A Game Theoretic Approach to Distributed Coverage of Graphs by Heterogeneous Mobile Agents

A. Yasin Yazıcıoğlu* Magnus Egerstedt* Jeff S. Shamma*

* *Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: {yasin,magnus,shamma}@ gatech.edu).*

Abstract: In this paper, we study the problem of covering an apriori unknown network structure by a group of mobile agents with possibly nonidentical coverage and communication capabilities. The network is represented as a graph, and the agents are mobile devices that explore and cover the graph in a decentralized fashion. We study this problem in a game theoretic setting and achieve optimal coverage through learning in games.

1. INTRODUCTION

In many networks, one typical task is to provide some service over the network via distributed agents with limited capabilities. This service may take on different forms such as security, maintenance, delivery, or access to some resources (e.g., Goddard et al. (2005); Du et al. (2003); Berbeglia et al. (2010)). One possible way for a group of distributed agents to achieve such a task is to partition the network into responsibility regions and let each agent take care of its own region. This approach turns the initiation of the task into a *distributed coverage* problem on a graph.

The distributed coverage problem is widely studied for continuous structures (e.g., Cortés et al. (2004); Poduri and Sukhatme (2004); Schwager et al. (2009); Pimenta et al. (2008)), whereas some extensions to discretized spaces (e.g., Durham et al. (2009); Yun and Rus (2012)) also exist in the literature. There is also a rich literature in graph theory on related combinatorial optimization problems such as *location-allocation*, *graph partitioning*, and *set cover*. These problems pertain to tasks such as optimally locating a number of resources on a graph, partitioning the nodes into a number of minimally-interconnected communities, or choosing a small number of node-subsets, whose union contains all the nodes of a graph, from a given set of options. Detailed reviews of the literature on these combinatorial optimization problems can be found in Reese (2006); Fjällström (1998); Caprara et al. (2000) and the references therein.

In this work, we study a *distributed graph coverage* problem, where a group of mobile agents with possibly different coverage and communication capabilities are initially deployed at some arbitrary nodes of an unknown graph. Depending on the deployment, initially the agents may be randomly scattered on the graph, or they may be at the same location. The agents maximize the number of covered nodes by simultaneously exploring the graph and covering the nodes within their responsibility regions. We assume that each agent has distance limitations in their coverage and communication capabilities resulting in a cover range and a communication range respectively. The responsibility region of an agent contains the nodes within its cover range, and this region is only known to the other agents within its communication range. The distributed graph coverage problem constitutes a large amount of uncertainty given that the graph is apriori unknown, and the initial configuration of the agents is arbitrary. In order to tackle the challenges in this problem, we adopt a game theoretic formulation and utilize learning in games.

Game theoretic methods have been used to analyze many related problems such as vehicle-target assignment (e.g., Arslan et al. (2007)), coverage optimization in static sensor networks (e.g., Marden and Wierman (2008)), or dynamic vehicle routing (e.g. Arsie et al. (2009)). In Marden et al. (2009), the authors establish a connection between the game theoretic approaches and cooperative control problems such as coverage or consensus. The use of potential games and variants of log-linear learning in solving similar multi-agent problems is presented in Marden and Shamma (2012). More recently, in Zhu and Martínez (2013), the authors employ learning in games to achieve power-aware coverage of a discretized space by sensors with limited capabilities.

In this paper, we formulate the distributed graph coverage problem in a game theoretic setting. More precisely, we design a potential game for this problem and utilize a variant of log-linear learning to maximize the coverage. Throughout the dynamics, each agent only covers the nodes within its cover range, and it can observe the actions of the other agents only when it is within their communication range. Furthermore, the agents are only allowed to update their actions locally, i.e. in each time step, every agent either maintains its position or moves to a neighboring node. We show that, under this setting, the proposed scheme globally maximizes (in probability) the number of covered nodes.

The organization of this paper is as follows: Section 2 presents the distributed graph coverage problem. Section 3 is on the proposed game theoretic formulation and solution. Some simulation results are presented in Section 4. Finally, Section 5 concludes the paper.

★ This research was supported by ONR project #N00014-09-1-0751.

2. DISTRIBUTED GRAPH COVERAGE

In this section, we introduce the distributed graph coverage (DGC) problem. The goal is to maximize the number of covered nodes subject to limited coverage and communication capabilities of the mobile agents. We start by presenting some graph theoretic preliminaries.

2.1 Graph Theory Concepts

An undirected graph $\mathcal{G} = (V, E)$ consists of a *node set* V and an *edge set* $E \subseteq V \times V$. For an undirected graph, the edge set consists of unordered node pairs (v, v') denoting that the nodes v and v' are adjacent.

For any two nodes v and v', the *distance* between the nodes $d(v, v')$ is the number of edges in a shortest path connecting them. A graph is *connected* if the distance between any pair of nodes is finite.

The set of nodes containing a node v and all the nodes adjacent to v is called the *(closed) neighborhood* of v, and it is denoted as N_v. Similarly, for any $\delta \geq 0$, the δ-neighborhood of v, N_v^δ, is the set of nodes that are at most δ away from v, i.e.

$$N_v^\delta = \{v' \in V \mid d(v, v') \leq \delta\}. \tag{1}$$

2.2 Problem Formulation

Consider a network represented as a connected undirected graph $\mathcal{G} = (V, E)$, and let a set of m mobile agents $P = \{p_1, p_2, \ldots, p_m\}$ be located on some nodes of $\mathcal{G}$. Let each mobile agent, p_i, be covering some $C_i \subseteq V$ for $i = 1, 2, \ldots, m$. We refer to C_i as the *coverage set* of agent p_i. Then, the set of covered nodes, $C \subseteq V$, is given as

$$C = \bigcup_{i=1}^{m} C_i. \tag{2}$$

In general, the mobile agents may be heterogeneous in their sensing and mobility mechanisms resulting in different coverage capabilities. We assume that each agent p_i has a *cover range* δ_i, and it can cover nodes that are at most δ_i away from its current position, $v_i \in V$. As such, the coverage set of each p_i is the δ_i-neighborhood of v_i, i.e.

$$C_i = N_{v_i}^{\delta_i}. \tag{3}$$

Similarly, we assume that each agent has a distance-limited communication capability, and each agent p_j conveys its current coverage set to the other agents within its *communication range* δ_j^c. Communication is used to let the agents covering the same nodes know about each other's actions. Through the distance-limited communications, an agent p_i can only observe the current coverage set of p_j, if it is within a distance δ_j^c of p_j's current position. As such, the set of agents whose coverage sets are observable by p_i is a function of the agent positions, $O_i(v_1, \ldots, v_m)$, defined as

$$O_i(v_1, \ldots, v_m) = \{p_j \in P \mid d(v_i, v_j) \leq \delta_j^c\}, \tag{4}$$

where v_i and v_j denote the current positions of p_i and p_j respectively. Given a set of possibly heterogeneous mobile agents with limited coverage and communication capabilities, the goal in the distributed graph coverage problem is to define locally applicable rules for the agents to follow such that the number of covered nodes is asymptotically maximized. A rule is considered to be *locally applicable*, if its execution by an agent p_i only depends on the current actions of the agent itself and the other agents who are currently observable by p_i as given in (4).

Definition *(Distributed Graph Coverage Problem)*: Let m mobile agents $P = \{p_1, p_2, \ldots, p_m\}$ be initially arbitrarily deployed on an unknown connected graph $\mathcal{G} = (V, E)$. For each p_i, let $v_i(t) \in V$ denote its position at time t, and let its coverage set $C_i(t)$ be given as in (3). The distributed graph coverage (DGC) problem aims to find some locally applicable rules for the agents to follow in order to asymptotically maximize the number of covered nodes, $|C(t)|$, subject to $v_i(t) \in N_{v_i(t-1)}$ for all $p_i \in P$.

The constraint in the DGC problem imposes locality on the movement of agents on the graph. As such, each agent can either maintain its position or move to an adjacent node at each time step.

2.3 Solution Approach

In the DGC problem, a group of possibly heterogeneous agents are required to explore an unknown graph and to cover as many nodes as possible. A widely used approach in solving this type of combinatorial optimization problems is to find a sufficiently good approximation in a short period of time (e.g. Jia et al. (2002); Kuhn and Wattenhofer (2005); Abrams et al. (2004)). In this fashion, one possible approach to the DGC problem is to utilize a distributed greedy method. Accordingly, the agents may try to maximally increase the number of nodes they cover at each step until none of them can make a further improvement. However, the resulting performance significantly depends on the graph structure and the initial configuration. This method may quickly lead to a good approximation if the agents start with a sufficiently good initial coverage. On the contrary, it may lead to very inefficient configurations for arbitrary graphs and initial conditions. For instance, consider a simple case in Fig. 1, where 2 agents with cover ranges of 1 can obtain an optimal coverage in 2 time steps. In this example, the agents can not reach the optimal configuration through a greedy approach, and the resulting approximation ratio can be made arbitrarily small by adding even more nodes attached to the initially uncovered hub.

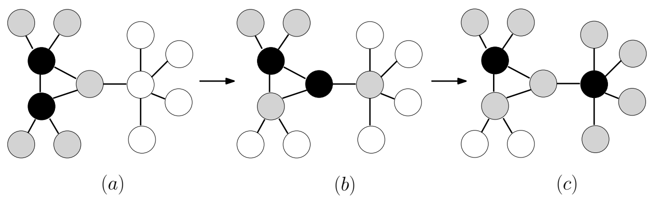


Fig. 1. A possible trajectory to an optimal coverage in a simple example. Two agents (shown as black) with cover ranges of 1 are initially located as in (a). The number of covered nodes (shown as gray) is reduced in the intermediate step (b), and it is maximized later in the final configuration (c).

In order to achieve an optimum configuration in the DGC problem, a solution method should be able to handle the uncertainties in the setting. To this end, the solution may need to occasionally allow for graph exploration at the expense of a better coverage. In this work, we present such a solution by approaching the problem from a game theoretic perspective. As such, we design a corresponding game and employ learning in games to drive the agents to desired configurations. In the resulting scheme, while the agents have high probabilities of taking actions that locally improve coverage, they may also take locally worse actions with much smaller probabilities in order to allow for further exploration and search for a global optimum. In general, the resulting scheme is slower compared to greedy methods. However, the overall configuration converges (in probability) to the set of globally optimum configurations.

3. GAME THEORETIC FORMULATION

In this section, we present the game theoretic formulation of the DGC problem and the proposed solution. Before starting our analysis, we provide some game theoretic preliminaries.

3.1 Game Theory Concepts

A finite *strategic game* $\Gamma = (P, A, U)$ consists of three components: (1) a set of m *players (agents)* $P = \{p_1, p_2 \ldots, p_m\}$, (2) an m-dimensional *action space* $A = A_1 \times A_2 \times \ldots \times A_m$, where each A_i is the *action set* of player p_i, and (3) a set of *utility functions* $U = \{U_1, U_2, \ldots, U_m\}$, where each $U_i : A \mapsto \Re$ is a mapping from the action space to real numbers.

For any *action profile* $a \in A$, let a_{-i} denote the actions of players other than p_i. Using this notation, an action profile a can also be represented as (a_i, a_{-i}), and we will frequently use this representation in this paper.

An action profile $a^* \in A$ is called a *Nash equilibrium* (NE) if for all players $p_i \in P$,

$$U_i(a_i^*, a_{-i}^*) = \max_{a_i} U_i(a_i, a_{-i}^*). \tag{5}$$

In this paper, we will consider a particular class of games that is widely utilized in cooperative control problems, namely *potential games*. In potential games, there exists a *potential function* $\phi : A \mapsto \Re$ such that the change of a player's utility resulting form its unilateral deviation from an action profile equals the resulting change in ϕ. More precisely, for each player p_i, for every $a_i, a_i' \in A_i$, and for all $a_{-i} \in A_{-i}$,

$$U_i(a_i', a_{-i}) - U_i(a_i, a_{-i}) = \phi(a_i', a_{-i}) - \phi(a_i, a_{-i}). \tag{6}$$

For cooperative control applications, usually the potential game is designed such that the potential function ϕ represents some global score depending on the actions of all agents.

3.2 Game Design

In order to formulate the DGC problem in a game theoretic setting, first we need to design a corresponding game. More specifically, we are interested in designing a potential game $\Gamma_{DGC} = (P, A, U)$ that will be repetitively played by the agents. For potential games, the agents can asymptotically maximize the corresponding potential function by following a learning algorithm such as log-linear learning proposed in Blume (1993). Hence, if we design Γ_{DGC} such that it has a potential function aligned with the number of covered nodes, then the agents can utilize a similar learning algorithm to solve the DGC problem.

We start the game design by defining the action sets. In the distributed graph coverage problem, each agent's coverage set is defined by its cover range δ_i and position v_i. Since each agent has a fixed cover range, its coverage set only depends on its position, which can be defined as its action. As such each agent has its action set equal to the node set of $\mathcal{G} = (V, E)$, i.e.

$$A_i = V, \quad \forall i \in \{1, 2, \ldots, m\}. \tag{7}$$

Based on the action sets in (7), an action profile $a \in A$ is the vector of current agent positions.

Next, we need to define the utility functions to complete the game design. In order to obtain a potential game that can be used to solve the DGC problem, first we pick the number of covered nodes to be the potential function $\phi(a)$, i.e.

$$\phi(a) = |\bigcup_{i=1}^{m} C_i|. \tag{8}$$

Then, we set U such that $\phi(a)$ in (8) is a potential function for the resulting game. This can be achieved by setting the agent utility functions as

$$U_i(a) = |C_i \setminus \bigcup_{j \neq i} C_j|. \tag{9}$$

In other words, for any action profile, each agent gets a utility equal to the number of nodes that are only covered by itself. Note that this utility is equal to the marginal contribution of the corresponding agent to the potential function.

Proposition 1. Utility functions in (9) lead to a potential game $\Gamma_{DGC} = (P, A, U)$ with the potential function given in (8).

Proof. Let $a_i = v_i$ and $a_i' = v_i'$ be two possible actions for agent p_i, and let a_{-i} denote the actions of other agents. Note that

$$\begin{aligned}\phi(a) &= |\bigcup_{i=1}^{m} C_i| = |C_i \setminus \bigcup_{j \neq i} C_j| + |\bigcup_{j \neq i} C_j| \\ &= U_i(a_i, a_{-i}) + |\bigcup_{j \neq i} C_j|. \end{aligned} \tag{10}$$

Using (10) we get,

$$\phi(a_i', a_{-i}) - \phi(a_i, a_{-i}) = U_i(a_i', a_{-i}) - U_i(a_i, a_{-i}). \tag{11}$$

3.3 Learning Dynamics

In the game theoretic formulation of the DGC problem, starting from an arbitrary initial configuration, the agents repetitively play the coverage game Γ_{DGC}. At each time instant, $t \in \{0, 1, 2, \ldots\}$, each agent, $p_i \in P$, plays an action, $a_i(t)$, and receives a utility, $U(a_i(t), a_{-i}(t))$. In this setting, the role of learning is to provide an action update rule to the agents such that, in repetitive plays, the agent actions converge to the set of desired action profiles. For the DGC problem, this is the set of action profiles that globally maximize the number of covered nodes.

For a potential game, a learning algorithm known as *log-linear learning* (LLL) can be used to induce an irreducible, aperiodic Markov chain on the action space whose stationary distribution has arbitrarily small entries for action profiles that do not maximize the potential function $\phi(a)$ Blume (1993). However, the classical LLL assumes that each player p_i has access to all the actions in its action set A_i as well as the hypothetical utilities it would gather by playing them. In general, the convergence to the set of potential maximizers is not guaranteed when the system evolves on constrained action sets, i.e. when each agent p_i is allowed to choose its next action action $a_i(t+1)$ only from a subset of actions $A_i^c(a_i(t))$ that depends on its current action $a_i(t)$. Note that this is indeed the case for the DGC problem, and we have

$$A_i^c(v) = N_v. \tag{12}$$

The issue of constrained action sets was addressed in Marden and Shamma (2012), and it was shown that a variant learning algorithm called *binary log-linear learning* (BLLL) may be used to achieve guaranteed convergence (in probability) to the set of potential maximizer action profiles if the constrained action sets satisfy the following two properties.

Property 1 (Reachability) For any agent $p_i \in P$ and any action pair $a_i^0, a_i^k \in A_i$, there exists a sequence of actions $\{a_i^0, a_i^1, \ldots, a_i^k\}$ such that $a_i^r \in A_i^c(a_i(r-1))$ for all $r \in \{1, 2, \ldots, k\}$.

Property 2 (Reversability) For any agent $p_i \in P$ and any action pair $a_i, a_i' \in A_i$,

$$a_i' \in A_i^c(a_i) \iff a_i \in A_i^c(a_i').$$

Lemma 2. The constrained action sets in (12) satisfy Properties 1 and 2 if the graph $\mathcal{G} = (V, E)$ is connected.

Proof. If the graph is connected, then there exists a finite-length path $\{v^0, \ldots, v^k\}$ between any pair of nodes $v^0, v^k \in V$, and Property 1 is satisfied. Furthermore, for undirected graphs, $v' \in N_v$ if and only if $v \in N_{v'}$. Hence, Property 2 is also satisfied.

In BLLL, a single agent is randomly chosen at each time step. The selected agent, assuming that all the other agents are stationary, updates its action depending on its current utility and the hypothetical utility it would receive by playing a random action in its constrained action set. The general BLLL algorithm presented in Marden and Shamma (2012) is as follows:

BLLL Algorithm (Marden and Shamma (2012))

initialization: $t = 0$, $T \in \Re^+$ small, $a(0) \in A$
while (1)
  Pick a random $p_i \in P$.
  Pick a random $a_i' \in A_i^c(a_i(t))$.
  $a_j(t+1) = a_j(t)$ for all $p_j \neq p_i$.
  $\alpha = e^{U_i(a(t))/T}$.
  $\beta = e^{U_i(a_i', a_{-i}(t))/T}$.
  $a_i(t+1) = \begin{cases} a_i(t) & \text{w.p. } \frac{\alpha}{\alpha+\beta}, \\ a_i' & \text{otherwise.} \end{cases}$
  $t = t + 1$.
end while

Note that the selection of a single agent at each time step can be achieved without the necessity of agents coordinating to decide who should update. For instance, using the asynchronous time model in Boyd et al. (2006), where each agent has a clock that ticks according to a rate 1 Poisson process, an agent is chosen whenever its clock ticks.

3.4 Sufficient Communications

The DGC problem can be solved by a group of mobile agents, if all agents employ BLLL in a repetitive play of the proposed potential game $\Gamma_{DGC} = (P, A, U)$. However, for this approach to work, each agent should be able to gather some necessary information from the actions of agents it can observe as given in (4). More specifically, based on the actions of agents in O_i, each agent p_i should be able to compute the utility it gathers from the current action profile as well as the hypothetical utilities it may gather by moving to a neighboring node. This requirement can be met if the agents have some sufficient communication ranges. In order to keep the overall process as local as possible, we present the smallest communication ranges that ensure proper execution of the algorithm.

Lemma 3. Let $P = \{p_1, \ldots, p_m\}$ be a set of m agents, and let each agent p_i have a cover range δ_i and a communication range δ_i^c. For any action profile $a \in A$, an agent p_i located at a node v_i can measure the number of agents covering a node v if

$$d(v_i, v) \leq \min_{j=1,\ldots,m} (\delta_j^c - \delta_j). \tag{13}$$

Proof. We will show that if (13) is satisfied, then p_i is within the communication range of any agent covering v, i.e. $d(v_i, v_k) \leq \delta_k^c$ for any agent p_k covering v. Let p_k be an agent located at a node v_k, and let p_k be covering v. From triangle inequality we have

$$d(v_i, v_k) \leq d(v_i, v) + d(v_k, v). \tag{14}$$

Plugging (4) and (13) into (14) we get,

$$d(v_i, v_k) \leq \min_{j \in \{1,\ldots,m\}} \delta_j^c - \delta_j + \delta_k. \tag{15}$$

Since $\min_{j \in \{1,\ldots,m\}} \delta_j^c - \delta_j \leq \delta_k^c - \delta_k$ we get

$$d(v_i, v_k) \leq \delta_k^c - \delta_k + \delta_k = \delta_k^c, \tag{16}$$

which implies that agent p_i is within the communication range of p_k.

Using Lemma 3, we present sufficient communication ranges for agents to gather the necessary information to execute BLLL.

Corollary 4. Let $P = \{p_1, \ldots, p_m\}$ be a set of m agents, and let each agent p_i have a cover range δ_i and a communication range δ_i^c. Each agent p_i can compute $U_i(a_i', a_{-i}(t))$ for any action $a_i' \in A_i^c(a_i(t))$ if

$$\delta_j^c - \delta_j \geq \delta^* + 1, \quad \forall p_j \in P, \tag{17}$$

where δ^* is the maximum cover range defined as

$$\delta^* = \max_{j \in \{1,2,\ldots,m\}} \delta_j. \tag{18}$$

Proof. If (17) holds, then we have

$$\min_{j=1,\ldots,m} (\delta_j^c - \delta_j) \geq \delta^* + 1. \tag{19}$$

Lemma 3 and (19) together imply that an agent p_i located at v_i knows the number of agents covering any node v satisfying

$$d(v_i, v) \leq \delta^* + 1. \tag{20}$$

In light of (12), an updating agent p_i can at most be 1 hop away from its current position in the next time step. Hence, its coverage set in the next time step can only contain nodes that are at most $\delta_i + 1$ away from its current position. Note that any such node v satisfies (20) since for any agent p_i we have $\delta_i \leq \delta^*$. Hence, p_i can compute $U_i(a_i', a_{-i}(t))$ for any action $a_i' \in A_i^c(a_i(t))$.

Theorem 5. Let $P = \{p_1, p_2, \ldots, p_m\}$ be a set of m heterogeneous agents, each having a cover range δ_i and a communication range δ_i^c satisfying (17). If all the agents follow BLLL in evolving their actions along the constrained action sets given in (12), then the number of covered nodes is asymptotically maximized (in probability) in any repetitive play of Γ_{DGC}.

Proof. If each agent has a communication range satisfying (17), then Corollary 4 implies that any updating agent can implement BLLL to pick its new action from the constrained action set given in (12). Due to Theorem 5.1 in Marden and Shamma (2012), our Lemma 2 ensures that, in the resulting repetitive play of Γ_{DGC}, the action profile converges (in probabilty) to the set of actions that globally maximize the number of covered nodes.

4. SIMULATION RESULTS

In this section, we present some simulation results for the proposed method and compare them to the results obtained through a greedy approach. We simulate a case where a group of heterogeneous agents are initially placed at an arbitrary node of a connected random geometric graph. This can represent a scenario where a group of heterogeneous mobile robots are deployed to an arbitrary room of an unknown building, and they are required to explore and cover the overall structure. The random geometric graph used in the simulation has a diameter of 16, and it consists of 100 nodes and 281 edges. A group of 15 agents are initially placed at an arbitrary node of the graph. 10 of the agents have cover ranges of 1 and communication ranges of 4, whereas 5 of them have cover ranges of 2 and communication ranges of 5. The agents run BLLL algorithm with $T = 0.1$, and they achieve a complete coverage within 3064 time steps. The change in the number of covered nodes throughout the dynamics is shown in Fig. 2, whereas the coverage at some instants are depicted in Fig. 3. After $t = 3064$, the agents maintain a complete coverage with a very high probability.

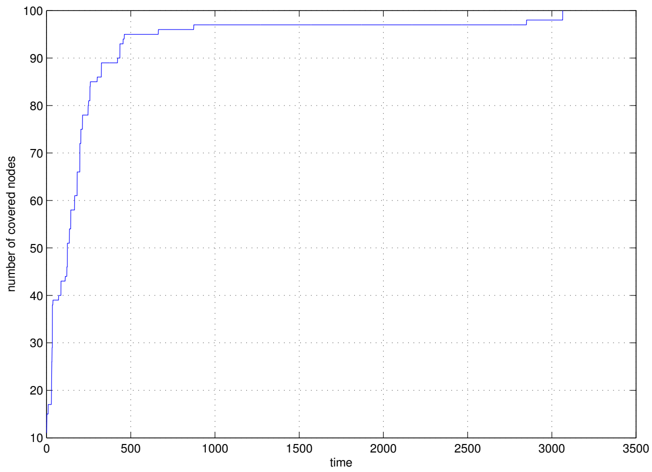


Fig. 2. The number of covered nodes as a function of time. 15 heterogeneous agents initially start at an arbitrary location and use the proposed method to cover a graph consisting of 100 nodes. The number of covered nodes is initially 11, whereas a complete coverage is reached in 3064 time steps.

We also simulated the same scenario for agents starting at the same initial condition as the one shown in Fig. 3 and following a greedy approach. The greedy approach is implemented in two fashions. In both of them, a single agent is picked to update its action at each time step. In the first version, we assume that an agent, which can make the largest increase in the number of covered nodes, is picked at each step. Since this requires a global coordination among the agents, we refer to this version as the coordinated greedy approach. In the second version, a random agent is picked at each time step. We refer to this version as the uncoordinated greedy approach, since it does not require a global coordination. In both versions, if the picked agent can increase the number of covered nodes by moving to a neighboring node, then it moves to the one resulting in the maximum increase. Note that, from a game theoretic perspective, these schemes realize better-reply paths on the action space of Γ_{DGC}, and they are guaranteed to converge to an equilibrium since Γ_{DGC} is a potential game. The coordinated greedy approach rapidly (in 9 time steps) converges to an action profile where 37 nodes are covered. For the uncoordinated case, the randomness in the choice of updating agent results in possibility of converging to different equilibrium points. The uncoordinated greedy approach was simulated for 100 times, and the resulting distributions for the number of covered nodes in steady state and the convergence time were obtained. Results for the greedy methods are depicted in Fig. 4. As these results also point out, the greedy methods are generally much faster compared to the proposed method, but they may get stuck at some very inefficient equilibrium.

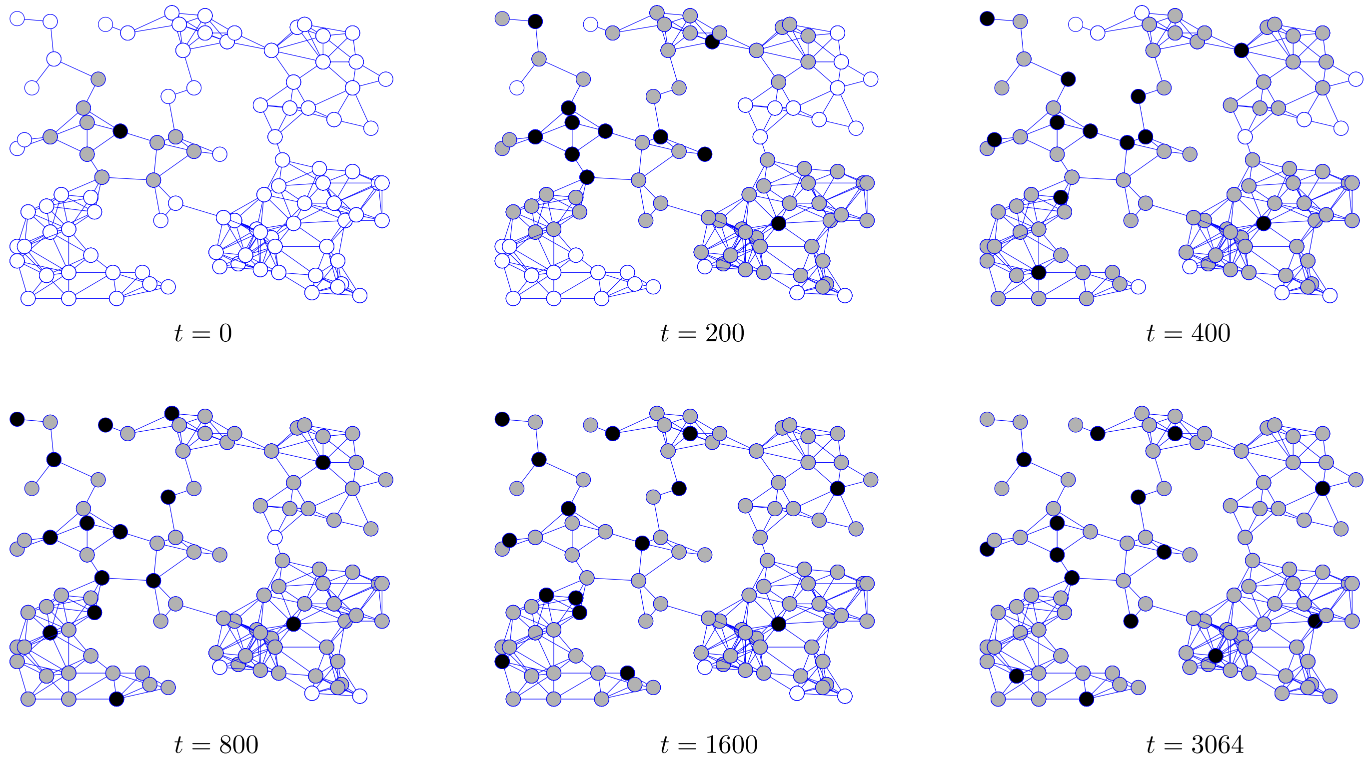
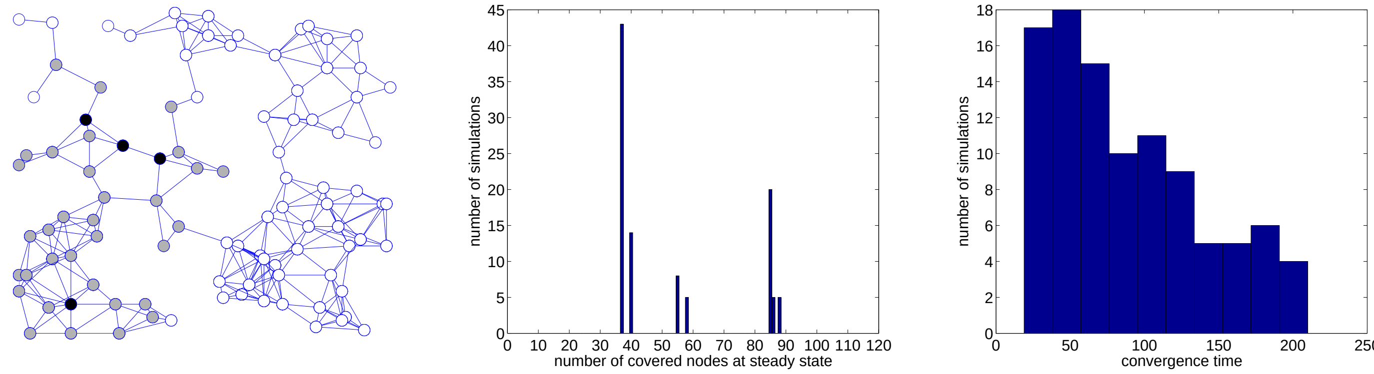


Fig. 3. Distributed coverage of an apriori unknown connected graph by a group of 15 agents initially located at the same node. 10 agents have cover ranges of 1 and communication ranges of 4, whereas 5 agents have cover ranges of 2 and communication ranges of 5. At each instant, covered nodes are shown as white, nodes having at least one guard located at them are shown as black, and nodes covered by at least one guard are shown as gray.



(a) (b) (c)

Fig. 4. Coverage results for the coordinated and uncoordinated greedy approaches for agents starting at the same initial condition as the one in Figure 3. The coordinated case converges in 9 steps to the configuration in (a) where only 37 nodes are covered. The uncoordinated case converges to different equilibrium points due to the random agent-pick at each step, and the distribution for the number of covered nodes at steady state (b) and the coverage time (c) are shown for a set of 100 simulations.

5. CONCLUSION

In this work, distributed coverage of apriori unknown graphs by possibly heterogeneous mobile agents with local sensing and communication capabilities is studied. In this context, the agents are considered to be heterogeneous in terms of their distance-limited coverage and communication capabilities. The problem is approached from a game theoretic perspective, and a solution is obtained by designing a corresponding potential game and employing binary log-linear learning. It is shown that if the agents have sufficient communication ranges as presented in the paper, then they can asymptotically maximize the number of covered nodes by using the proposed method in locally updating their actions. Simulation results for the proposed method are also presented in the paper along with a comparison to the results of a greedy approach.

REFERENCES

Abrams, Z., Goel, A., and Plotkin, S. (2004). Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In *International Symposium on Information Processing in Sensor Networks*, 424–432.

Arsie, A., Savla, K., and Frazzoli, E. (2009). Efficient routing algorithms for multiple vehicles with no explicit communications. *IEEE Transactions on Automatic Control*, 54(10), 2302–2317.

Arslan, G., Marden, J., and Shamma, J.S. (2007). Autonomous vehicle-target assignment: a game theoretical formulation. *ASME Journal of Dynamic Systems, Measurement, and Control*, 584–596.

Berbeglia, G., Cordeau, J.F., and Laporte, G. (2010). Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202(1), 8–15.

Blume, L.E. (1993). The statistical mechanics of strategic interaction. *Games and Economic Behavior*, 5(3), 387–424.

Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006). Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6), 2508–2530.

Caprara, A., Toth, P., and Fischetti, M. (2000). Algorithms for the set covering problem. *Annals of Operations Research*, 98(1-4), 353–371.

Cortés, J., Martínez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2), 243–255.

Du, T.C., Li, E.Y., and Chang, A.P. (2003). Mobile agents in distributed network management. *Communications of the ACM*, 46(7), 127–132.

Durham, J.W., Carli, R., Frasca, P., and Bullo, F. (2009). Discrete partitioning and coverage control with gossip communication. In *ASME Dynamic Systems and Control Conference*, 225–232.

Fjällström, P.O. (1998). Algorithms for graph partitioning: A survey. *Linköping University Electronic Press*, 3.

Goddard, W., Hedetniemi, S.M., and Hedetniemi, S.T. (2005). Eternal security in graphs. *J. Combin. Math. Combin. Comput*, 52, 169–180.

Jia, L., Rajaraman, R., and Suel, T. (2002). An efficient distributed algorithm for constructing small dominating sets. *Distributed Computing*, 15(4), 193–205.

Kuhn, F. and Wattenhofer, R. (2005). Constant-time distributed dominating set approximation. *Distributed Computing*, 17(4), 303–310.

Marden, J.R., Arslan, G., and Shamma, J.S. (2009). Cooperative control and potential games. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(6), 1393–1407.

Marden, J.R. and Shamma, J.S. (2012). Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation. *Games and Economic Behavior*, 75(2), 788–808.

Marden, J.R. and Wierman, A. (2008). Distributed welfare games with applications to sensor coverage. In *IEEE Conference on Decision and Control*, 1708–1713.

Pimenta, L., Kumar, V., Mesquita, R.C., and Pereira, G. (2008). Sensing and coverage for a network of heterogeneous robots. In *IEEE Conference on Decision and Control*, 3947–3952.

Poduri, S. and Sukhatme, G.S. (2004). Constrained coverage for mobile sensor networks. In *IEEE International Conference on Robotics and Automation*, 165–171.

Reese, J. (2006). Solution methods for the p-median problem: An annotated bibliography. *Networks*, 48(3), 125–142.

Schwager, M., Rus, D., and Slotine, J.J. (2009). Decentralized, adaptive coverage control for networked robots. *International Journal of Robotics Research*, 28(3), 357–375.

Yun, S. and Rus, D. (2012). Distributed coverage with mobile robots on a graph: Locational optimization. In *IEEE International Conference on Robotics and Automation*, 634–641.

Zhu, M. and Martínez, S. (2013). Distributed coverage games for energy-aware mobile sensor networks. *SIAM Journal on Control and Optimization*, 51(1), 1–27.