleet [33] for efficient communication between the server and agents. The entire framework is implemented in ROS Melodic. At initial planning or a re-planning step, the search server generates a set of n paths and then sends those to the agents (see Fig. 4).

The search experiment is performed in the Aerospace Engineering building at the University of Texas at Austin while looking for a static volleyball and the total mission time is one minute 14 seconds. The experiment is depicted in Fig. 5.

V. CONCLUDING REMARK

This paper addresses online search for a general heterogeneous multi agent systems. The mission completion time of our method is compared with a state-of-the-art algorithm and shown to outperform it. We further validate our algorithm with extensive simulation results in Gazebo using ROS. Finally, we demonstrate the efficacy of this proposed method in a real robot experiment. Overall results validate the effectiveness and robustness of the proposed algorithm. Several future works remain, however, current efforts are towards including a target motion model for dynamic targets and performing continuous coverage.

REFERENCES

[1] C. Cao, H. Zhu, H. Choset, and J. Zhang, “Tare: A hierarchical framework for efficiently exploring complex 3d environments,” in Robotics: Science and Systems Conference (RSS), Virtual, 2021.
[2] F. Amigoni and V. Caglioti, “An information-based exploration strategy for environment mapping with mobile robots,” Robotics and Autonomous Systems, vol. 58, no. 5, pp. 684–699, 2010.
[3] B. Charrow, G. Kahn, S. Patil, S. Liu, K. Goldberg, P. Abbeel, N. Michael, and V. Kumar, “Information-theoretic planning with trajectory optimization for dense 3d mapping,” in Robotics: Science and Systems, vol. 11, 2015.
[4] B. Charrow, S. Liu, V. Kumar, and N. Michael, “Information-theoretic mapping using cauchy-schwarz quadratic mutual information,” in 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015, pp. 4791–4798.
[5] P. R. Tokekar, “Placement and motion planning algorithms for robotic sensing systems,” Ph.D. dissertation, University of Minnesota, 2014.
[6] R. Bähnemann, N. Lawrance, J. J. Chuang, M. Pantic, R. Siegwart, and J. Nieto, “Revisiting boustrophedon coverage path planning as a generalized traveling salesman problem,” in Field and Service Robotics. Springer, 2021, pp. 277–290.
[7] L. Collins, P. Ghassemi, E. T. Esfahani, D. Doermann, K. Dantu, and S. Chowdhury, “Scalable coverage path planning of multi-robot teams for monitoring non-convex areas,” arXiv preprint arXiv:2103.14709, 2021.
[8] S. McCammon and G. A. Hollinger, “Topological path planning for autonomous information gathering,” Autonomous Robots, vol. 45, no. 6, pp. 821–842, 2021.
[9] G. A. Hollinger and G. S. Sukhatme, “Sampling-based robotic information gathering algorithms,” The International Journal of Robotics Research, vol. 33, no. 9, pp. 1271–1287, 2014.
[10] V. Cavinato, T. Epenberger, D. Youakim, R. Siegwart, and R. Dubé, “Dynamic-aware autonomous exploration in populated environments,” arXiv preprint arXiv:2104.02696, 2021.
[11] R. Pėniūka, M. Saska, C. Reymann, and S. Lacroix, “Reactive dubs traveling salesman problem for replanning of information gathering by uavs,” in 2017 European Conference on Mobile Robots (ECMR). IEEE, 2017, pp. 1–6.
[12] S. Kohlbrecher, F. Kunz, D. Koert, C. Rose, P. Manns, K. Daun, J. Schubert, A. Stumpf, and O. v. Stryk, “Towards highly reliable autonomy for urban search and rescue robots,” in Robot Soccer World Cup. Springer, 2014, pp. 118–129.
[13] G. Hollinger, S. Singh, J. Djugash, and A. Kehagias, “Efficient multi-robot search for a moving target,” The International Journal of Robotics Research, vol. 28, no. 2, pp. 201–219, 2009.
[14] H. Choset, “Coverage of known spaces: The boustrophedon cellular decomposition,” Autonomous Robots, vol. 9, no. 3, pp. 247–253, 2000.
[15] E. I. Grotli and T. A. Johansen, “Path planning for uavs under communication constraints using splat! and mlp,” pp. 265–282, 2012.
[16] T. I. F E. Erik J. Forsmo, Esten I. Grotli and T. A. Johansen, “Optimal search mission with unmanned aerial vehicles using mixed integer linear programming,” 2013.
[17] A. P. Premkumar, K. Yu, and P. Tokekar, “A geometric approach for multi-robot exploration in orthogonal polygons,” in Algorithmic Foundations of Robotics XII. Springer, 2020, pp. 896–911.
[18] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, and S. Leutenegger, “Fast frontier-based information-driven autonomous exploration with an mav,” arXiv preprint arXiv:2002.04440, 2020.
[19] B. Yamauchi, “A frontier-based approach for autonomous exploration,” in Computational Intelligence in Robotics and Automation, 1997. CIRA’97., Proceedings. 1997 IEEE International Symposium on. IEEE, 1997, pp. 146–151.
[20] L. Zhou, V. Tzoumas, G. J. Pappas, and P. Tokekar, “Resilient active target tracking with multiple robots,” IEEE Robotics and Automation Letters, vol. 4, no. 1, pp. 129–136, 2018.
[21] I. E. Rabban and P. Tokekar, “Improved resilient coverage maximization with multiple robots,” arXiv e-prints, pp. arXiv–2007, 2020.
[22] M. Sedighpour, M. Yousefikhoshbakht, and N. Mahmoudi Darani, “An information driven path planning problem with multiple robots,” IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 3894–3901, 2021.
[23] K. Yu, J. M. O’Kane, and P. Tokekar, “Coverage of an environment using energy-constrained unmanned aerial vehicles,” in 2019 international conference on robotics and automation (ICRA). IEEE, 2019, pp. 3259–3265.
[24] X. Cai, B. Schlotfeldt, K. Khosoussi, N. Atanasov, G. J. Pappus, and J. P. How, “Non-monotone energy-aware information gathering for heterogeneous robot teams,” arXiv preprint arXiv:2101.11093, 2021.
[25] K. Cesare, R. Skeee, S.-H. Yoo, Y. Zhang, and G. Hollinger, “Multi-robot exploration with limited communication and battery,” in 2015 IEEE international conference on robotics and automation (ICRA). IEEE, 2015, pp. 2230–2235.
[26] B. Woosley, P. Dasgupta, J. G. Rogers, and J. Twigg, “Multi-robot information driven path planning under communication constraints,” Autonomous Robots, vol. 44, no. 5, pp. 721–737, 2020.
[27] G. Shi, I. E. Rabban, L. Zhou, and P. Tokekar, “Communication-aware multi-robot coordination with submodular maximization,” arXiv preprint arXiv:2011.01476, 2020.
[28] G. A. Hollinger and S. Singh, “Multirobot coordination with periodic connectivity: Theory and experiments,” IEEE Transactions on Robotics, vol. 28, no. 4, pp. 967–973, 2012.
[29] T. Sakamoto, S. Bonardi, and T. Kubota, “A routing framework for heterogeneous multi-robot teams in exploration tasks,” IEEE Robotics and Automation Letters, vol. 5, no. 4, pp. 6662–6669, 2020.
[30] M. Kim, R. Gupta, and L. Sentis, “Information-theoretic based target search mission with unmanned aerial vehicles using mixed integer linear programming,” 2013.