Dynamic Area Coverage using Faulty Multi-Agent Swarms

Ke Cheng, Prithviraj Dasgupta
Computer Science Department
University of Nebraska, Omaha, NE 68182, USA
E-mail: {kcheng, pdasgupta}@mail.unomaha.edu

Abstract

We consider the problem of distributed coverage of an unknown two-dimensional environment using a swarm of mobile mini-robots. In contrast to previous approaches for robotic area coverage, we assume that each robot(agent) 1 in our system is limited in its communication range and memory capacity. Agents are also susceptible to sensor noise while communicating with other agents, and, can be subject to transient or permanent failures. The objective of the agents is to cover the entire environment while reducing the coverage time and the redundancy in the area covered by different agents. First, we describe our distributed coverage algorithm where each agent uses a local heuristic based on Manhattan distances and the information gained from other agents at each step to decide on its next action(movement). We then describe and analyze the fault model of our agents and show that the local heuristic used by the agents deteriorate linearly as the communication noise increases. Finally, we verify the performance of our system empirically within a simulated environment and show that the system is able to scale efficiently in the number of robots and size of the environment, and, determine the effect of communication faults and robot failures on the system performance.

Keywords: Swarming, distributed area coverage, fault, failure.

1 Introduction

Over the past few years, swarm-based multi-agent systems have emerged as an attractive paradigm for designing large scale distributed systems consisting of behaviorally simple swarm units(agents). Swarm-based multi-agent systems are particularly attractive because they are inherently distributed, and robust against failure of a significant number of the swarm units(agents). However, in the absence of any centralized control it is often challenging to monitor the system’s global behavior, and ensure that the local agent rules translate efficiently into the global objective. In this paper, we concentrate on the problem of distributed coverage of an unknown grid-like environment using a swarm of multi-robots(agents) in the presence of agent faults and failures. Each agent is limited in its communication range and memory capacity. Moreover, the communication between agents is susceptible to sensor noise, and each agent can fail transiently or permanently. The global objective in the system is to have every cell in the grid-environment visited by at least one agent while 1) reducing the coverage time, and 2) reducing the redundancy in the area covered by the agents. Our analytical and experimental results show that the local agent algorithms described in the paper translate to efficient, scalable and fault-tolerant coverage of the environment under different operational constraints, even in the presence of agent faults and failures.

Distributed coverage of unknown environments using multiple mini-robots is an essential task in different applications including terrain mapping, search and rescue, aerial reconnaissance, and space exploration. Previous research in the area of distributed coverage using robot swarms makes some limiting assumptions such as using complex negotiation-based protocols that do not consider any limits on the communication range between robots[8], or, perform a tradeoff between communication range and solution quality[5]. On the other hand, heuristics-based strategies such as LRTA* and node counting used in[6] assume that robots can exchange coverage information about the entire environment with each other through trails that are deposited within the environment. However, in many scenarios, the environment might not be conducive to depositing trail information (e.g., aerial robots, or, adversaries erasing trails), or, robots might not have appropriate sensors to directly sense trail information from the environment. In such a scenario, robots could possibly exchange coverage information maps with each other to decide their actions. However, with limited communication range and limited on-board memory of agents, exchanging local coverage maps between all agents might be prohibitively expensive and infeasible. Therefore, it makes sense to address
the problem of distributed coverage of an unknown environment using multi-agent swarms comprised of agents that are limited in communication and memory resources.

Our paper makes two major contributions towards dynamic coverage of unknown environments by multi-agent swarms. First, we describe a probabilistic hill-climbing algorithm that is used locally by each agent to navigate itself in the environment. This algorithm uses a simple local heuristic based on Manhattan distances and the information gain perceived by an agent at each step to decide its next action. We then describe and analyze a fault model for the communication between agents in our system and show that the heuristic deteriorates linearly with increase in the sensor noise. To analyze the global behavior of our system, we have verified the system’s performance empirically within a simulated environment under different operational constraints, including sensor noise and malfunctioning of agents. Our simulation results show that although area coverage degrades as the agent failures increase, the agents are still able to cover the area, albeit taking a longer time, until every agent fails to perform about 10% of the actions it attempts. On the other hand, sensor noise causes a linear decrease in overall coverage time as reported in from the analysis of our fault model.

2 Model

We consider a set of $R$ agents in a two-dimensional, square, grid-like environment. Each agent has four degrees of freedom represented by its set of actions $\{up, down, left, right\}$. All agents enter into the environment from the same location. Each agent is equipped with a radio transceiver that allows it to exchange information with other agents within a finite radius from its current location. The parameters used by our model are:

- $R$ Set of agents
- $Ac$ Set of actions of an agent, $Ac = \{up, left, down, right\}$
- $\mu$ Radius for coverage map of an agent
- $\mu_r^t$ Coverage map of agent $r \in R$ at time $t$
- $\xi_r^t$ Communication range of an agent
- $\xi_r$ Communication map for agent $r \in R$ at time $t$
- $Ac_r$ Set of joint actions of $R$ agents, $Ac = \times_R Ac$
- $ac_r^t$ $i$-th joint action of agent, $ac_r^t \in Ac$
- $loc_r^t$ Location of agent $r$ at time $t$
- $V_r^{Thr}$ Threshold value for pheromone in a cell

The objective of our coverage algorithm is to enable the agents to completely cover the environment while reducing coverage time and the redundancy in coverage. To achieve this, each agent maintains two maps within itself: (a) Communication Map. The communication map $\xi_r^t$ for agent $r$ at time $t$ contains the locations of other agents that are within the communication range $\xi$ of agent $r$. That is, $\xi_r^t = \{loc_r^t\}$ where, $r' \in R - \{r\}$ and $loc_r^t \in \{loc_r^t - \xi, loc_r^t + \xi\}$.

(b) Coverage Map. When an agent $r \in R$ visits a cell it marks it as visited or covered and records the information within itself in a local coverage map $\mu_r^t$ at time $t$. We assume that the coverage map of an agent extends over a radius $\mu$ from the agent’s current location. The coverage map of agent $r$ is given by the set of cells represented by their two-dimensional coordinates $(x, y)$. Following [6], each cell contains the number of times it has been visited by agents. That is, $\mu_r^t = \{\mu_{x,y}^t\}$, where $\mu_{x,y} \in [0, \infty]$ and $(x, y) \in \{loc_r^t - \mu, loc_r^t + \mu\}$.

The value of a cell in an agent’s coverage map could be considered similar to the deposit of a synthetic chemical substance called pheromone by agents at that cell. Agents update the pheromone value in a cell when they visit the cell. We also allow the pheromone information deposited by an agent to evaporate over time to model the volatility of coverage information in the environment (that is, how of-

![Algorithm](image)

**Figure 1.** Algorithm used by an agent to select its next action with limited communication range and memory constraints on each agent.
ten agents should revisit covered areas). At each time step, an agent exchanges its own local coverage map $\mu^r_i$ with all other agents within its communication map $\xi^r_i$. Each agent fuses the information in the coverage map received from other agents before selecting its action for the next time step.

### 3 Coverage Algorithms

In our model, when an agent selects an action while using size-limited communication and coverage maps, its coverage information could improve from information fusion, only if taking the action brings the agent within communication range of at least one more agent. On the other hand, moving closer to other agents to improve the coverage map of an agent through information fusion can limit the dispersion of the agents across the environment and adversely affect the coverage rate. Clearly, at each step an agent faces a tradeoff between the coverage information it can gain from other agents and the quality of coverage achieved by the agents. To address this issue, each agent selects its action probabilistically based on a linear weighted sum of the number of agents with which its distance decreases and the number of agents with which its distance increases and the potential information gain from the action, as shown in Figure 1. In the selectAction algorithm, each agent $r$ determines the set of other agents $R_{r,ac}$ that would enter into its coverage map by taking action $ac \in Ac$. If the coverage map of agent $r$ is empty for all possible actions, agent $r$ cannot gain any coverage information from other agents by taking any of its actions. It then uses the selectActionECM algorithm described below and illustrated in Figure 2, so that the agents disperse away from each other to cover the environment. On the other hand, if the set $R_{r,ac}$ is non-empty for some subset of actions of agent $r$, it calculates the probability associated with that subset of actions as a linear weighted sum between agent dispersion and information gain.

In the selectActionECM method (Figure 2), each agent determines the set of possible joint actions with other agents within its communication range which will take each agent to the cell with the minimum value of $\mu_{x,y}$ (for each agent’s neighboring cells). Each agent then calculates the sum of the Manhattan distances between other agents for each joint action. The normalized values of these combined Manhattan distances are then used to probabilistically select the next possible action at each agent. Using this technique, the expected action of an agent $r$ is given by: $E(ac_r) = arg\max_{ac_r} \max P_{\xi^r_i} | ac_r$. The Manhattan distance based heuristic ensures that agents have the highest probability of selecting the joint action that maximizes their combined Manhattan distances with each other. This results in agents dispersing away from each other in successive steps to explore the environment. In certain scenarios, depending on the shape of the environment, a dispersion-only coverage strategy between the agents can lead to regions of unvisited cells at the center of the environment. This is addressed in our algorithm, by the probabilistic nature of the action selection by agents and allows them to select a joint action that does not correspond to the maximum combined Manhattan distance with a non-zero probability. The method used in Figure 2 executes the action $ac \in Ac$ for agent $r \in R$ at its current location $loc^r_t$ at time $t$.

A challenging issue faced by the agents with limited sized coverage maps is that each agent has to discard its own coverage history as well as fused coverage information obtained from other agents with every move, corresponding to the cells that exit its coverage radius $\mu$ because of the movement. This increases the redundancy in coverage because

---

2For fusing coverage maps of multiple agents, we add the node count value of overlapping cells in the coverage map of the agents.
each agent decides its action based only on the coverage information of its immediate neighborhood obtained from its coverage map. To address this problem, we enable each agent to infer the possible future movements of other agents in its vicinity by observing the direction of pheromone gradient in its coverage map. The pheromone gradient of the agent’s coverage map indicates the direction of movement of agents in the coverage map. Each agent could use this pheromone gradient information to infer the direction of other agents in its vicinity and select an action that balances the information gain and dispersion from other agents. We have used three different strategies for pheromone deposit and update in the coverage map of agents to model different degrees of information volatility and pheromone gradient in the coverage map of an agent, as described in Figure 3.

4 Agents with Communication Faults

The local heuristic $h_{MD}$ used by an agent in our algorithm relies extensively on the correct location coordinates of other agents located within its coverage and communication maps. However, most sensors, especially GPS sensors that are commonly used to obtain location coordinates, are characterized by noisy readings. Therefore, it makes sense to analyze the operation of the agents’ local algorithm as well as the global system behavior in the presence of sensor noise. To model faulty communication of coordinates in our system, we assume that sensor noise is distributed as a uniform, zero mean distribution. Let $\delta$ denote the maximum number of units of error along each coordinate that can be reported in a sensor reading. As before, we consider $|R|$ agents in a two-dimensional environment. Because we consider the sensor noise as zero-mean, uniform distribution a mean value analysis of the effect of the noise on $h_{MD}$ is likely to return a zero value. Therefore, we quantify the effect of sensor noise in terms of the standard deviation in $h_{MD}$.

**Proposition.** The standard deviation in $h_{MD}$ due to the presence of sensor noise with a maximum of $\delta$ units in each dimension, distributed uniformly with a zero mean has an upper bound of $O(\delta)$.

**Proof.** For simplifying our analysis, we assume that all agents are within communication range of each other. Then, the number of possible sensor readings that can be received by agent $r \in R$ is given by $(2\delta + 1)^2 |R|$. Out of this set of readings, the ones that have the same amount of error with opposite signs along the $x$ (latitude) and $y$ (longitude) coordinates give zero error in $h_{MD}$, because the Manhattan distance heuristic adds the distances along the two coordinate axes. Therefore, the total number of values of $h_{MD}$ that have zero error is given by $\delta |R|$. Correspondingly, the number of values with $e$ units of error in $h_{MD}$ is given by $[(2\delta + 1) - |e|] |R|$, where $1 \leq e \leq 2\delta$. Since errors are distributed uniformly with zero mean, the standard deviation in the error in calculating $h_{MD}$ is given by:

$$SD_{err} = \sqrt{\frac{2}{(2\delta + 1)^2 |R|} \sum_{e=1}^{2\delta} [(2\delta + 1) - |e|] |R| \times e^2}$$

or, $SD_{err} = O(\sqrt{\frac{2(2\delta + 1)^2 |R|}{(2\delta + 1)^2 |R|}}) = O(\sqrt{\delta}) = O(\delta)$.

In our system, we have designed the agent rules at the local level, while the global behavior of the system is manifest by the interactions between the agents. In the next section, we analyze the global behavior of our system in terms of the two metrics coverage time and coverage redundancy under different operational constraints, including faults and failures of the agents.

5 Experimental Results

For our experiments, each agent is simulated as a generic DifferentialWheels robot whose speed and direction are controlled by changing the relative rotation speed between the two wheels. The maximum speed of each wheel was set to 1 meter/time unit. Robots measure 1 meter $\times$ 1 meter. Each robot has the following sensors: (1) GPS: x, z location and heading. (2) Distance sensor for obstacle avoidance. (3) Long-range radio transmitter and receiver for sending and receiving over channel 0 with a maximum range of 11 meter. The environment is in the shape of a square measuring $100 \times 100$ meter$^2$ and 5 agents, each with a coverage map radius of 5 meter are used, unless otherwise stated. In all experiments, except the first set, we consider that node
count of cells decreases (pheromone evaporates) over time and agents consider the pheromone gradient or trails while selecting their next action. All results are averaged over 20 simulation runs.

In our first set of experiments shown in Figure 4, we observe the coverage of the environment with different coverage strategies. Each agent has a coverage map radius of 5 meter (denoted by $MR_5$ in the figure caption). One of the strategies tested allows agents to select a random action at each step. For the strategy labeled Binary, the value in each cell is considered as a binary number instead of a real value. In the strategies labeled Inc, the pheromone (node count) information does not decrease over time. The caption Trail denotes that the agents use trail gradient information in their coverage map to select their next action, as described in Figure 3. As shown in Figure 4, the strategies that use the trail information obtain the highest coverage because the agents decide their actions by combining the coverage information from their own coverage maps, as well as the possible directions of movement of nearby agents estimated from the pheromone gradient in their coverage map. The strategies that use node counting without considering trail information perform slightly worse because each agent does not consider the possible locations of other agents while taking its action. The strategy with binary node counting performs poorly because binary node counting can capture less coverage information than real valued node counting. Finally, and quite expectedly, the random strategy performs very poorly because of the total absence of coordination among the agents.

In our second set of experiments, we analyzed the effect of changing the number of agents in the environment. Quite evidently, as the number of agents increases the coverage quality improves as long as the environment does not change. To further quantify the effect of the number of agents on the overall solution quality, we analyzed the redundancy in coverage in the environment by observing the number of cells that were revisited a specific number of times by agents. The results are shown in Figure 5. Quite interestingly, we observe that as the number of agents increases from 5 to 20, the peak of the redundancy curve shifts rightwards. Although this result simply indicates that as the number of agents increases while keeping the environment unchanged, the redundancy in coverage increases, these curves offer the useful insight of determining the optimal number of agents that result in the least redundancy for a given environment.

In our third set of experiments, we once again observe the variation in coverage redundancy while changing the size of the coverage map radius of each agent from 5 meter to 11 meter. As shown in Figure 6, as the size of the coverage map increases, each agent has more coverage in-
formation about its immediate neighborhood and is able to select its actions more efficiently to reduce its coverage redundancy.

In our next set of experiments, we analyze the effect of agent communication faults on the coverage time and coverage redundancy in the system. For the communication fault model, we have followed the sensor noise model described in [12] which is given by \( y(n) = x(n) + a(0.0015N(0,1) + 0.0012cos(2\pi n / r_w) + \phi_0) \). This models approximates sensor noise as a combination of a high frequency, zero mean, white Gaussian noise and a low frequency cosine wave with random period and random initial phase. For our simulation purposes, we have assumed, \( a = 100, \ r_w = 1600 \) and \( \phi_0 = 0 \). Figure 7 shows the coverage of the environment using with and without the sensor noise. In the presence of sensor noise, agents obtain about 35% less coverage than with noiseless communication, over the same number of timesteps. As observed in our mathematical analysis of the sensor noise model, the effect of sensor noise decreases the coverage linearly. A possible explanation of this behavior is offered by the coverage redundancy graphs with and without sensor noise shown in Figure 8. With sensor noise, the redundancy in coverage decreases. Therefore, although fewer cells get covered with sensor noise, many of the uncovered cells are those that were being covered redundantly when there was no noise.

In our last set of experiments, we measured the robustness of our system when some of agents failed and recovered at different time steps. During each time step, each agent might fail with a given failure probability and subsequently recover with a given recovery probability. Figure 9 illustrates the percentage of the coverage with 5 agents while keeping the failure probability constant at 0.01. The recovery probability of each agent is varied between 0.01 and 0.10. With increased recovery probability, the percentage of the coverage increased. Although some curves intersected possibly due to sensor noise, higher recovery probabilities got better coverage. Figure 10 illustrates the results of a similar experiment where we kept the recovery probability constant at 0.10 and varied the failure probability.
between 0.01 and 0.10. Interestingly, we observe that the agent failure probability has a significantly more detrimental effect on the coverage than the agent recovery probability. Also, in Figure 10, the two non-monotonically increasing curves near the failure probability of 0.10 show that that a high agent failure probability not only results in worse coverage, but fails to keep up with the volatility of coverage information (pheromone evaporation) in the system, causing coverage to reduce over time and rendering the system unstable. In both these cases, the redundancy in coverage decreased as the respective probabilities increased, although a more distinct reduction in redundancy was obtained over the various recovery probabilities. 4

6 Related Work

Distributed coverage of unknown environments using a group of robots has been an important area of research in multi-robot coordination over the last decade. More recently, various strategies for robot coordination with limited communication for area-coverage tasks are described in [3, 4, 7]. However, no information about memory constraints and communication ranges of robots used for the experiments are provided in these papers. A grid-map based decentralized approach is been used in [5] where the environment is divided into convex polygons. [10] describes extensions of a single robot coverage algorithm that uses cellular decomposition to multiple robots with limited communication. [9] describes different coverage strategies for mobile robot swarms under no communication. In [2], the authors describe a coordination algorithm for multiple robots for exploration tasks using a probabilistic utility-based approach that dynamically adjusts the utility of an explored area based on the number of robots observing that area. Complementary to the work described in this paper, [11] describes a coverage map based algorithm where robots dynamically update the probability of coverage status of a region using a Bayesian model. Robots subsequently use a closest location and/or an information gain based heuristic to plan their motion. [1] describe swarm-based algorithms with theoretical bounds for a floor cleaning application. The work described in this paper is complementary to these approaches.

7 Discussions and Conclusion

In this paper, we have described the effect of sensor noise and transient faults on a swarmed system of mini-robots with limited communication and memory for dynamic coverage of an unknown environment. Comparing our strategy with other area coverage strategies is slightly challenging because most other strategies are tailored towards specific environments, and include limiting assumptions such as unrestricted communication and/or unconstrained information exchange between agents. Nevertheless, adapting the fiducial strategy used by robots for area coverage [9] to our environment showed that our algorithm performs roughly 5 – 10% better. Using a single shared memory that all robots can access to get global coverage information[6] evidently performs better by about 30% than our scenario, where each robot, separately, stores the map of the entire environment within itself, due to the communication overhead involved in communicating the coverage map at each step between all robots. However, we emphasize the fact that using shared memory to communicate between robots is contrary to our main objective of achieving fully distributed coverage using the robot swarm. Currently, we are extending our model in several directions using market-based techniques for robot coordination and porting our algorithms to the e-puck platform. We believe addressing these issues will solve some of the challenges in multi-agent coordination of robot swarms with memory and communication constraints for different problems.

References