Abstract—This paper studies the coverage problem in an unknown environment by a Mobile Sensor Network (MSN). Each agent in the MSN has sensing, communication, computation and moving capabilities to complete sensing tasks. Here the agents need to relocate themselves, from their initial random locations, to their optimal configuration. The proposed algorithm is based on game theory control where a collection of distributed agents use their local information to make decisions. A state-based potential game is defined in which each agent’s utility function is designed to consider the trade off between the worth of the covered area and the energy consumption. The agents employ binary log-linear learning to update their actions in each iteration in order to converge to the Nash equilibrium. As the agents do not have the knowledge of the sensing area, a Maximum Likelihood estimation scheme is used to estimate the unknown parameters of a Gaussian Mixture Model (GMM). Then in order to feed the estimation algorithm with more informative data, a mutual information term is added to the agents’ utility functions. The mutual information is utilized to determine which observation can improve the agent’s knowledge of the unobserved area more. Simulation results are provided to verify the performance of the proposed algorithm.

I. INTRODUCTION

In the coverage problem, a group of mobile sensor agents seek to collect and process data to collectively cover an area. The coverage problem for Mobile Sensor Networks (MSNs) can be modelled as one of the resource allocation problems. These problems entail a collection of dispersed interacting components that seek to optimize a global collective objective through local decision making. Limited communication capabilities, local and dynamic information, faulty components, and an uncertain environment make the problem very complicated. Hence distributed control of MSNs is the main focus of recent research in this area.

In most previous studies on the coverage problem, a probabilistic function is defined which represents the probability or occurrence frequency of events in a task area. Agents start from their initial random locations in the task area and search to find their best configuration for sensing more valuable regions. In [1], the virtual force, a combination of attractive and repulsive forces, has been utilized to determine the updated sensors’ locations at each iteration. After finding an effective configuration, a one-time movement with energy consideration is done. In their study, the coverage worth of the area is assumed to be a uniform function where the agents just change their locations slightly in order to minimize the intersection of their covered area. In [2], an optimization problem is defined to maximize the sensors’ coverage while taking into account the communication cost. Another perspective is used in [3] where the area is partitioned to Voronoi regions and each agent is just responsible for sensing its own region. Then each agent converges to the centroid of its region while at the same time the Voronoi partitions are adjusted using the worth of the area and neighbors’ locations. In all the above mentioned studies, the distribution of the sensing worth in the area is known a priori by all agents. Unfortunately, this is not a realistic assumption in most applications.

It is desirable to introduce a method to enable the optimal coverage of an unknown environment. There are just a few works which assume that the agents do not have this prior knowledge. These studies assume that instead, each agent determines the worth of an area after sensing it. In [4], a solution similar to [3] is offered. The summation of $n$ distributions is assumed as an estimation model in which the estimation algorithm can adjust just the weighting functions of these distributions. Because the only optimization variables are the weighting coefficients, the designer has to use a large number of distributions with different means and covariances to improve the accuracy of the model. This estimation scheme does not perform well in real applications where the mission area is large because $n$ has to be increased proportional to the size of the mission area and hence a high computational burden is caused for the MSN.

One approach for solving the coverage problem is to model the problem as a non-cooperative game, where the players (mobile sensors here) independently pursue their own objectives. The game theory control method has many other advantages including robustness to failures and environmental disturbances, reducing communication requirements and improving scalability. The primary goal of game theory-based approaches is to design rules that guarantee the existence and efficiency of a pure Nash equilibrium [5]. Proper utility functions and reinforcement learning methods are designed for the coverage game of MSNs in [5], [6]. In these algorithms, each player must have access to the utility values of its alternative actions. However, in an unknown environment such an assumption is infeasible due to the lack of knowledge of the area’s worth. Such information constraints are taken into account in [7], [8]. Different learning algorithms are defined where the agents experiment alternative actions without pre-evaluation of these actions’ values. These learning algorithms can be interpreted as a trial-and-error approach for the agents. However, such trial-and-error approach results in unnecessary movements of the agents and hence increases the energy consumption. It also decreases the rate of convergence and causes chattering in applications.

In this paper, we solve the coverage problem in an unknown environment using a game theory method by extending the results in [5] to overcome the above mentioned
limitations. We can categorize our results into three main parts: (1) In Section III, the game theory control is used to design a utility function based on each agent’s marginal sensing contribution and the energy consumption of the agent. Each agent uses only local information to complete its tasks in a distributed manner. It is shown that the defined game is a state-based potential game which is guaranteed to have at least one Nash equilibrium. In addition, the binary log-linear learning is used for updating the agents’ actions which guarantees that the agents will converge stochastically to a Nash equilibrium. However, this game and learning design just works when the agents have a prior knowledge about the worth of the area.

(2) In Section IV, an estimation scheme is defined to overcome the lack of prior knowledge of the worth of the area in an unknown environment. A Gaussian Mixture Model (GMM) is introduced where each agent uses its previous observations to estimate the unknown model parameters. Then the maximum likelihood (ML) method is employed to calculate these parameters. Choosing future actions wisely by using the agent’s previous observations will reduce the improper effect of the trial-and-error method in the literature.

(3) In Section V, the entropy criterion is further utilized to select informative observations. The agents will consider the mutual information between their observed and unobserved regions in order to find more informative observations. It will be shown that the modified game with the addition of a proper term corresponding to the mutual information to the utility function will remain a state-based potential game.

II. PROBLEM DEFINITION

In this paper, a class of MSN applications is considered where a limited number of sensors are randomly deployed in a task area. The area’s worth is different at different locations. The sensors’ task is to change their locations in order to maximize the total worth of the covered area at their final configuration while minimizing the total energy consumption. The area’s worth can be interpreted as the probability of finding a target by sensors in that area. The agents do not have prior knowledge about the worth. However, the worth of a location will be determined by an agent after sensing it.

A convex two-dimensional mission space, that is discretized into a (squared) lattice, is considered. Each square of the lattice has unit dimensions and is labelled with the coordinate of its center \( q = (x_q, y_q) \). The collection of all squares of the lattice is denoted by \( Q \). A numerical variable \( f_q \geq 0 \) is assigned to the worth or the probability of the occurrence of an event in each square with center \( q \in Q \). The larger the worth \( f_q \), the more important the sensing of the square with center \( q \). In this paper, \( f_q \) is assumed to be stationary. However, the result of this paper can be extended to a non-stationary field. There are \( N \) mobile sensor agents in the field where the location of agent \( i = 1, \ldots, N \) is denoted by \( a_i(t) = (x_i(t), y_i(t)) \in Q \). The sensing region of agent \( i \) is modeled as a disc with the center \( a_i(t) \) and the radius \( a_i^r(t) \). Each area \( a_i(t) \) is chosen from a discrete set with the minimum and the maximum equal to \( r_{\text{min}} \) and \( r_{\text{max}} \), respectively. It is assumed that all the agents have the same \( r_{\text{min}} \) and \( r_{\text{max}} \) while they can have different discrete sets. The area covered by agent \( i \), \( S(a_i(t)) \), is a function of the agent’s action \( a_i(t) \), where \( a_i(t) := (a_i^x(t), a_i^y(t)) \in A_i \) and \( A_i \) is the available action set for agent \( i \). The action profile of all agents is denoted by \( a(t) = (a_1(t), \ldots, a_N(t)) \in A := \prod_{i=1}^N A_i \). The motion of the agents will be limited to their adjacent lattices if there is no obstacle there. We use \( C_i(a_i(t-1)) \) to denote the available actions for agent \( i \) at time step \( t \). Each agent is able to communicate with its neighbors to exchange information. The set of neighbors of agent \( i \) is given by \( N_i^{\text{comm}}(a_i(t)) := \{j = 1, \ldots, N | (x_i - x_j)^2 + (y_i - y_j)^2 \leq (R_i^{\text{comm}})^2 \} \), where \( R_i^{\text{comm}} \) is the communication range of agent \( i \).

III. GAME DESIGN

A. Utility Design

Mobile sensors consume energy for their movement, communication and sensing. Due to energy limitations, taking the energy consumption into account in the utility design will improve the performance of MSNs. To decrease the communication energy usage the communication range and the amount of data transfer should be reduced. On other hand, if \( R_i^{\text{comm}} \) is assumed less than \( 2r_{\text{max}} \), there might be a region which is sensed by two agents while they can not communicate with each other. This will cause a waste of energy. Here it is assumed that \( R_i^{\text{comm}} = 2r_{\text{max}}, i = 1, \ldots, N \). In addition, each agent has to just send its own action \( a_i(t) \) to its neighbors which does not need significant communication between agents. Because the communication range, \( R_i^{\text{comm}} \), and the amount of data transfer cannot be further reduced in our problem set-up, there is no need to incorporate the communication energy into our optimization cost function.

We then consider the energy consumption caused by sensing. There is a trade off between the power usage and the size of the covered region. The energy consumption of agent \( i \) due to sensing is defined as \( E_i^{\text{sense}}(a_i^z(t))^2 \), where \( K_i > 0 \) is a coefficient. Hence \( a_i^z(t) \) is used as an optimization variable where each agent attempts to optimize its own power usage by finding a proper radius.

The energy consumption of agent \( i \) due to movement is defined as \( E_i^{\text{move}} = K_i^0(a_i^r(t) - z_i(t))^2 \), where \( K_i^0 > 0 \) is a coefficient and \( z_i(t) \equiv a_i^r(t - 1) \) is the previous location of agent \( i \). As it is seen the movement energy consumption, and the utility to be designed next as a result, is a function of the state (the agent’s previous location in this case). Here the term state has a different meaning in the context of game theory in comparison with that in control theory. In our problem the term state denotes the previous location of the agent. This is the reason that a state-based potential game will be designed for this problem.

We now proceed to formulate our coverage problem as a state-based game. A utility function \( U_i \) is designed for each agent that aims to capture the trade off between the worth of the covered area and the energy consumption by agent \( i \).

The utility function for agent \( i \) is designed as

\[
U_i(a(t), z(t)) = F(a_i(t), a_{-i}(t)) - F(a_i^0(t), a_{-i}(t)) - K_i(a_i^r(t))^2 - K'_i(|a_i^z(t) - z_i(t)|)
\]

\[
F(a(t)) = \sum_{q \in \bigcup_{i=1}^N S(a_i(t)) \cap Q} f_q,
\]

(1)
where \( F(a(t)) \) denotes the worth of the covered area by the agents, \( a_i^0(t) \) is the null action for agent \( i \) (equivalently, the nonexistence of agent \( i \)), \( a_{-i}(t) \) is the actions of all agents other than agent \( i \), so \( a_i(t) = (a_i(t), a_{-i}(t)) \). Here \( F(a_i(t), a_{-i}(t)) - F(a_i^0(t), a_{-i}(t)) \) is agent \( i \)'s marginal contribution to sense the area. Note that the utility function \( U_i \) is local over the sensed region by agent \( i \). Here \( U_i \) is dependent only on the actions of \( \{i\} \cup N_i^{1\text{sn}}(a(t)) \), where \( N_i^{1\text{sn}}(a(t)) \) is the set of all agents whose sensing regions have an intersection with that of agent \( i \). As mentioned before, by setting \( R_i^{\text{comm}} = 2r_{\text{max}}, \forall i = 1, ..., N \), if \( j \in N_i^{1\text{sn}}(a(t)) \), it follows that \( j \in N_i^{1\text{sn}}(a(t)) \). That is, when the agents have a sensing intersection, they can communicate with each other. The definition of the marginal contribution term shows that there is no need for an agent to know the actions of the agents that do not have a sensing intersection with this. It makes the defined utility function local. After defining our game ingredients, now a state-based game \( \mathcal{Y} \) will be introduced. The following will show that our defined game is a state-based potential game.

**Lemma 3.1:** The coverage state-based game \( \mathcal{Y} := \langle N, A, U_{\text{conv}} \rangle \), where \( U_{\text{conv}} = \{U_j, j = 1, ..., N\} \), is an exact state-based potential game with the potential function

\[
\Phi(a(t), z(t)) = \sum_{j=1}^{N} \left( F(a_j(t), a_{-j}(t)) - F(a_j^0(t), a_{-j}(t)) \right) - \sum_{j=1}^{N} K_j(a_j^0(t))^2 - \sum_{j=1}^{N} K_j'(|a_j^0(t) - z_j(t)|),
\]

where \( z(t) = (z_1(t), ..., z_N(t)) \) is the game state.

**Proof:**
As shown in [9], a state-based potential game has to satisfy two conditions:
1. For any agent \( i = 1, ..., N \) and action \( a_i'(t) \in A_i \)
   \[
   \Phi(a_i'(t), a_{-i}(t), z(t)) - \Phi(a_i(t), z(t)) = U_i(a_i'(t), a_{-i}(t), z(t)) - U_i(a_i(t), z(t)),
   \]
   (3)
2. For any state \( z'(t) \) in the support of \( (a(t), z(t)) \), the inequality \( \Phi(a(t), z'(t)) \geq \Phi(a(t), z(t)) \) holds.

We first verify condition 1). According to (1) and (2), we have

\[
\Phi(a_i'(t), a_{-i}(t), z(t)) - \Phi(a_i(t), z(t)) = \sum_{j=1, j \neq i}^{N} \left( F(a_j(t), a_{-j}(t)) - F(a_j^0(t), a_{-j}(t)) \right)
- K_j(a_j^0(t))^2 - K_j'(|a_j^0(t) - z_j(t)|) + F(a_i'(t), a_{-i}(t))
- F(a_i(t), a_{-i}(t)) - K_i(a_i(t))^2 - K_i'(|a_i(t) - z_i(t)|)
+ \sum_{l=1}^{N} \left( - F(a_l(t), a_{-l}(t)) + F(a_l^0(t), a_{-l}(t)) \right)
+ (K_i(a_i'(t))^2 + K_i'(|a_i'(t) - z_i(t)|))
= F(a_i(t), a_{-i}(t)) - F(a_i(t), a_{-i}(t)) - K_i(|a_i(t) - z_i(t)|)
+ K_i'(|a_i'(t)^2 - (a_i'(t))^2) - |a_i'(t) - z_i(t)|)
= U_i(a_i(t), a_{-i}(t), z(t)) - U_i(a_i(t), z(t)),
\]

where in the second equality, the terms corresponding to \( a_j(t) \) and \( a_i(t) \) cancel each other except for agent \( i \) whose action is changed from \( a_i(t) \) to \( a_i'(t) \). Thus (4) shows that (3) holds.

We next verify condition 2). The fact that \( z'(t) \) is in the support of \( (a(t), z(t)) \) implies that \( z'(t) = a_i^0(t) \). Thus we have

\[
- \sum_{j=1}^{N} K_j(|a_j^0(t) - z_j(t)|) \leq - \sum_{j=1}^{N} K_j(|a_j^0(t) - z_j'(t)|) = 0.
\]

As a result it follows that \( \Phi(a(t), z'(t)) \geq \Phi(a(t), z(t)) \).

**B. Reinforcement Learning**

Converging to a Nash equilibrium requires distributed adaptation rules. Game theoretic reinforcement learning provides a starting point for the construction of iterative algorithms to reach a Nash equilibrium [10]. Binary log-linear learning is a modified version of the standard log-linear learning for potential games in which variant available action sets can be used by the agents. As mentioned in Section II the feasible actions that are available to an agent are restricted by the agent’s state in the game thus the available action set will be time-varying. In [11], the binary log-linear method has been analyzed. Theorem 5.1 in [11], shows that a potential game will converge to stochastically stable actions. These actions are the set of potential maximizers if all agents adhere to the binary log-linear learning where the following assumptions should be satisfied on the agents’ available sets.

1. Feasibility: For any agent \( i = 1, ..., N \) and any action pair \( a_i(0), a_i(m) \in A_i \), there exists a sequence of actions from \( a_i(0) \) to \( a_i(m) \) satisfying \( a_i(t) \in C_i(a_i(t-1)) \) for all \( t \in \{1, 2, ..., m\} \).
2. Reversibility: For any agent \( i = 1, ..., N \) and any action pair \( a_i'(t), a_i''(t) \in A_i \), \( a_i'(t) \in C_i(a_i'(t)) \leftarrow a_i''(t) \in C_i(a_i''(t)) \).

It is easy to check that in our problem set-up the above assumptions are satisfied. In this method, at each time \( t \), one agent is randomly selected and allowed to alter its action while all other agents repeat their actions, i.e., \( a_i(t) = a_i(t-1) \). The selected agent \( i \) chooses a trial action \( a_i'(t) \) uniformly randomly from the available action set \( C_i(a_i(t-1)) \). The player calculates the utility function for this trial action. Then agent \( i \) randomizes its action according to

\[
P_i^a(t-1)(t) = \frac{\exp(\frac{1}{\tau}U_i(a_i(t-1), z(t-1))) + \exp(\frac{1}{\tau}U_i(a_i'(t), a_{-i}(t-1), z(t)))}{\exp(\frac{1}{\tau}U_i(a_i(t-1), z(t-1))) + \exp(\frac{1}{\tau}U_i(a_i'(t), a_{-i}(t-1), z(t)))},
\]

\[
P_i^{a_i^0(t)}(t) = \frac{\exp(\frac{1}{\tau}U_i(a_i(t-1), z(t-1))) + \exp(\frac{1}{\tau}U_i(a_i'(t), a_{-i}(t-1), z(t)))}{\exp(\frac{1}{\tau}U_i(a_i(t-1), z(t-1))) + \exp(\frac{1}{\tau}U_i(a_i'(t), a_{-i}(t-1), z(t)))},
\]

\[
P_i^{a_i'(t)}(t) = 0, \forall a_i(t) \neq a_i'(t), a_i(t-1).
\]

where \( P_i^a(t-1)(t) \) denotes the probability of choosing action \( a_i(t) \) at time \( t \) and \( \tau > 0 \) is a coefficient that determines how likely agent \( i \) is to choose a suboptimal action. In other words, the agent uses a distribution to randomly select between its previous action, \( a_i(t-1) \), and an alternative action \( a_i'(t) \), where \( a_i'(t) \) is uniformly randomly selected from the set \( C_i(a_i(t-1)) \).

**IV. GAUSSIAN MIXTURE MODEL ESTIMATION**

In this section, we introduce an estimation scheme to overcome the limitations discussed in Section I. To apply the binary log-linear method (5), the agents have to calculate their future utility functions. However, the lack of the prior knowledge about the worth of the uncovered area makes this
impractical. In our new set-up, the agents will keep the worth of sensed regions in their memories. This information will be used to estimate the worth of uncovered regions. The GMM is chosen as a distribution model of the worth of the area. The GMM is a parametric probability density function represented as a weighted sum of Gaussian component densities defined as $\sum_{k=1}^{M} w_k g(x|\mu_k, \Sigma_k)$ and

$$g(x|\mu_i, \Sigma_i) = \frac{1}{2\pi|\Sigma_i|^{\frac{1}{2}}} \exp \left[-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)\right],$$

where $x$ is a 2-dimensional location data vector, $M$ is the number of Gaussian functions in the GMM, $w_i$ is the $i$th weighting coefficient, and $g(x|\mu_i, \Sigma_i)$ is a 2-variate Gaussian function with mean vector $\mu_i$ and covariance matrix $\Sigma_i$. The weights have to satisfy the constraint that $\sum_{k=1}^{M} w_k = 1$.

In this paper, the GMM is parametrized by $3M$ components given by $\lambda = \{w_j, \mu_j, \sigma_j | j = 1, 2, ..., M\}$, where $w_j, \mu_j$ and $\sigma_j$ are the weighting coefficient, mean and covariance corresponding to the $j$th Gaussian of the GMM, respectively. Agent $i$ has its own estimate of $\lambda$ denote by $\hat{\lambda}_i = \{\hat{w}_j, \hat{\mu}_j, \hat{\sigma}_j | j = 1, 2, ..., M\}$, where $\hat{w}_j, \hat{\mu}_j$ and $\hat{\sigma}_j$ are the estimations of $w_j, \mu_j$ and $\sigma_j$ by agent $i$, respectively. One of the most accurate and well-established method for estimation is the ML. Agent $i$ has the observation $O_i t = a^i_\delta(t)$ at time step $t$. The sequence of these observations, from the step $1$ to $T$, will be used as a training vector $O^i = \{O^i_1, O^i_2, ..., O^i_T\}$. Given the training vector $O^i$, we wish to find the estimate $\hat{\lambda}_i$ that maximizes the probability of the occurrence of the observation sequence vector $O^i$ denoted by $p(O^i|\hat{\lambda}_i)$. The ML estimation can be written as $\max_{\lambda_i} p(O^i|\lambda_i) = \max_{\lambda_i} \prod_{t=1}^{T} p(O^i_t|\lambda_i)$. Although it is a non-linear function of the parameter $\lambda_i$ and direct maximization is not possible, the expectation-maximization (EM) has been introduced in [12] to obtain an iterative algorithm. Using the same EM approach, the GMM parameters can be estimated as

$$\hat{w}_j = \frac{1}{T} \sum_{t=1}^{T} Pr^i(j|O^i_t, \hat{\lambda}_i),$$

$$\hat{\mu}_j = \frac{1}{\sum_{t=1}^{T} Pr^i(j|O^i_t, \hat{\lambda}_i)} \sum_{t=1}^{T} Pr^i(j|O^i_t, \hat{\lambda}_i) O^i_t + \hat{\mu}_j,$$

$$\hat{\sigma}_j^2 = \frac{1}{\sum_{t=1}^{T} Pr^i(j|O^i_t, \hat{\lambda}_i)} \sum_{t=1}^{T} Pr^i(j|O^i_t, \hat{\lambda}_i) \sum_{t=1}^{T} Pr^i(j|O^i_t, \hat{\lambda}_i) O^i_t - \hat{\mu}_j \hat{\mu}_j + \hat{\sigma}_j^2,$$

$$Pr^i(j|O^i_t, \hat{\lambda}_i) = \frac{\hat{w}_j g(O^i_t|\hat{\mu}_j, \hat{\sigma}_j^2)}{\sum_{k=1}^{M} \hat{w}_k g(O^i_t|\hat{\mu}_k, \hat{\sigma}_k^2)},$$

where $Pr^i(j|O^i_t, \hat{\lambda}_i)$ is a posteriori probability of the observation $O^i_t$ for the $j$th Gaussian of agent $i$.

The problem encountered using algorithm (6) is that the EM method optimizes the likelihood of the parameters given the observed location without taking the worth of that location into account. Thus a minor change has to be made to the ML estimation algorithm (6). The same problem has been pointed out in the image processing literature [13]. We use a modified version of the solution introduced in [13] to repeat the ML algorithm $m$ times in worthy locations with $m$ chosen as

$$m = \begin{cases} 1 + \gamma \text{round}(\frac{f_\gamma}{f_\text{mode}}) & f_\gamma \geq f_\text{mode} \\ 1 & \text{otherwise} \end{cases}$$

where $f_\text{mode}$ and $\gamma$ are a threshold and a correction factor, respectively. Here the locations with the worth lower than the threshold $f_\text{mode}$ are used only once. It is apparent that if the correction factor, $\gamma$, is increased, a region with a higher worth will be repeated more often.

In our solution set-up, the agents use the estimated parameter $\lambda^i$ to calculate the estimate of $f_q$ at each iteration. As mentioned in Section II, $f_q$ is assumed stationary but the estimate of $f_q$ is no longer stationary. Hence the game parameters are not stationary any more. However, a standard assumption in game theory is that the game parameters are stationary. In [14], the case of slow variations of the game parameters is investigated. Corollary 4 in [14] shows that using the binary log-linear learning the probability of converging to a Nash equilibrium will be greater than $1 - \delta_1$ if for any $\delta_1 > 0$, there exist $\delta_2 > 0$ and $\delta_3 > 0$ such that $|\lambda(t + 1) - \lambda(t)| < \delta_2$ and $|\lambda - \lambda(t)| < \delta_3$, where $\delta_2$ and $\delta_3$ are the bound on the changing rate of $\lambda$ and the bound on the estimation error of $\lambda$, respectively. Under the assumption of a stationary environment we will have $|\lambda(t + 1) - \lambda(t)| = 0$, thus there exists a positive bound $\delta_2$. In addition, by more observations agents will have a better knowledge of the area and the agents’ estimation error $|\lambda - \lambda(t)|$ decreases. Hence there exists a bound $\delta_3$ on the estimation error. Thus the conditions are satisfied and the agents will converge to a Nash equilibrium stochastically.

V. MUTUAL INFORMATION

In this section, we add a mutual information term to each agent’s utility function in order to find an action with a more informative observation. Sending rich observation data can improve the estimation performance. In order to select informative observations, the entropy criterion has been used in information theory and applied mathematics context [15]. In information theory, the conditional entropy quantifies the amount of information needed to describe the outcome of a random variable $Y$ given the value of variable $X$, which is written as $H(Y|X)$. Here our goal is to reduce the uncertainty of the unseen area by finding more informative locations for future observations. In [16], the mutual information criterion has been proposed for observation selections where it is a measure of the mutual independence of two variables. It is shown that maximizing the mutual information between an observed and an unobserved area is more effective than the conditional entropy to reduce the uncertainty. The objective function can be written as

$$I(X^O_t : X_{Q^O_t}) = H(X_{Q^O_t} | X^O_t) - H(X_{Q^O_t} | X_{Q^O_t}),$$

where $I$, $X^i$, $X_{Q^O_t}$ and $X_{Q^O_t}$ are the mutual information, and the observed and unobserved variables corresponding to agent $i$, respectively. To compute the mutual information two entropy functions have to be calculated in (8). Fortunately, for Gaussian processes there exists a close form to compute the conditional entropy which is given by

$$H(X_{Q^O_t} | X^O_t) = \frac{1}{2} \log(2\pi e \sigma_{X_{Q^O_t} | X^O_t}^2),$$

where $\epsilon$ is the Euler number and $\sigma_{X_{Q^O_t} | X^O_t}^2$ is the variance of the conditional distribution of $X_{Q^O_t}$ given $X^O_t$. The covariance $\sigma_{X_{Q^O_t} | X^O_t}^2$ calculated as
\[ \sigma^2_{X_{Q \setminus O^t} | X_{O^t}} = K(X_{Q \setminus O^t}, X_{Q \setminus O^t}) - \sum_{X_{Q \setminus O^t}, X_{O^t}}^{n} \sum_{X_{O^t}, X_{Q \setminus O^t}}^{n} \Sigma_{X_{Q \setminus O^t}, X_{O^t}}, \]

where \(\Sigma_{X_{Q \setminus O^t}, X_{O^t}}\) is a covariance matrix each of whose entries is a function of the kernel function \(K(a, b)\) for \(a \in X_{Q \setminus O^t}, b \in X_{O^t}\), and \(\Sigma_{X_{Q \setminus O^t}, X_{O^t}} = \sum_{X_{Q \setminus O^t}, X_{O^t}}^{n}\). The kernel function determines the correlation of the observations where for spatial phenomena the correlation between observations depends on the distance of their locations. One of the most frequently used kernel functions is \(K(a, b) = \exp(-h\|a-b\|)\), where \(\|a-b\|\) is the distance between the locations \(a\) and \(b\) and \(h\) is a constant. As can be seen, the variance does not depend on the observed values, so the mutual information can be calculated ahead of an agent’s new action. Before adding the mutual information term to the utility function defined in (1), it should be discussed whether the modified game will remain a state-based potential game or not.

Lemma 3.1: The coverage game \(\Gamma\) for \(U_i(a(t), z(t)) = H(X_{Q \setminus O^t}) - H(X_{Q \setminus O^t} | X_{O^t})\) is an exact state-based potential game with the potential function

\[ \Phi(a(t), z(t)) = \sum_{j=1}^{N} \left( H(X_{Q \setminus O^t}) - H(X_{Q \setminus O^t} | X_{O^t}) \right) \]  

Proof: Note that

\[ \Phi(a_i(t), a_{-i}(t), z(t)) - \Phi(a(t), z(t)) \]

\[ = \sum_{j=1, j \neq i}^{N} \left( H(X_{Q \setminus O^t}) - H(X_{Q \setminus O^t} | X_{O^t}) \right) + H(X_{Q \setminus O^t}) \]

\[ - H(X_{Q \setminus O^t} | X_{O^t}) - \sum_{i=1}^{N} \left( H(X_{Q \setminus O^t}) - H(X_{Q \setminus O^t} | X_{O^t}) \right) \]

\[ = H(X_{Q \setminus O^t}) - H(X_{Q \setminus O^t} | X_{O^t}) - H(X_{Q \setminus O^t}) + H(X_{Q \setminus O^t} | X_{O^t}) \]

\[ = U_i(a_i(t), a_{-i}(t), z(t)) - U_i(a(t), z(t)). \]

Hence the first condition for a state-based potential game is satisfied. As can be seen, the state \(z(t)\) does not appear in the utility function here, so there is no need to check the second condition as discussed in Lemma 3.1. ■

Now, we define a new utility function as

\[ U_i(a(t), z(t)) = F(a_i(t), a_{-i}(t)) - F(a_i^0(t), a_{-i}(t)) \]

\[ - K_i(a_i^0(t))^2 - K_i([a_i^0(t) - z_i(t)]) \]

\[ + \eta(t)(H(X_{Q \setminus O^t}) - H(X_{Q \setminus O^t} | X_{O^t})). \]  

where \(\eta(t)\) is a time varying coefficient which adjusts the importance of the mutual information term in comparison with the sensing optimization part. The search of \(\eta(t)\) follows the following principle. Starting the search, the agents do not have a good estimate of the area and gathering proper data serves as an important role for improved sensing. However, by having a more accurate estimate, the weight of the sensing part has to be increased over time because we need to put more effort on finding the best configuration of the MSN.

Remark 5.1: It is easy to show that the game \(\Gamma\) is an exact state-based potential game for the utility function (11), because the sum of (2) and (10) is used as a potential function.

Remark 5.2: The mutual information term can be used when \(M\) in Section IV is selected as 1. As shown in (9), the agents are able to calculate the conditional entropy before sensing an area. However, calculating the conditional entropy for the GMM with \(M > 1\), requires knowing the future weighting coefficient \(\hat{w}^t_j\) that is not available. Thus the designer has two choices: 1) The utility function (1) can be used without the mutual information term while having the freedom of selecting \(M \geq 1\). 2) Employ the utility function (11) with the mutual information term introduced while fixing \(M = 1\). Due to the estimation of the worth of the uncovered regions using historical observations, both choices can overcome certain limitations in the existing literature and offer better performance than the existing methods.

VI. SIMULATION AND DISCUSSION

Our results guarantee the stochastic stability of the Nash equilibrium. Although the theoretical results do not provide any estimate of the convergence rate, the simulation results in this section show an acceptable convergence rate in practice. Consider a \(30 \times 30\) square in which each grid is \(1 \times 1\) where a group of five mobile sensors \((N = 5)\) are deployed in this area and each sensor can choose its \(a_i^t\) from the set \(\{0, 1, 2, 3\}\), which means \(r_{min} = 0, r_{max} = 3\) and \(K_{i, mode} = 6\). We let \(M = 1\) and use the Gaussian distribution as an estimation model. A different distribution has been used to create \(f_q\) in order to clarify the ability of the proposed approach. Here, the sum of two Copula distributions from the Frank family [17] with the scalar parameters \(\rho = -0.2, 15\) is used to generate \(f_q\) for all \(q \in Q\). Fig. 1 shows the distribution worth of the area. It can be seen that there is a line that has more worth. It bears mentioning that the agents do not know the worth of the area before sensing them.

The mobile sensors’ initial locations were chosen completely randomly. The agents employ the defined approach using (5), (6), and (7) with the utility function (11). They change their configurations in order to maximize their utility function. The simulation parameters are chosen as \(K_i = 3 \times 10^{-5}, K_i^t = 3 \times 10^{-4}, i = 1, 2, \ldots, N, \tau = 5 \times 10^{-3}, f_{mode} = 10^{-5}, \gamma = 0.1\) and \(\eta(t) = \frac{0.001t+1}{100-t}\). Fig. 2 shows the final locations of the mobile sensors where they are aligned with the regions corresponding to the highest worth. The \(5^{th}\) agent shown as a green circle is far from the worthy region. Thus the agent has decided to turn off its own sensor to save power. Changing the actions of the sensing neighbors would create some space in more worthy area for agent 5. In such a situation the sensor will change its action and start sensing the area again.

In the next part of our simulation, we investigate the
effect of using the mutual information in the utility function and compare our method with an existing method in the literature. First, two simulations for \( \eta(t) = \frac{0.001t + 1}{100t^2 + 1} \) and \( \eta(t) = 0 \) are compared with each other where the initial locations are the same for both methods. As it can be seen in Fig. 3, when the mutual information is used, almost 70 percent of the area’s worth is covered by the MSN after only 1700 time steps. However, without using the mutual information term, the MSN needs much more time to converge to its Nash equilibrium. Fig. 3 clarifies that using informative data to feed the estimation algorithm (6) will help the system to cover the area faster. Second, we compare our method with the distributed inhomogeneous synchronous coverage learning (DISCL) method introduced in [7]. The DISCL method’s best performance is obtained with an exploration rate \( \left( \frac{1}{\sqrt{2\pi t}} \right)^2 \) and hence this exploration rate is used for the DISCL method in simulation. The worth of the covered area using the DISCL method is illustrated in Fig. 3. The simulation results show that our methods with or without using the mutual information outperform the DISCL method. This illustrates that estimating the worth of the area using the knowledge of previous observations can improve the MSN performance. Sudden changes in the worth of the covered area can be seen in the case using the DISCL method. The reason is that the agents choose their future temporary actions uniformly from the set of available actions without considering the utility value of these actions. As a result, some of these actions might not be so valuable.

**Remark 6.1:** As mentioned in Remark 5.2, using the utility function (1) allows us to select \( M \geq 1 \). However, our simulations illustrate that for \( M > 1 \) the weights \( \hat{w}^j \) become very peaked after only a limited number of observations. As a result, the estimated distribution will be one Gaussian instead of a mixture of Gaussians. Such a result occurs because the agents maximize their utility functions while they are observing the area at the same time. Hence the agent’s observations will be focused more on one of the Gaussians in the GMM. This result shows that fixing \( M = 1 \) does not degrade the performance of our method.

**VII. CONCLUSIONS**

In this paper, we have discussed the sensor coverage problem using a MSN where the mobile sensor agents’ task is to reposition themselves from their random initial locations to their optimal configuration. A utility function is defined based on the marginal contribution of each agent on sensing the area while considering the energy consumption of the MSN. A state-based potential game was introduced using the defined utility function. Then each agent applied binary log-linear learning to update its own action at each iteration. Taking into account the fact that an agent needs the knowledge about the worth of its action set to use this learning algorithm, an estimation scheme has been introduced to predict the worth of uncovered regions using the agents’ previous observations. The ML method has been employed to estimate the unknown parameters of the GMM.

In the last part of this paper, the mutual information term has been added to the agent’s utility function in order to find an action which creates more informative observation.

**REFERENCES**


