

Simulations of Hierarchical Belief Sharing for Multi-Agent Coverage with Heterogeneous Targets

Andrew Meighan*, Jonathan Ponniah† and Or D. Dantsker‡

Indiana University, Bloomington, IN 47408

A new way of sharing beliefs for multi-agent coverage is proposed. The approach, Hierarchical Belief Sharing (HBS), is based on a multi-level clustering strategy in which agents dynamically form clusters (and clusters of clusters) that belong to a single hierarchical tree. Clusters compress and forward local observations through this tree, to their members and sibling clusters. This approach has the advantage of both reducing the total throughput required to maximize coverage, and simplifying the control policy by exploiting local-interactive structure in the global state space. Simulations of HBS under multi-agent coverage scenarios with heterogeneous targets are performed to test this hypothesis using potential-field motion-primitives. It is shown that HBS compares favorably to scenarios where agents observe the perfect global state, and even outperforms these scenarios for certain target distributions. The relevance of HBS to motion-planning via reinforcement learning is discussed.

Nomenclature

$B(t)$	=	belief matrix at iteration t
I	=	number of rows in the belief matrix
J	=	number of columns in the belief matrix
$b(t)$	=	belief matrix entry for cell resolution at iteration t
$a(t)$	=	belief matrix entry for age of cell observation at iteration t
Φ	=	generic function for synthesis of two belief matrices
Φ_A	=	function for synthesis of two belief matrices that prioritizes age
Φ_R	=	function for synthesis of two belief matrices that prioritizes resolution
c	=	compression constant
v	=	velocity
$\mathbf{F}_1(t)$	=	cumulative attractive force generated by the targets resolutions at iteration t
$\mathbf{F}_2(t)$	=	cumulative attractive force generated by targets freshness at iteration t
$\mathbf{F}_3(t)$	=	cumulative repelling force generated by other agents at iteration t
$\mathbf{F}_4(t)$	=	cumulative repelling force generated by grid boundary at iteration t
$\mathbf{F}(t)$	=	total force generated by four component forces at iteration t
$Q_v(t)$	=	number of targets visited at iteration t
Q	=	total number of targets
$\bar{Q}(t)$	=	fraction of targets visited at iteration t
$\bar{X}(t)$	=	average throughput of target resolutions per agent at iteration t
N	=	total number of agents

Superscripts

n = agent identifier

*Undergraduate Researcher, Department of Intelligent Systems Engineering, ameighan@iu.edu

†Senior Researcher, Department of Intelligent Systems Engineering, jponniah@iu.edu

‡Assistant Professor, Department of Intelligent Systems Engineering, odantske@iu.edu

Subscripts

i = row identifier for belief matrix
 j = column identifier for belief matrix

Acronyms

UAV = unmanned aerial vehicle
HBS = hierarchical belief sharing
MDP = markov decision process
VANET = vehicular ad-hoc network
POMDP = partially observable markov decision process
DRL = deep reinforcement learning
MARL = multi-agent reinforcement learning

I. Introduction

The coverage problem, in which agents (i.e., autonomous UAVs, robots) find and visit moving targets on a two-dimensional grid, appears in many applications of interest. Civilian examples include delivering fire retardant to burning trees for wildfire containment,^{1,2} delivering emergency supplies to survivors for disaster relief/recovery,³ and providing wireless connectivity to first-responders in areas without pre-existing wireless infrastructure.⁴ Military applications include detecting/intercepting hostile intrusions in remote areas over land and sea.⁵ These applications have a common theme; a decentralized, heterogeneous system of agents shares local observations (or beliefs) and coordinates activity without a priori knowledge of the target locations. The system objective is to maximize coverage in the sense of visiting the most targets in the shortest possible time.

In this paper, we investigate the performance of one specific method of belief sharing for multi-agent coverage via a series of simulations. The method, hierarchical belief sharing (HBS), is based on multi-level clustering, where agents dynamically form “clusters” (i.e., organized groups of agents) to share their beliefs and schedule/coordinate their activity.⁶ Clusters rely on elected cluster-heads to facilitate both intra and inter-cluster communication. Higher-level clusters consist of cluster-heads, which in turn elect higher-level heads, until the entire system forms a hierarchical tree. Agents can leave and join new clusters better adapted to their evolving trajectories, while clusters can split and transfer agents as more agents spawn into the simulation, or merge and disband as agents de-spawn.

This approach serves a dual purpose. First, it allows cluster-heads to compress/down-sample information within their cluster before forwarding to agents outside their cluster. Each agent synthesizes a global view of the state-space, centered at itself, using the profile of compressed observations received via its cluster-head. Such processing reduces the bandwidth required for all agents to exchange local state information, an important task when the communication medium is constrained. Second, this approach simplifies the control/path-planning problem. The underlying hypothesis of this paper is that coverage applications are locally interactive. Agents require precise local state information (i.e., the locations of neighboring targets and agents for collision avoidance and coordination) but aggregated global state information to recognize high-level trends, for the system to maximize coverage. This hypothesis is implicit in both graphical models as well as graph attention networks in reinforcement learning. The hierarchical tree encodes the chain of local inter-agent dependencies; agents coordinate their activity within their cluster and rely on the cluster-head to ensure their actions complement those of the agents outside of their cluster.

Our simulations of HBS incorporate both the communication and control aspects of multi-agent coverage. Inter-agent communication occurs via the hierarchical tree. The control policy of each agent (i.e., its current heading), relies

on potential fields that attract agents to targets and repel agents from each other. Agents compute potential fields from their view of the global state-space, synthesized from the compressed local observations received from their cluster-heads. Although these fields have local minima that can immobilize or trap agents in loops, they are a useful proxy for more advanced methods based on Markov-Decision Processes (MDPs) that compute Nash equilibria or apply reinforcement learning. For example, game-theoretic approaches often rely on reward functions that simulate field potentials or have convex structure to ensure a Nash equilibrium exists.

We study and compare three different multi-agent coverage scenarios: first, where agents have a perfect view of the global state (i.e., the positions of the agents and targets are known), second, where agents rely on HBS to obtain a synthesized global view centered at themselves, and third, where agents rely only on their local observations. We show that HBS compares favorably to scenarios where each agent has perfect global state information (with less bandwidth), and significantly outperforms scenarios where agents are restricted to their local views. It is notably advantageous when targets are heterogeneous (i.e., require coverage from multiple agents), since agents see the views of their cluster members in fine detail (and external clusters in aggregate). For certain target distributions, HBS performs better than perfect state information; it distinguishes high-level features in the state-space from those that are local-interactive.

Exact optimal solutions (i.e., the shortest path for each agent to maximize coverage) are computationally infeasible in real-time, since there are multiple agents, the targets are mobile, and both agents and targets are heterogeneous. However, MDP-based methods can approximate optimal performance and remain computationally feasible when the state-space is locally-interactive. We validate and exploit the locally-interactive hypothesis (via HBS) to reduce the bandwidth required for system-wide coordination. This sets the stage for computationally feasible policies derived from reinforcement learning and/or graph attention networks.

The rest of the paper is organized as follows. Section II provides an overview of previous related work, Section III describes multi-level clustering, HBS, and the method of generating field potentials for motion-primitives from agent beliefs, Section IV includes the simulation results and analysis, and Section V concludes the paper and offers suggestions for future research.

II. Background

Multi-agent coverage involves wireless ad-hoc networking to support inter-agent communication, as well as multi-agent control and path-planning.^{7,8} All wireless communication through a shared medium is subject to fundamental scaling constraints on the total throughput. These constraints are independent of any communication protocol and depend only on properties of the channel such as noise and the path loss exponent.⁹ It turns out that the throughput per agent is sub-linear in the number of agents over all fading regimes and that scaleable coverage requires some form of hierarchical compression.¹⁰⁻¹² Clustering is common in protocols for wireless sensor networks and Vehicular Ad-Hoc Networks (VANETs). Agents within clusters usually share relatively stable communication channels.¹³ They elect cluster-heads, determine cluster members,¹⁴ and employ iterative methods during cluster formation to achieve consensus.¹⁵ Metrics that incorporate both the relative positions and velocities of the agents have also been proposed for creating stable clusters.¹⁶⁻²⁰

The “belief” of an agent in a Partially Observable Markov Decision Process (POMDP) is a random function of the state, which the agent observes in lieu of the state itself.^{21,22} Finding optimal control policies for high-dimensional state-spaces is generally NP-hard. However, this complexity can be reduced if the state-space is locally-interactive.²³⁻²⁷ Graphical models are a common way of representing local-interactive state-spaces (hierarchical clustering is one such representation). Belief propagation over graphical models of local-interactive dependencies (also known as Bayesian prediction) can achieve decentralized coordination for multi-agent, single-target coverage.²⁸ The Bayesian approach requires an acyclic network topology to ensure convergence (a topology that is present by design in hierarchical clustering but usually absent in ad-hoc networks).²⁹

Graphical models also appear in feature representation for deep reinforcement learning (DRL). These types of models, known as graph attention networks, appear in many applications of multi-agent reinforcement learning (MARL). They have been used to achieve scalable multi-robot scheduling for manufacturing operations³⁰ and induce multi-agent coordination for several MARL benchmarks such as the StarCraft II Multi-Agent Challenge.³¹ Coordinated action-

spaces in multi-agent systems have exponentially increasing complexity in the number of agents. One way of mitigating this complexity is via centralized training with a “critic” that knows the global state-space, but decentralized execution where the agents only have local knowledge.³² Another strategy, called credit assignment, seeks to reward those actions of individual agents that actually benefit the system as a whole instead of giving all agents the same flat reward for the system state.³³

The first efforts to integrate communication and control theory focus on the stability of multi-agent tracking with communication constraints.^{34,35} Subsequent work explores controlling robot formations so that robots remain connected.^{36,37} Contemporary approaches include hierarchical clustering with graph attention networks for reinforcement learning.³⁸ Since the optimal communication protocol for any multi-agent system is application-specific, the protocol syntax in principle, is a learning problem itself that can be solved using reinforcement learning³⁹ and/or graph attention networks.⁴⁰ Certain applications of multi-agent systems (i.e., multi-robot package-delivery) can be modeled as cooperative/competitive “graphical games” with time-varying communication channels,⁴¹ where the agent reward functions are field-potentials.⁴² In some instances, these games have Nash equilibria.⁴³ The optimal policies for such games can be found using graph attention networks.⁴⁴ Potential fields can also directly define multi-robot trajectories as in this paper.⁴⁵⁻⁴⁷

III. Methodology

We first review the basic concepts of multi-level clustering. Next, we introduce HBS which relies on multi-level clustering to share and aggregate the agent beliefs for multi-agent coverage. Finally, we generate potential fields for motion-primitives at each agent, from the beliefs obtained through HBS.

A. Hierarchical Multi-Level Clustering

This section describes rules that determine how agents dynamically form clusters based on their movement and positions, without bias towards any pre-existing hierarchy. As soon as an agent enters the environment it begins broadcasting messages to either join an existing cluster or create new clusters with other agents. Once an agent belongs to a cluster, its cluster-head determines the agents communication and clustering protocols.

1. Basic Rules

The following rules are tailored for multi-level clustering in real world environments.⁶

SINGLE-CLUSTER CONTROL Any given agent can only be the head of at most one cluster. Cluster-heads are responsible for cluster maintenance and all internal and external communication. This prevents dependency loops in which agents must wait for permission from themselves.

SUB-TREE INCLUSION Every agent, regardless of location in the hierarchy, conducts basic mission tasks which include observing local information in the environment. By rule, all cluster-heads must belong to level one clusters in their own sub-trees. This ensures agents transitioning from one cluster to another, do not issue conflicting directives.

2. Dynamic Clustering

Dynamic clustering in a hierarchical, multi-level system is built off of the basic clustering rules discussed above. Agents also require a cluster maintenance phase to ensure clusters remain stable when their relative positions and trajectories begin to shift. A cluster “token” is a data structure that resides with the cluster-head and tracks the members, head, and level of the cluster. Agents can leave their clusters, even when they are cluster-heads, by transferring their tokens to other cluster members.

Cluster-heads perform maintenance both periodically and when triggered by specific events. The following section describes the different protocols employed during cluster maintenance including how the cluster-token can be transferred

or destroyed, when and how cluster members transfer between neighboring clusters, how oversized clusters can split, and how nearby clusters can merge.

TOKEN TRANSFER & DESTRUCTION During the maintenance phase, the cluster-head can decide to transfer or destroy its token. A cluster-head initiates a transfer request in two types of scenarios. The first occurs during periodic maintenance in which a forced transfer of token is not necessary. If distance between the current cluster-head and the centroid is less than or equal to the distance between the centroid and the nearest idle member, it will retain its token. This avoids the cost associated with unnecessary transfer. The second occurs when the cluster-head can no longer perform its duties. In this case, the transfer becomes forced and the idle member nearest to the centroid receives the token regardless of current cluster-head position. During higher level transfers, cluster-heads consider all idle members of the subtree.

Cluster-heads can also opt to destroy their token as their member count approaches zero. In higher-level clusters, this demotion occurs when a cluster-head only has one member. When this happens, the cluster-head holding the higher-level token retains its position as a level one member in its subtree but loses its status as a cluster-head. In level one clusters, destruction of token only occurs when all members have left the cluster. At this point, the token is destroyed and the agent returns to an unclustered state.

MEMBER TRANSFER Agents can drift out of the communication range of their cluster while pursuing targets. Unscheduled transfers occur when the cluster-head drops the drifting agent resulting in a temporary disconnection from the tree. This forces the drifting agent to join another cluster on its own. By contrast, scheduled transfers avoid any loss of data. The cluster-head initiates a transfer request to its parent, which checks whether or not more suitable clusters are available. The parent possibly escalates the request further if none are found. When a suitable cluster is found, the agent is dropped from its current cluster and added to the new one.

SPLIT CLUSTER When a cluster becomes too large for onboard systems to maintain communication overhead between members, the cluster must split. The number of agents allowed in a cluster relies on a hyper-parameter set before simulation. This protocol allows a secondary agent to take partial responsibility while creating new space for unclustered agents to join the larger system. When a cluster-head realizes it does not have space for more members, it creates a new token to assign to the idle member closest to the centroid. Each cluster member is then assigned to one of the two clusters based on distance to the token-holder. Sometimes, the cluster initiating a split does not have a higher level cluster. Without additional action, this would result in two disconnected subtrees. To avoid this, the split-initializing cluster-head creates a third, higher-level token containing both cluster-heads from the split as members. This token is then given to an idle member closest to the two cluster-heads.

MERGE CLUSTER As agents move between clusters through join and transfer protocols, the size of clusters can become unbalanced. To mitigate this problem, smaller clusters can merge with neighboring clusters of the same level given the sum of members does not exceed the maximum cluster size. Clusters should not merge if they would have to split post operation. Once the receiving cluster confirms all agents can be held within one cluster, it conducts the transfers and the secondary cluster-token is destroyed.

B. Hierarchical Belief Sharing

The clustering method used in this paper supports intra- and inter- cluster communication. This section briefly describes the methods of information collection. It also describes how agents share their beliefs through cluster communication channels.

1. Agent Belief

Agents record local information from their current location and the surrounding area dictated by their observation radius. While the movement of agents occurs on a continuous plane, the agents maintain their beliefs in a discretized space

represented by a grid. Suppose the grid has $I \in \mathbb{N}$ rows and $J \in \mathbb{N}$ columns. The resolution $b_{i,j}^{(n)}(t) \in [0, 1]$ indicates the spatial resolution of data within the collected cell (i, j) at time-step (or iteration) t available to agent n . For local observations all resolutions are assigned a maximum resolution of 1. The age $a_{i,j}^{(n)}(t) \in \mathbb{N}$ specifies the time-step t that corresponds to the time at which the resolution $b_{i,j}^{(n)}(t)$ was recorded. The belief of agent n at time-step t is the matrix $B^{(n)}(t) := \{B_{i,j}^{(n)}(t) : i \in I, j \in J\}$ as depicted in Figure 1, where each entry $B_{i,j}^{(n)}(t) := (b_{i,j}^{(n)}(t), a_{i,j}^{(n)}(t))$ specifies the resolution and age of the data in cell (i, j) .

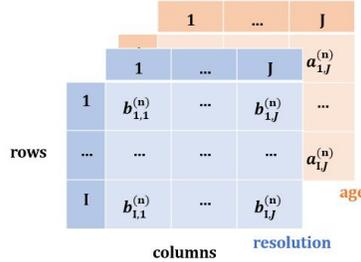


Figure 1. 3D belief matrix $B_{i,j}^{(n)}(t)$ held by an agent n at iteration t composed of 2D resolution and age matrices.

2. Aggregation

During the process of belief sharing, agents aggregate their most recent local and aggregate beliefs before sharing with their cluster-heads. A general aggregation function $\Phi : B \times B \rightarrow B$ is defined which maps two beliefs into one. Agents apply this general formulation regardless of whether they are aggregating their current global belief with another agent's belief or with their own most recent local observation. As paths overlap and agents begin to collect data from the same cells at the same or different times, choice of aggregation method becomes increasingly relevant. Two definitions of Φ are used in simulation. The first, Φ_A , prioritizes the age (or freshness) of the observation, and the second, Φ_R , prioritizes resolution.

AGE PRIORITIZATION Age prioritization applies to highly dynamic situations in which targets move around the environment. In this scenario, old beliefs with high resolution provide no value because of the rapidly evolving environment. Given agents m and n , cell-wise implementation of $\Phi_A(B^{(m)}, B^{(n)})$ is defined as

$$\Phi_A(B_{i,j}^{(m)}, B_{i,j}^{(n)}) := \begin{cases} B_{i,j}^{(m)} & \text{if } a_{i,j}^{(m)} > a_{i,j}^{(n)} \text{ or } (a_{i,j}^{(m)} = a_{i,j}^{(n)} \wedge b_{i,j}^{(m)} > b_{i,j}^{(n)}) \\ B_{i,j}^{(n)} & \text{otherwise} \end{cases} \quad (1)$$

This method of aggregation synthesizes beliefs $B^{(m)}(t)$ and $B^{(n)}(t)$ of agent m and n at time-step t . Simulation scenarios with mobile targets use age prioritization as the default aggregation method.

RESOLUTION PRIORITIZATION Resolution prioritization applies to scenarios where detailed knowledge of a cell outweighs the importance of age. Because of this, agents use the resolution prioritization in scenarios with stationary targets. Cell-wise implementation of $\Phi_R(B^{(m)}(t), B^{(n)}(t))$ is defined as

$$\Phi_R(B_{i,j}^{(m)}, B_{i,j}^{(n)}) := \begin{cases} B_{i,j}^{(m)} & \text{if } b_{i,j}^{(m)} > b_{i,j}^{(n)} \text{ or } (b_{i,j}^{(m)} = b_{i,j}^{(n)} \wedge a_{i,j}^{(m)} > a_{i,j}^{(n)}) \\ B_{i,j}^{(n)} & \text{otherwise} \end{cases} \quad (2)$$

3. Compression

The amount of information contained in a belief is, by assumption, directly proportional to the resolution of the data. This resolution multiplied with the data makes up the total information contained in the belief. For the rest of this section we omit the data component and only compare beliefs with the same data per full resolution cell.

The size of a belief is proportional to the resolution of its observations. As resolution approaches 1, the size of belief increases in size. Compressing the beliefs reduce both the communication burden and the complexity of computing field potentials. Locally interactive state-spaces can mitigate the effect of such compression on the system performance. If $c \in [0, 1]$ denotes the compression factor, define the compression calculation as

$$c \times B^{(n)}(t) := \{(c \times b_{i,j}^{(n)}(t), a_{i,j}^{(n)}(t)) : i \in I, j \in J\}. \quad (3)$$

The update rule for agent m after receiving agent n 's belief matrix, $B^{(n)}(t)$, through HBS at time-step $t + 1$ is:

$$B^{(m)}(t + 1) := \Phi(B^{(m)}(t), c \times B^{(n)}(t)), \quad (4)$$

where Φ in (4) is given by either (1) or (2) and then combined with (3). The locally interactive hypothesis implies that shared beliefs need not retain full resolution to provide value to agents. Thus incoming beliefs are compressed by a factor c . This reduces the amount of data agents transfer between each other while still conveying information on the global state.

4. Distribution of Beliefs

An agent carries two types of beliefs at all times while in simulation. The first, its ego belief, describes the agent's most recent local observation. The second, its aggregate belief, consists of the previous local observations along with other agents' beliefs acquired through HBS. Both are described in further detail below.

EGO BELIEF The local observation of an agent, O_t , is called its ego belief. At iteration t , cells within the agent's observation range $(b^{(n)}(t), a^{(n)}(t))$ are set to $(1, t)$. Assuming Φ in (5) is given by either (1) or (2), the agent can update its aggregate belief by combining the previous aggregate belief and the most recent observation such that

$$B^{(n)}(t) := \Phi(B^{(n)}(t - 1), O^{(n)}(t)). \quad (5)$$

AGGREGATE BELIEF The combination of the previous aggregate belief and the most recent observation makes up an agent's current aggregate belief. Agents can further augment this belief as they receive aggregated cluster beliefs from higher level cluster-heads. During simulation, cluster-heads generate a cluster belief by aggregating the compressed beliefs of all members. This cluster belief is then forwarded to higher-level clusters enabling them to aggregate their own cluster beliefs built from lower level observations. Once the highest level cluster generates its cluster belief, it passes the belief back down the chain such that level one members (leaf nodes) can receive information of the global state.

C. Field Potentials

Real-time path planning describes methods of determining an agent's next action based on its current knowledge of the environment. These methods allows agents to update their trajectory in real-time as they receive new information. In this work, potential fields direct agent motion towards targets scattered throughout the environment while repelling obstacles.

1. Field Forces

Potential fields are calculated based on an agents understanding of its view comprised of both local observations and global trends communicated by other agents through HBS. Potential fields have four components: target attraction, age attraction, agent repulsion, and boundary repulsion. Target and age attractions make up the positive forces in this work's potential fields. The attractive force from targets is calculated as

$$\mathbf{F}_1^{(n)}(t) = w_1 \sum_{q \in Q} b_{p_q}^{(n)}(t) \frac{p_q(a_{p_q}^{(n)}(t)) - p_n(t)}{|p_q(a_{p_q}^{(n)}(t)) - p_n(t)|^3}, \quad (6)$$

where $p_q(a_{p_q}^{(n)}(t))$ represents the location of target q at the age of agent n 's view matrix aligning with target q 's position. Here, w_1 acts as an attractive weight and $p_n(t)$ represents the location of agent n at iteration t . Variable $b_{p_q}^{(n)}(t)$ represents the agent's resolution of the cell containing target q at iteration $a_{p_q}^{(n)}(t)$. A secondary attractive force, age attraction, also affects the agents in an attempt to induce exploration of the environment. This attractive force is given through

$$\mathbf{F}_2^{(n)}(t) = w_2 \sum_{i \in I, j \in J} (t - a_{ij}^{(n)}(t)) \frac{p_{ij} - p_n(t)}{|p_{ij} - p_n(t)|^4}, \quad (7)$$

with w_2 being an attractive weight and p_{ij} representing the center position of the cell (i, j) . Factor $(t - b_{p_{ij}}^{(n)}(t))$ represents current iteration minus the agent's most recent recording of cell (i, j) . An increase in this difference results in a larger force towards the cell.

Agent and boundary repulsion make up the negative forces in this work's potential fields. The repulsive forces from nearby agents are calculated as

$$\mathbf{F}_3^{(n)}(t) = w_3 \sum_{\forall m \neq n} \frac{p_n(t) - p_m(t)}{|p_n(t) - p_m(t)|^3}. \quad (8)$$

Here, agents $m \neq n$ represent all agents in the vicinity of agent n . Variable w_3 represents a repulsive weight and $p_m(t)$ represents the location of the agent m at iteration t . The repulsive forces from boundaries are calculated as

$$\mathbf{F}_4^{(n)}(t) = w_4 \sum_g \frac{p_n(t) - p_g}{|p_n(t) - p_g|^4}, \quad (9)$$

where g represents a point from each of the four boundary lines closest in distance to agent n . The rectangular shape of the boundary facilitates this method of calculation. If the boundary shape became more complex, this method would become ineffective. The total force acting on an agent in the potential field equals the sum of the four forces above:

$$\mathbf{F}^{(n)}(t) = \mathbf{F}_1^{(n)}(t) + \mathbf{F}_2^{(n)}(t) + \mathbf{F}_3^{(n)}(t) + \mathbf{F}_4^{(n)}(t). \quad (10)$$

The resulting velocity of an agent n at iteration t is then given as

$$v_n = v_o \frac{\mathbf{F}^{(n)}(t)}{|\mathbf{F}^{(n)}(t)|}, \quad (11)$$

where all agents have a set velocity v_o .

2. Limitations

As the distance between the agent and various objects approaches infinity, the force approaches zero. In these scenarios, agents can have large numbers of objects in their global view, thus distance plays a critical role in finding the optimal path towards the nearest target.

While agents used potential fields with success in this study, fields do have a number of limitations that reduced simulation performance of all aggregation methods. Agents directed by potential fields can become stuck in local minima. As agents approach the targets position, the positive force acting on the agent becomes large enough to overpower all other forces. In this case, the agent becomes effectively trapped at the target's location. If removing the target requires coordination between agents, the ratio of agents to targets must be high. Without ample agents, all agents can end up trapped in local minima waiting for assistance from other agents. Additional modifiers can be added to increase the incentive to move towards targets who have agents nearby, but if the density of targets is high enough, agents will become trapped in local minima as they move towards stuck agents. Even with the limitations described above, potential fields are a useful precursor towards simulating cooperative behavior with more advanced path planning methods.

IV. Simulations and Results

To evaluate the performance of multi-level clustering communication, three coverage-related scenarios are simulated. Within these scenarios, agents use three aggregation methods to gain information about the environment which is then used for path planning. In the first method, all agents receive a perfect belief of the global state at time of spawn. This negates the necessity for communication, but places the computational burden of maintaining a perfect view of the environment on each agent. In the second method, agents rely on HBS to obtain a global view. Agents use their aggregated belief composed of local observations and shared beliefs facilitated by cluster communication. In the third method, agents rely only on the aggregation of their own local observations and avoid communication entirely.

Typical simulation environments for each of the three scenarios are shown in Figure 2. All scenarios occur in a grid-like environment in which targets are distributed throughout. These targets have a number of hyper-parameters that dictate their strength, distribution, and ability to move. Agents all spawn in a centralized location before expanding outwards in an effort to find and neutralize targets.

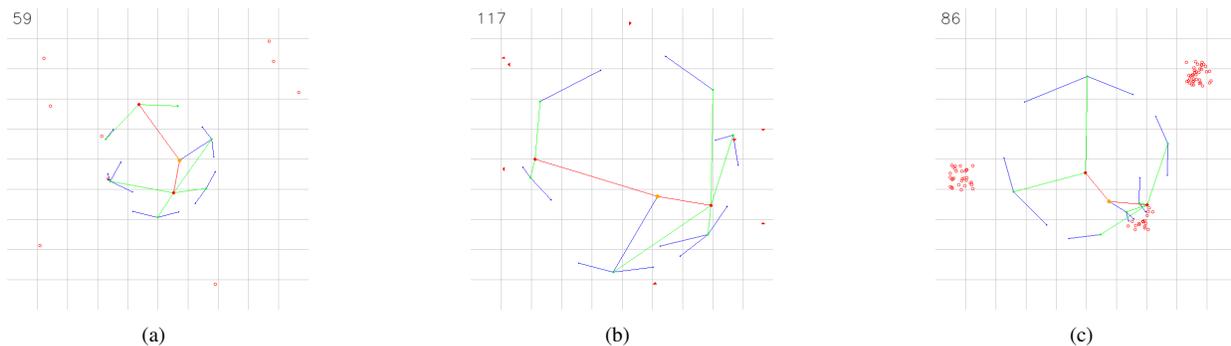


Figure 2. Simulation snapshots for uniform, moving, and dense targets. Level one clusters have blue links, level two clusters have green links, and level three clusters have red links. Cluster-members have the color of their cluster level, whereas cluster-heads have the color of a cluster one level higher. (a) Uniformly distributed targets denoted by hollow red circles. (b) Moving targets denoted by solid red triangles. (c) Densely clustered targets denoted by hollow red circles.

A. Metrics

The primary metric discussed is performance on the control problem. For agents using belief sharing, compression factor is also evaluated.

CONTROL PROBLEM METRICS The ratio of visited targets and the average throughput determines performance in the three control scenarios. The ratio of targets visited is represented by $\bar{Q}(t)$, and is defined as

$$\bar{Q}(t) = \frac{Q_v(t)}{Q}, \quad (12)$$

where $Q_v(t)$ is the number of visited targets at time t and Q is the total number of targets at $t = 0$. The second metric, average throughput at time t is represented by $\bar{X}(t)$, and is defined as

$$\bar{X}(t) = \frac{\sum_{n=1}^N \sum_{i=1}^I \sum_{j=1}^J b_{i,j}^{(n)}(t)}{N}, \quad (13)$$

where N represents the number of agents in simulation. When agents have a perfect view of the scenario, this metric becomes a constant value equal to the number of elements in the view matrix. Because the resolution relates closely with the complexity of the potential field calculation, this metric allows the communication burden to be shown in relation to the efficiency of agents.

B. Target Scenarios

As mentioned previously, three types of targets were used in simulation testing. Twenty seeds were chosen at random for each scenario. For communication, compression factors $c \in 1.0, 0.8, 0.6, 0.4, 0.2$ were chosen. Figure 2 shows an average view of each scenario at varying stages of an individual trial. Each scenario and their respective results are described in further detail below.

1. Uniformly Distributed Targets

The initial scenario imitates the well-known traveling salesman problem for multiple agents. Target locations are randomly sampled from a uniform distribution before getting placed on the grid. To induce cooperation between agents, target locations must be visited concurrently by two agents before getting removed from the target list. Without coordination, agents will not reach an optimal solution. Figure 3(a) illustrates the convergence of the target ratio for each of the three aggregation methods. Within the methods shown in the legend, “C c ” denotes a system of agents using HBS with a compression factor of c .

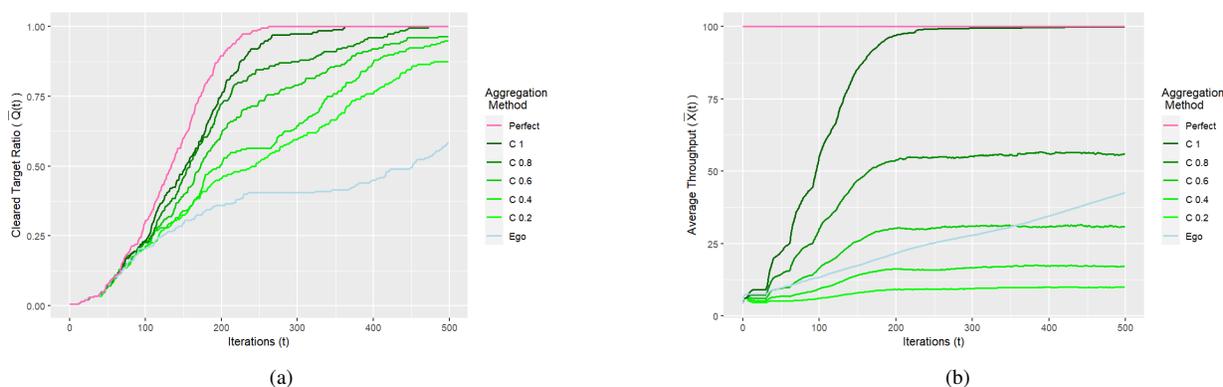


Figure 3. The performance of HBS for uniformly distributed targets under different compression factors. (a) The target-cleared ratio vs time plot. Perfect state knowledge has the fastest clearance rate whereas ego-exclusive views have the slowest rate. Multi-agent systems using HBS perform best without compression. (b) The average throughput vs time plot. Perfect state knowledge requires the most throughput, while HBS converges to a maximum throughput only when the systems uses no compression. Ego-exclusive views require more throughput than HBS at certain compression factors.

In early stages of simulation, agents must expand outwards to cover the space. This occurs regardless of prior knowledge because of the agents’ starting location. This is shown in Figure 3(a) where all aggregation methods follow a similar trajectory at the start. As the simulation progresses, it becomes clear that a perfect view prior to spawn results in the quickest removal of all the targets from the grid. Cluster communication, especially with smaller compression factors, lags only slightly behind. The real benefit of cluster communication is shown in Figure 3(b) which shows that when compression reduces information in shared beliefs by twenty percent, the average throughput decreases by forty-five percent. Not only does this decrease the communication cost, it also decreases the complexity of potential field calculations. While this does result in a decrease in clearance rate, it is only pronounced near the end of simulation when agents are tracking down interest points on the outskirts of the grid. In comparison, the ego view struggles greatly and does not clear more than sixty-percent of the targets on average. Because these agents do not share their beliefs, coordination is not possible.

2. Moving Targets

A second scenario takes the random sample of uniformly distributed targets from before and enables the targets to move around the grid space. In this scenario two agents must visit a target concurrently to remove it from the grid. Figures 4(a) and 4(b) show similar results when compared with the first scenario. The main difference is a slightly slowed rate of convergence. This result is expected as agents must continuously update their trajectory as they approach a

moving target. Again, two main results should be noted. First, the difference between perfect view and HBS with no compression contains minimal difference apart from the end in which agents using HBS take a longer to find targets at edge of the grid space. Second, the average throughput drastically reduces as compression increases. Drops in target acquisition rate do occur, but reduction of complexity delivers obvious benefits in real world environments.

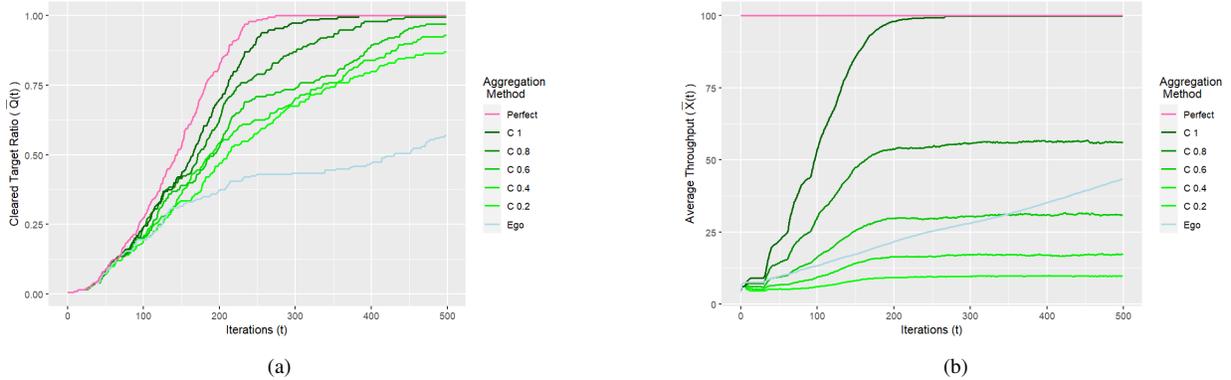


Figure 4. The performance of HBS for moving targets under different compression factors. (a) The target-cleared ratio vs time plot. (b) The average throughput vs time plot.

3. *k*-Clustered Targets

The final scenario is modeled after the wild-fire containment problem. Target locations are sampled from a Gaussian Mixture Model with three components formulated below.

Let $K = 3$ be the number of components in the mixture. The weights π_k are all set equal to $\frac{1}{K}$. Let μ_k be the mean vector in which each mean is randomly sampled from a uniform distribution of the grid space, and Σ_k be equal to $\sigma^2 \times I$ where I denotes the identity matrix.

$$k \sim \text{Discrete}(\pi_1, \pi_2, \dots, \pi_K) \quad (14)$$

$$x \sim \mathcal{N}(\mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} \exp\left(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)\right) \quad (15)$$

First, the component index k is sampled according to the weights π_k . The target positions are sampled from the Gaussian distribution indexed by k . Since the target density is high in this scenario, only one agent need visit a target to remove it from the grid. Therefore agents must share their observations in high density cells to attract neighboring agents for assistance that would otherwise be idle. Without belief sharing, such sub-optimal outcomes are inevitable.

Figure 5(a) shows clear differences from previous scenarios. With densely clustered targets, HBS outperforms perfect prior knowledge. Moreover, even within systems using HBS, higher levels of compression outperforms no compression. The distinction between between perfect prior knowledge and belief sharing becomes most pronounced as agents attempt to locate and travel to the third and final cluster. This surprising outcome suggests that agents can sometimes have too much information for their own good (i.e., ignorance is bliss).

The explanation is as follows. Agents with perfect views instantly receive all target locations after spawning. Hence, many agents simply move to the same closest cluster of targets. This (sub-optimal) behavior is increasingly evident as the number of targets increases; potential fields that repel agents from each other, are overwhelmed by the attractive fields of the target clusters. However, HBS at high compression levels suppresses these extreme field potentials and allows agents to spread out and explore and find other target clusters beyond those initially encountered. Figure 5(a) offers proof that belief sharing with no compression performs worse than all other levels of compression. This also suggests restricting an agent's view of the global state via compression can lead to implicit coordination; agents act in

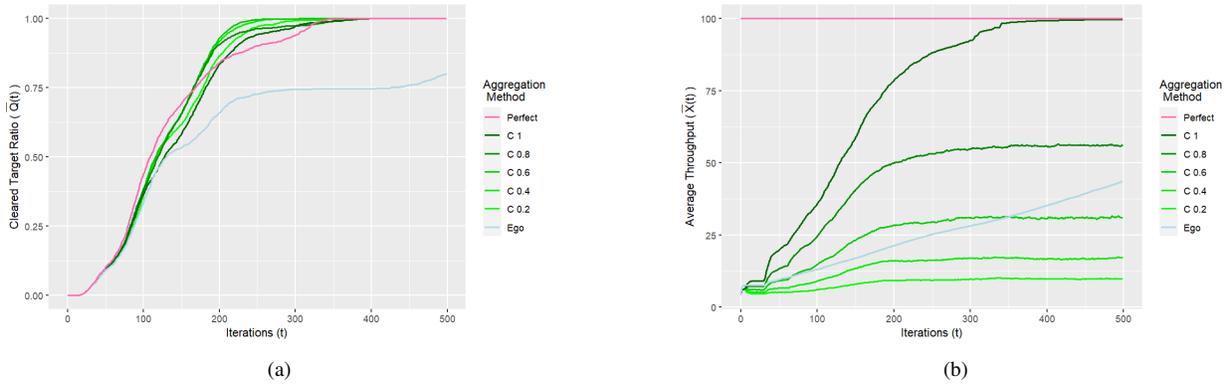


Figure 5. The performance of HBS for dense targets under different compression factors. (a) The target-cleared ratio vs time plot. HBS exceeds no-compression because it is less sensitive to remote low-level features of the state-space (b) The average throughput vs time plot. HBS significantly reduces the average throughput over the operating lifetime.

the interest of the system (as opposed to their own) because remote low-level trends are lost in the compression but high-level trends remain.

V. Summary and Future Work

This paper proposed HBS for multi-agent coverage with different target classes and distributions. HBS uses multi-level clustering so that agents can synthesize global views of the state space from the profile of compressed beliefs received from their cluster-heads. Multi-agent coverage is local-interactive; agents only require precise local observations but aggregated remote observations of the state-space to behave in a coordinated manner. The local-interactive hypothesis was tested by simulations of HBS with field-potential-induced motion primitives. The simulations showed that HBS reduces the communication throughput required for optimal coverage and surpasses the performance of systems with perfect knowledge of the global state.

These results suggest that HBS fits well with more advanced methods of motion-planning based on reinforcement learning. Hierarchical trees model the inter-agent dependencies (i.e., features) that simplify the learning process. Similar techniques such as graph attention networks have already shown promise. HBS is also inherently scaleable (in the number of agents, targets, and grid dimension) due to its hierarchical structure and tailored for much higher levels of heterogeneity than simulated in this paper. Future work will explore these limits in more sophisticated and challenging settings.

References

- ¹Haksar, R. N. and Schwager, M., “Distributed Deep Reinforcement Learning for Fighting Forest Fires with a Network of Aerial Robots,” 2018 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1067–1074.
- ²Julian, K. D. and Kochenderfer, M. J., “Distributed Wildfire Surveillance with Autonomous Aircraft Using Deep Reinforcement Learning,” *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 8, 2019, pp. 1768–1778.
- ³Goodchild, A. and Toy, J., “Delivery by drone: An evaluation of unmanned aerial vehicle technology in reducing CO2 emissions in the delivery service industry,” *Transportation Research Part D: Transport and Environment*, Vol. 61, 2018, pp. 58–67, Innovative Approaches to Improve the Environmental Performance of Supply Chains and Freight Transportation Systems.
- ⁴Katila, C. J., Di Gianni, A., Buratti, C., and Verdone, R., “Routing protocols for video surveillance drones in IEEE 802.11s Wireless Mesh Networks,” 2017 *European Conference on Networks and Communications (EuCNC)*, 2017, pp. 1–5.
- ⁵Nuske, S., Choudhury, S., Jain, S., Chambers, A., Yoder, L., Scherer, S., Chamberlain, L., Cover, H., and Singh, S., “Autonomous Exploration and Motion Planning for an Unmanned Aerial Vehicle Navigating Rivers: Autonomous Exploration and Motion Planning for a UAV Navigating Rivers,” *Journal of Field Robotics*, Vol. 32, 06 2015.
- ⁶Ponniah, J., Theile, M., Dantsker, O., and Caccamo, M., “Autonomous Hierarchical Multi-Level Clustering for Multi-UAV Systems,” *AIAA Scitech 2021 Forum*, 2021, p. 0656.
- ⁷Ren, W. and Beard, R., “Consensus seeking in multiagent systems under dynamically changing interaction topologies,” *IEEE Transactions on Automatic Control*, Vol. 50, No. 5, 2005, pp. 655–661.

- ⁸Ponniiah, J. and Dantsker, O. D., “Strategies for Scaleable Communication and Coordination in Multi-Agent (UAV) Systems,” Aerospace, Vol. 9, No. 9, 2022.
- ⁹Gupta, P. and Kumar, P., “The capacity of wireless networks,” IEEE Transactions on Information Theory, Vol. 46, No. 2, 2000, pp. 388–404.
- ¹⁰Xie, L.-L. and Kumar, P., “A network information theory for wireless communication: scaling laws and optimal operation,” IEEE Transactions on Information Theory, Vol. 50, No. 5, 2004, pp. 748–767.
- ¹¹Ozgun, A., Leveque, O., and Tse, D. N., “Hierarchical Cooperation Achieves Optimal Capacity Scaling in Ad Hoc Networks,” IEEE Transactions on Information Theory, Vol. 53, No. 10, 2007, pp. 3549–3572.
- ¹²Ghaderi, J., Xie, L.-L., and Shen, X., “Hierarchical Cooperation in Ad Hoc Networks: Optimal Clustering and Achievable Throughput,” IEEE Transactions on Information Theory, Vol. 55, No. 8, 2009, pp. 3425–3436.
- ¹³Vodopivec, S., Bester, J., and Kos, A., “A Survey on Clustering Algorithms for Vehicular Ad-Hoc Networks,” 07 2012, pp. 52–56.
- ¹⁴Ephremides, A., Wieselthier, J., and Baker, D., “A design concept for reliable mobile radio networks with frequency hopping signaling,” Proceedings of the IEEE, Vol. 75, No. 1, 1987, pp. 56–73.
- ¹⁵Daenabi, A., Pour Rahbar, A. G., and Khademzadeh, A., “VWCA: An efficient clustering algorithm in vehicular ad hoc networks,” Journal of Network and Computer Applications, Vol. 34, No. 1, 2011, pp. 207–222.
- ¹⁶Basu, P., Khan, N., and Little, T., “A mobility based metric for clustering in mobile ad hoc networks,” Proceedings 21st International Conference on Distributed Computing Systems Workshops, 2001, pp. 413–418.
- ¹⁷Zhang, Z., Boukerche, A., and Pazzi, R., “A novel multi-hop clustering scheme for vehicular ad-hoc networks,” 10 2011, pp. 19–26.
- ¹⁸Wei Fan, Yan Shi, Shanzhi Chen, and Longhao Zou, “A mobility metrics based dynamic clustering algorithm for VANETs,” IET International Conference on Communication Technology and Application (ICCTA 2011), 2011, pp. 752–756.
- ¹⁹Hassanabadi, B., Shea, C., Zhang, L., and Valaee, S., “Clustering in Vehicular Ad Hoc Networks using Affinity Propagation,” Ad Hoc Networks, Vol. 13, No. PART B, 2014, pp. 535–548.
- ²⁰Katsikas, L., Chatzikokolakis, K., and Alonistioti, N., “Implementing Clustering for Vehicular Ad-Hoc Networks in Ns-3,” Proceedings of the 2015 Workshop on Ns-3, WNS3 ’15, Association for Computing Machinery, New York, NY, USA, 2015, p. 25–31.
- ²¹Sondik, E. J., “The Optimal Control of Partially Observable Markov Processes Over the Infinite Horizon: Discounted Costs,” Operations Research, Vol. 26, No. 2, 1978, pp. 282–304.
- ²²Kaelbling, L. P., Littman, M. L., and Cassandra, A. R., “Planning and Acting in Partially Observable Stochastic Domains,” Artificial Intelligence, Vol. 101, 1995, pp. 99–134.
- ²³Boutilier, C., “Planning, Learning and Coordination in Multiagent Decision Processes,” Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge, TARK ’96, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996, p. 195–210.
- ²⁴Stone, P. and Veloso, M., “Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork,” Artificial Intelligence, Vol. 110, No. 2, 1999, pp. 241 – 273.
- ²⁵Roth, M., Simmons, R., and Veloso, M., “Exploiting Factored Representations for Decentralized Execution in Multiagent Teams,” Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS ’07, Association for Computing Machinery, New York, NY, USA, 2007.
- ²⁶Oliehoek, F. A., Whiteson, S., and Spaan, M. T., “Approximate solutions for factored Dec-POMDPs with many agents,” Belgian/Netherlands Artificial Intelligence Conference, 2013, pp. 340–341.
- ²⁷Amato, C., Chowdhary, G., Geramifard, A., Üre, N. K., and Kochenderfer, M. J., “Decentralized control of partially observable Markov decision processes,” 52nd IEEE Conference on Decision and Control, Dec 2013, pp. 2398–2405.
- ²⁸Bourgault, F. and Durrant-Whyte, H. F., “Communication in General Decentralized Systems and the Coordinated Search Strategy,” In Proc. of FUSION’04, 2004, pp. 723–770.
- ²⁹Grime, S., Durrant-Whyte, H. F., and Ho, P., “Communication in Decentralized Data-Fusion Systems,” 1992 American Control Conference, 1992, pp. 3299–3303.
- ³⁰Wang, Z., Liu, C., and Gombolay, M., “Heterogeneous graph attention networks for scalable multi-robot scheduling with temporospatial constraints,” Autonomous Robots, Vol. 46, No. 1, 2022, pp. 249–268.
- ³¹Li, S., Gupta, J. K., Morales, P., Allen, R., and Kochenderfer, M. J., “Deep Implicit Coordination Graphs for Multi-Agent Reinforcement Learning,” Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS ’21, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2021, p. 764–772.
- ³²Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I., “Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments,” Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17, Curran Associates Inc., Red Hook, NY, USA, 2017, p. 6382–6393.
- ³³Zhou, M., Liu, Z., Sui, P., Li, Y., and Chung, Y. Y., “Learning Implicit Credit Assignment for Cooperative Multi-Agent Reinforcement Learning,” Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS’20, Curran Associates Inc., Red Hook, NY, USA, 2020.
- ³⁴Olfati-Saber, R. and Murray, R., “Distributed structural stabilization and tracking for formations of dynamic multi-agents,” Proceedings of the 41st IEEE Conference on Decision and Control, 2002., Vol. 1, 2002, pp. 209–215 vol.1.
- ³⁵Saber, R. and Murray, R., “Flocking with obstacle avoidance: cooperation with limited communication in mobile networks,” 42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475), Vol. 2, 2003, pp. 2022–2028 Vol.2.
- ³⁶Ji, M. and Egerstedt, M., “Distributed Coordination Control of Multiagent Systems While Preserving Connectedness,” IEEE Transactions on Robotics, Vol. 23, No. 4, 2007, pp. 693–703.
- ³⁷Fax, J. and Murray, R., “Information flow and cooperative control of vehicle formations,” IEEE Transactions on Automatic Control, Vol. 49, No. 9, 2004, pp. 1465–1476.
- ³⁸Ryu, H., Shin, H., and Park, J., “Multi-Agent Actor-Critic with Hierarchical Graph Attention Network,” Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, No. 05, Apr. 2020, pp. 7236–7243.
- ³⁹Foerster, J. N., Assael, Y. M., de Freitas, N., and Whiteson, S., “Learning to Communicate with Deep Multi-Agent Reinforcement Learning,” Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS’16, Curran Associates Inc., Red Hook, NY, USA, 2016, p. 2145–2153.

⁴⁰Niu, Y., Paleja, R., and Gombolay, M., “Multi-Agent Graph-Attention Communication and Teaming,” Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '21, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2021, p. 964–973.

⁴¹Fernando, M., Senanayake, R., Azad, A., and Swany, M., “Graphical Games for UAV Swarm Control Under Time-Varying Communication Networks,” Workshop on Intelligent Aerial Robotics: From Autonomous Micro Aerial Vehicles to Sustainable Urban Air Mobility and Operations, Vol. abs/2205.02203, 2022.

⁴²Fernando, M. and Liu, L., “Swarming of Aerial Robots with Markov Random Field Optimization,” 2020.

⁴³Fernando, M., Senanayake, R., and Swany, M., “CoCo Games: Graphical Game-Theoretic Swarm Control for Communication-Aware Coverage,” IEEE Robotics and Automation Letters, Vol. 7, No. 3, 2022, pp. 5966–5973.

⁴⁴Fernando, M., Senanayake, R., Choi, H., and Swany, M., “Graph Attention Multi-Agent Fleet Autonomy for Advanced Air Mobility,” Robotics: Science and Systems, Vol. abs/2302.07337, 2023.

⁴⁵Khatib, O., “Real-time obstacle avoidance for manipulators and mobile robots,” Proceedings. 1985 IEEE International Conference on Robotics and Automation, Vol. 2, 1985, pp. 500–505.

⁴⁶Chuang, Y.-L., Huang, Y. R., D’Orsogna, M. R., and Bertozzi, A. L., “Multi-Vehicle Flocking: Scalability of Cooperative Control Algorithms using Pairwise Potentials,” Proceedings 2007 IEEE International Conference on Robotics and Automation, 2007, pp. 2292–2299.

⁴⁷Bennet, D. J. and McInnes, C. R., “Distributed control of multi-robot systems using bifurcating potential fields,” Robotics and Autonomous Systems, Vol. 58, No. 3, 2010, pp. 256–264, Towards Autonomous Robotic Systems 2009: Intelligent, Autonomous Robotics in the UK.