Supervisory Control for Cost-Effective Redistribution of Robotic Swarms

Ruikun Luo
Department of Mechaincal Engineering
College of Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: luoruikun1989@gmail.com

Nilanjan Chakraborty
Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: nilanjan@cs.cmu.edu

Katia Sycara
Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: katia@cs.cmu.edu

Abstract—**Dynamic assignment and re-assignment of large number of simple and cheap robots across multiple sites is relevant to applications like autonomous survey, environmental monitoring and reconnaissance. In this paper, we present supervisory control laws for cost-effective (re)-distribution of a robotic swarm among multiple sites. We consider a robotic swarm consisting of tens to hundreds of simple robots with limited battery life and limited computation and communication capabilities. The robots have the capability to recognize the site that they are in and receive messages from a central supervisory controller, but they cannot communicate with other robots. There is a cost (e.g., energy, time) for the robots to move from one site to another. These limitations make the swarm hard to control to achieve the desired configurations. Our goal is to design control laws to move the robots from one site to another such that the overall cost of redistribution is minimized. This problem can be posed as an optimal control problem (which is hard to solve optimally), and has been studied to a limited extent in the literature when the cost objective is time. We consider the total energy consumed as the cost objective and present a linear programming based heuristic for computing a stochastic transition law for the robots to move between sites. We evaluate our method for different objectives and show through Monte Carlo simulations that our method outperforms other proposed methods in the literature for the objective of time as well as more general objectives (like total energy consumed).**

I. INTRODUCTION

Redistribution of a robotic swarm across multiple sites is relevant to applications like autonomous survey, environmental monitoring and reconnaissance[1]. It is practical to build the robotic swarm with large number of independent, simple and cheap robots with limited computation and communication capabilities. In this paper, we consider the redistribution problem of such robotic swarm among multiple sites. We consider the anonymous and homogeneous robots with limited communication capability. The robots also have the capabilities to recognize the sites and receive control signals from a central supervisory controller but cannot communicate with each other. We assume that the central supervisory controller can recognize the current distribution of the robotic swarm among these sites. It is difficult to control such robotic swarm with anonymous robots moving among multiple sites to achieve the desired configuration with minimum cost (e.g., energy, time). In this paper, we present supervisory control laws for cost-effective redistribution of such robotic swarm among multiple sites.

As the robots are anonymous, we cannot control the robotic swarm deterministically. We have to design the stochastic control laws for redistribution of the robotic swarm accross multiple sites. Some of the previous literature considers that all the robots will stop moving when the desired configuration is achieved [2], [9], [10], [4]. This literature only presents control laws for redistribution of a robotic swarm across multipe sites without any optimization of any types of cost. Other literature considers the dynamic desired configuration that the robots will not stop moving [3], [8], [7], [5], [6]. The cost function in this literature is just the number of robots that are moving among the sites. They only consider the optimization problem of the cost consumed by the robotic swarm after the desired configuration is achieved. In contrast, we consider the static desired configuration and minimize the cost (e.g., energy, time) consumed by all the robots to achieve the desired configuration. Our objective function consists of the cost of each robot to move from one site to another, such as energy, distance and time. The cost considered in this paper is more realistic than that of the previous works. The problem can be posed as an optimal control problem. This optimal control problem is in general hard to solve.

Our contributions are as follows:

1) We formulate a linear programming based heuristic (one-step lookahead) feedback control law.
2) We provide a closed form feedback control law that performs comparable with the LP-based feedback control law.
3) Both the above methods provide state-of-the-art performance.
4) Our simulation results show (a) convergence time is almost independent of the number of robots. (b) the average length of distance travelled by each robot is independent of the number of robots.

This paper is organized as follows: Section II discusses the related work from previous research. Section III formulates the problem. Section IV presents the embedded algorithm for the robots and proposes three control laws for the central controller. Section V shows the simulation results of the proposed methods and baseline. Finally, we conclude in Section VI.

II. Related Work

Some related re-assignment homogeneous robotic swarm methods have been proposed in the literature. The objectives of the previous research can be categorized into three types - minimize convergence time or number of state switches [2], optimize energy [3] and only propose a general control law [4]. The methods in the previous literature can be also categorized as closed loop (feedback) control laws and open-loop control laws. Furthermore, the models considered in the previous literature can be classified as continuous or discrete in the time domain.

In [5], [6], [7], [3] and [8], the authors assume the models are continuous in the time domain. They consider the problem of real robots moving among different sites, so they model the flow of robots as a differential equation model. Because they consider the robots travelling between two sites, they both use the time-delayed differential equation model. The author in [3] tries to maximize the convergence rate subject to the constraints of number of robots that are still moving among the sites at equilibrium (after achieving the desired configuration). However, in this paper, we consider the energy cost during the re-assignment process other than the energy cost at equilibrium which means that the re-assignment process will stop and robots will stop moving when the target configuration is reached. Similar to [3], the authors of [7] and [8] also model the problem as time-delayed differential equations but with feedback. The quorum based stochastic control policies proposed in these two papers consider the heuristic of maximum allowed number of robots passing between two sites whose number of robots is much larger or smaller than the desired number. They assume that the number of moving robots represents the energy cost. However, we consider the energy cost subject to the triangular inequality such as the movement distance of robots in the Euclidean space which is more appropriate because we take the geometry of the multiple sites into account. The authors in [6] and [5] consider the Laplace Transform of the differential equations and view the problem as a filtering problem. Note that the time-delayed differential equation model considered in the above literature assumes that not only the time is continuous but also the distribution of the robotic swarm over the multiple sites is continuous in order to solve the differential equations. In fact, the distribution of the swarm over multiple sites is discrete because the number of robots is an integer.

Except for continuous time assumption, some other works assume that time is discrete. The authors study the problem of controlling cellular artificial muscles which has the similar problem formulation as the redistribution of robotic swarm across multiple sites in [2], [9] and [10]. The cellular artificial muscle consists of multiple cellular units(agents) with binary state - 'ON' and 'OFF'. The problem in these papers is to control the agents to reach the target configuration of states. These papers propose methods of feedback control policies for multiple agents over two sites. They also assume that the agents can receive control signals from the central controller like other previous literature. However, the central controller will send message to the agents all the time

according to the current state and will send a 'stop' message when it observes that the agents reach the desired state. We use the same assumption in our paper. The authors in [11] and [4] model the general re-assignment problem as Markov chain model and directly give the stochastic matrix of the Markov chain. Both of the proposed methods in these two papers are just to compute a stochastic matrix with a given steady state which means that the methods in these two papers are open-loop control laws. In [4], Metropolis-Hastings algorithm(M-H algorithm) is proposed to compute the stochastic matrix with a given steady state subject to the motion constraints between each site. The conditions of the problem in [4] are similar with our problem. Thus we compare our proposed methods with the algorithm in [4] and show that our feedback control laws outperform this algorithm.

III. Problem Statement

Suppose we have N robots and m sites. Let $n_i(t)$ be the number of robots in site i at time t. Let $\mathcal{Q} = \{\mathbf{N}_i \in \mathbb{R}^m | \mathbf{N}_i = (n_1, n_2...n_m), st. \sum_{i=1}^{m} n_i = N\}$ be the space of all possible configurations of N robots distributed over m sites. Let $\mathbf{N}_i(t)$ be the configuration at time t. The goal of redistribution process is just to control the robot swarm from the initial configuration $\mathbf{N}_l(0)$ to reach the given target configuration $\mathbf{N}_{tf}$. In our case, we require no communication among the robots and a central controller can send control signal to all the robots at each time step. We assume that all the robots are homogeneous. Each robot can know the current site it is in and can decide which site to move to or stay according to the control signal. Similar with previous literature such as [10] and [3], the central controller can know the current configuration of the robot swarm over m sites.

Let $U \in \mathbb{N}^{m \times m}$ be robot flow matrix, where each element $U_{i,j}$ represents the number of robots that move from site i to site j. If $U_{i,j} > 0$, robots are moving out of site i, and if $U_{i,j} < 0$, robots are moving into the site i. We can also denote a cost matrix among the sites $D = \{d_{i,j} | i = 1, 2...m, j = 1, 2...m\}$, where $d_{i,j}$ is the cost that one robot consumes when it moves between site i and site j. We assume the cost has the following property that the cost to go from site i to site j is less than or equal to the sum of the cost go from site i to site k and site k to site j. One simple example of the cost is the distance between two sites. The goal of control law is to minimize total movement distance C.

$$C = \frac{1}{2} \sum_{t=1}^{tf} \sum_{i=1}^{m} \sum_{j=1}^{m} |U_{i,j}(t)| d_{i,j} \qquad (1)$$

Because the robots are anonymous, the central controller can not identify individual robot. We can only control the robotic swarm together and send the same signal to all the robots. Thus we can send a control signal as transition matrix P to the robotic swarm, where $P_{i,j}$ $(i \neq j)$ is the probability of robots in site i moving to site j and $P_{i,i}$ is the probability of robots in site i staying in site i. Then we should get the control

signal P by minimizing the expected total movement distance $L_c(\mathbf{N}_l(0), \mathbf{N}_{tf})$. The definition of expected movement distance is shown in (2). So the formulation of our problem is shown in (3).

$$L_c(\mathbf{N}_l(0), \mathbf{N}_{tf}) = \sum_{t=1}^{tf} \sum_{i=1}^{m} \sum_{j=1}^{m} P_{i,j} n_i(t) d_{i,j} \qquad (2)$$

$$\min \sum_{t=1}^{tf} \sum_{i=1}^{m} \sum_{j=1}^{m} P_{i,j}(t) n_i(t) d_{i,j}$$
$$\text{s.t.} \quad \sum_{j=1}^{m} P_{i,j} = 1, i = 1, ...m \qquad (3)$$
$$0 \le P_{i,j} \le 1, i = 1, ...k, j = 1, ..., m$$
$$\mathbf{N}(tf) = \mathbf{N}_{tf}$$

IV. SOLUTION APPROACHES

According to the problem statement, all the robots in the robotic swarm are homogeneous and they will select sites for the next step independently based on the control signal sent by the central controller. The site selection algorithm is embedded in each robot. The idea for this algorithm is that each robot propagates its position as a realization of the Markov Chain independently and treats the control signal P as the transition matrix. The first step of this algorithm is to recognize the robot's current index of the site. The last two steps are just the process to generate a random number from a multinomial distribution. Note that this algorithm is a common method for swarm robots.

Site Selection Algorithm(SSA)

1) Each robot recognizes its current site i, $i \in 1, 2, 3...m$

2) Each robot generates a random number y from a uniform distribution. $y \sim U(0, 1)$

3) The next site for each robot is j, where $\sum_{l=1}^{j-1} P_{i,l} \le y \le \sum_{l=1}^{j} P_{i,l}$

Then, the main problem is how to determine the control law P which can lead the robotic swarm to the target configuration $\mathbf{N}_{tf}$. In this section, we will propose three methods to compute the control laws. First, we will classify the control laws into two types - open-loop control and feedback control. Open-loop control means that the control signal P is only determined by the target configuration and will not change in each time step. It is actually a Markov Chain. In this case, the problem is computing a transition matrix P for a Markov Chain with the target configuration $\mathbf{N}_{tf}$ as its steady state. On the other hand, feedback control means that the control signal P is computed at each time step according to the current configuration and target configuration of the robotic swarm. In the following section, we will introduce one simple open-loop control law and two feedback control laws. In the next section, we will compare these methods with an open-loop control law - Metropolis-Hastings algorithm introduced in [4].

A. Closed Form Open-Loop Control Law

In this model, we considered Markov Chain model for our problem. Let $\mathbf{x}(t) = \mathbf{N}(t)/N$ be the proportion of robots in each site at time t. Note that $\mathbf{x}(t)$ is the robot distribution over m sites and it is not the probability distribution in Markov Chain. However, we can assume $\mathbf{x} = (P(s_1), P(s_2)...P(s_m))$ to be the probability distribution over m sites. Then $\mathbf{x}_{tf}$ is the steady state of this model and the control law P is just the transition matrix also called stochastic matrix. So the problem is equal to find transition matrix P with the steady state $\mathbf{x}_{tf}$. (4) shows the formulation of our problem. (5) shows one heuristic solution for this problem.

$$\begin{cases} \mathbf{x}_{tf} = \mathbf{x}_{tf} P \\ \sum_{j=1}^{m} P_{i,j} = 1 \\ 0 \le P_{i,j} \le 1 \end{cases} \qquad (4)$$

$$P_{i,j} = \begin{cases} \dfrac{1/x_i}{(m-l-1) \sum_{i=1, x_i \neq 0}^{m} (1/x_i)} & \text{, if } x_i \neq 0, x_j \neq 0, i \neq j \\ 1 - \sum_{j=1, j \neq i}^{m} P_{i,j} & \text{, if } i = j \\ 0 & \text{, else} \end{cases} \qquad (5)$$

where l is the number of sites that $x_i = 0$. Note that this method only depends on the steady state $\mathbf{x}_{tf}$, we only need to calculate P once before the redistribution process starts.

B. Linear Programming based Feedback Control Law

In this model, we are considering feedback control law which means that the control signal P is computed at each time step. In order to minimize the expected movement distance defined in (2) and avoid the global minimum discussed in Section III, we minimize the expected movement distance for each step satisfying one step convergence constraint. One step convergence constraint, defined as $\mathbf{N}(t)P(t) = \mathbf{N}_{tf}$, means that the expected configuration of robotic swarm in the next step is the target configuration. This constraint can ensure the convergence of the redistribution process. Thus the problem can be written as (6).

$$\min \sum_{i=1}^{m} \sum_{j=1}^{m} P_{i,j}(t) n_i(t) d_{i,j}$$
$$\text{s.t.} \quad \sum_{j=1}^{m} P_{i,j} = 1, i = 1, ...m$$
$$0 \le P_{i,j} \le 1, i = 1, ...k, j = 1, ..., m$$
$$n_i(tf) - n_i(t) =$$
$$\sum_{j=1, j \neq i}^{m} (P_{j,i}(t) n_j(t) - P_{i,j}(t) n_i(t)), i = 1, ..., m \qquad (6)$$

Then above LP can be solved using standard solver. The first two constraints are the constraints to ensure matrix $P(t)$ is a right stochastic matrix. The LP can be solved in polynomial time using interior point methods [12]. However, it is more desirable to have a closed form solution for the feedback control law. In the next subsection, we will present a closed

form feedback control law for this problem which performs similar with the linear programming based feedback control law.

C. Closed Form Feedback Control Law

Similar with the linear programming based feedback control law, we also want to minimize the expected movement distance for each step satisfying both the one step convergence constraint and right stochastic matrix constraint. However, we add more movement constraints which are trying to get less movement distance. And we will show that these constraints are necessary and solving these constraints can lead to a system of linear equations. The solution in this closed form feedback control law is just a specific solution for this linear system.

The feedback control law ensures the information of current configuration of robotic swarm can be reached, so we can get additional movement constraints based on the current and target configurations. The key idea for this method is to avoid unnecessary robot movement flow. According to the site selection algorithm(SSA), robots cannot move into site i, if $P_{j,i}(t) = 0$ for all $j \neq i$. In addition, robots cannot move out of site i, if $P_{i,i}(t) = 1$ and robots cannot stay in site i, if $P_{i,i}(t) = 0$. We will discuss the movement constraints in different cases. Because we are considering 'one step convergence' assumption, we have $n_i(t+1) = n_i(tf)$. In the following discussion, we use tf instead of $t+1$.

First, we separate the sites into two sets $\mathcal{S}, \mathcal{T}$ representing sink and source respectively. $\mathcal{S}$ is the set of sites with $n_i(t) \leqslant n_i(tf)$ which means that robots should move into this site. $\mathcal{T}$ is the set of sites with $n_i(t) > n_i(tf)$ which means that robots should move out of this site. Suppose that we have l sites in $\mathcal{S}$ and k sites in $\mathcal{T}$. Because we want to avoid unnecessary movement, we should have constraints as no robots moving out of sinks and no robots moving into sources. We discuss the problem according to these two sets.

Case 1 $i \in \mathcal{S}$ $(n_i(t) \leqslant n_i(tf))$. In this case, other robots should move into the site i, so we require that no robots move out of this site. Thus we have

$$P_{i,j}(t) = \begin{cases} 1 & \text{, if } i \in \mathcal{S}, i = j \\ 0 & \text{, if } i \in \mathcal{S}, i \neq j \end{cases} \quad (7)$$

Case 2 $i \in \mathcal{T}$ $(n_i(t) > n_i(tf))$. In this case, some of robots in the site i should move out to other sites. We need to consider the situation of other sites and we will discuss this case in two different subcases.

Case 2.1 $j \in \mathcal{T}$ $(n_j(t) > n_j(tf))$. In this subcase, site j is also a source which means that some of the robots in the site j also should move out to other sites. So we should require that there is no robot moving between site i and j in order to avoid unnecessary movement. Then we have $P_{i,j}(t) = 0$, if $i \in \mathcal{T}, j \in \mathcal{T}, i \neq j$. Now we try to compute $P_{i,i}(t)$, if $i \in \mathcal{T}$ which is a special case of Case 2.1 when $j = i$. Because

$i \in \mathcal{T}$, robots should move out of this site, we can require that no robots move into this site in order to avoid unnecessary movement. Then we have $P_{j,i}(t) = 0$, if $i \in \mathcal{T}, j \neq i$. Thus we have

$$\begin{aligned} n_i(tf) &= n_i(t) - \sum_{j=1, j \neq i}^{m} n_i(t) P_{i,j}(t) \\ &= n_i(t) - n_i(t) \sum_{j=1, j \neq i}^{m} P_{i,j}(t) \\ &= n_i(t) - n_i(t)(1 - P_{i,i}(t)) \\ &= n_i(t) P_{i,i}(t) \\ \therefore P_{i,i}(t) &= \frac{n_i(tf)}{n_i(t)}, \text{if } i \in \mathcal{T} \end{aligned}$$

So we have

$$P_{i,j}(t) = \begin{cases} \frac{n_i(tf)}{n_i(t)} & \text{, if } i \in \mathcal{T}, j \in \mathcal{T}, i = j \\ 0 & \text{, if } i \in \mathcal{T}, j \in \mathcal{T}, i \neq j \end{cases} \quad (8)$$

Case 2.2 $j \in \mathcal{S}$ $(n_j(t) \leqslant n_j(tf), i \neq j)$ which means that site j is a sink. Then we have

$$\begin{aligned} n_j(tf) - n_j(t) &= \sum_{i=1, i \neq j}^{m} (P_{i,j}(t) n_i(t) - P_{j,i}(t) n_i(t)) \\ &= \sum_{i=1, i \neq j}^{m} P_{i,j}(t) n_i(t) \\ &= \sum_{i=1, i \neq j, i \in \mathcal{T}}^{m} P_{i,j}(t) n_i(t) \end{aligned} \quad (9)$$

Note that the only case we haven't solved is site i is a source and site j is a sink. So we will have l similar equations as (9) and lk unknown variables $P_{i,j}(t)$, where l is the number of sinks and k is the number of sources. Thus we get a linear system (10).

$$\begin{cases} \sum_{i=1, i \neq j, i \in \mathcal{T}}^{m} P_{i,j}(t) n_i(t) = n_j(tf) - n_j(t) & \text{, for all } j \in \mathcal{S} \\ \sum_{j=1, j \neq i, j \in \mathcal{S}}^{m} P_{i,j}(t) + P_{i,i}(t) = 1 & \text{, for all } i \in \mathcal{T} \end{cases} \quad (10)$$

Then we can have a specific solution for this linear system as

$$P_{i,j}(t) = -\frac{\delta_j \delta_i}{n_i(t) \Delta}, \text{if } i \in \mathcal{T}, j \in \mathcal{S}, i \neq j \quad (11)$$

where $\delta_j = n_j(tf) - n_j(t)$, $\delta_i = n_i(tf) - n_i(t)$ and $\Delta = \sum_{j=1, j \in \mathcal{S}} \delta_j = -\sum_{i=1, i \in \mathcal{T}} \delta_i$ which is the total number of moving robots. In summary, we have

$$P_{i,j}(t) = \begin{cases} 1 & \text{, if } i \in \mathcal{S}, i = j \\ 0 & \text{, if } i \in \mathcal{S}, i \neq j \\ \frac{n_i(tf)}{n_i(t)} & \text{, if } i \in \mathcal{T}, j \in \mathcal{T}, i = j \\ 0 & \text{, if } i \in \mathcal{T}, j \in \mathcal{T}, i \neq j \\ -\frac{\delta_j \delta_i}{n_i(t) \Delta} & \text{, if } i \in \mathcal{T}, j \in \mathcal{S}, i \neq j \end{cases} \quad (12)$$

V. SIMULATION RESULTS

In this section, we will introduce the simulation results and compare our proposed methods with the M-H algorithm proposed in [4]. The M-H algorithm is an open-loop control law. First, we will test the four methods on both the two cases - different robotic swarm size and different number of sites. Both the average convergence time steps and the average movement to converge for each robot ($\bar{L} = L/N$) will be compared in each case. The average convergence time steps is defined as $\sum_{i=1}^{T} t_i/T$ where t_i represents the number of iterates to converge in the i th simulation and T represents the number of simulations. And the average movement to converge for each robot is defined as $\sum_{i=1}^{T} \sum_{t=1}^{t_i} \sum_{j=1}^{N} m_{j,t}/(NT)$ where $m_{j,t}$ represents the movement distance of robot j at the t th iterate in the i th simulation. Then detailed performance of the linear programming based feedback control law and closed form feedback control law will be shown. We ran 500 simulations for each case and every parameter configuration. We randomly generated a cost matrix D which satisfied Euclidean geometry for all the simulations which means that D did not change during all simulations. The initial configuration for all simulations is that all robots are in the first site and the target configuration is that the robots are equally distributed to all sites. Note that the simulation stopped if abs$(n_i(t) - n_i(tf)) \leq 1,$ for $i = 1, ...m$.

A. Different Robot Swarm Size

In this case, we used 6 sites and varying number of robots (100 to 500). Fig. 1 shows the relationship between average convergence time steps and number of robots. Fig. 2 shows the relationship between average movement to converge for each robot and number of robots. Fig. 1 and Fig. 2 show that the average convergence time steps and average movement for the closed form open-loop control law and M-H law increase as the number of robots increases. Because the scale of y - axis is too large, the curve of closed form feedback control law and the curve of linear programming based feedback control law overlap with each other. The figures shows that the feedback control laws perform much better than the open-loop control laws. Fig. 3 and Fig. 4 only show the performance of these two type of feedback control laws. **There is no significant difference between these two feedback control laws and the performance does not change much when the number of robots increases.**

B. Different Number of Sites

In this case, we used 100 robots and varying number of sites (3 to 5). Fig. 5 shows the relationship between average convergence time steps and number of sites. Fig. 6 shows the relationship between average movement to converge for each robot and number of sites. Fig. 5 and Fig. 6 shows that the average convergence time steps and average movement of the closed form open-loop control law and M-H law increase as the number of sites increases. Because the scale of y - axis is too large, the curve of closed form feedback control law and the curve of linear programming based feedback control law overlap with each other. The figures show that the feedback control laws perform much better than the open-loop control laws. Fig. 7 and Fig. 8 only show the performance of these two type of feedback control laws. **There is no significant**
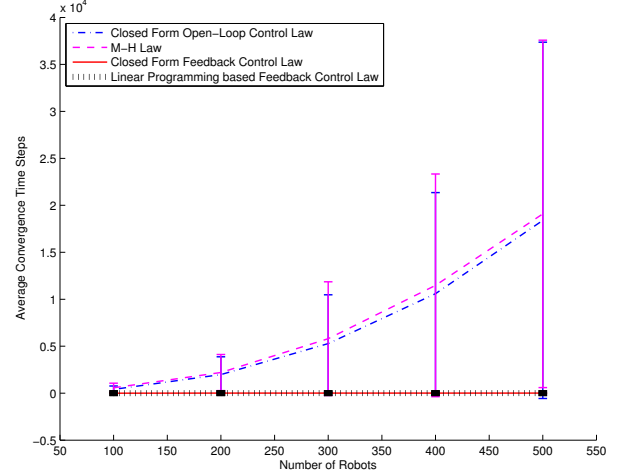

Fig. 1. Relationship between average convergence time steps and number of robots. The feedback control laws outperform open-loop control laws. Note that the closed form feedback control law and linear programming based feedback control law overlap with each other because of the scale of y - axis. Fig. 3 shows the difference of these two methods using appropriate scale in the same situation as this figure.
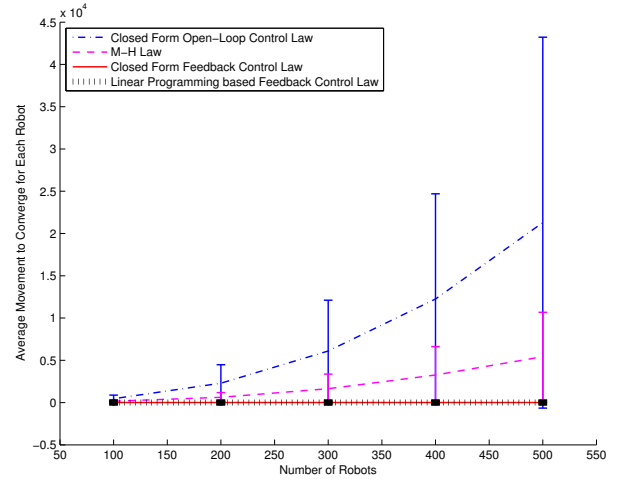

Fig. 2. Relationship between average movement to converge for each robot and number of robots. The feedback control laws outperform open-loop control laws. Note that the closed form feedback control law and linear programming based feedback control law overlap with each other because of the scale of y - axis. Fig. 4 shows the difference of these two methods using appropriate scale in the same situation as this figure.

difference between these two feedback control laws and the performance does not change much for different number of sites.

C. Performance of Linear Programming based Feedback Control Law

Fig. 1, 2, 5 and 6 show that linear programming based feedback control law outperforms the two open-loop control laws. Fig. 9 and Fig. 10 show the detailed performance of linear programming based feedback control law. There is no significant difference between the average convergence time steps for all cases.
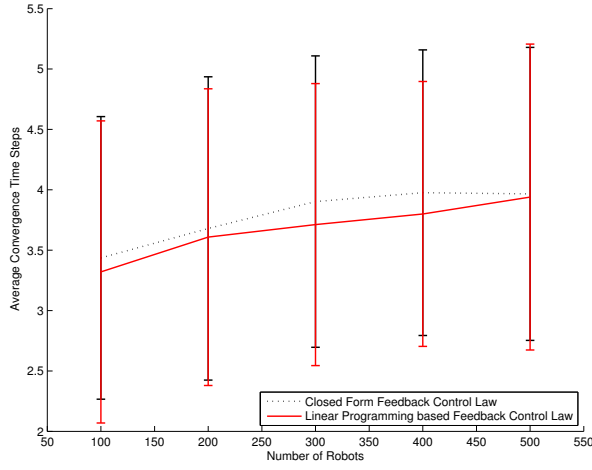
Fig. 3. Relationship between average convergence time steps and number of robots. The average convergence time does not change much for different number of robots.
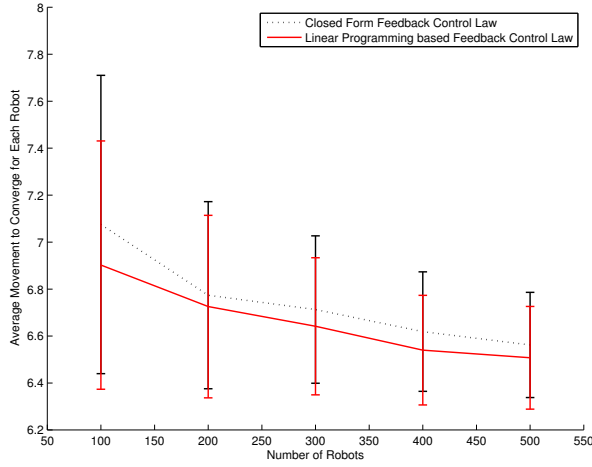

Fig. 4. Relationship between average movement to converge for each robot and number of robots. The average movement does not change much for different number of robots.

D. Performance of Closed Form Feedback Control Law

Fig. 1, 2, 5 and 6 show that closed form feedback control law outperforms the two open-loop control laws. Fig. 11 and Fig. 12 show the detailed performance of closed form feedback control law. There is no significant difference between the average convergence time steps for all cases.

VI. CONCLUSION

This paper categorizes the control policies of redistribution of robotic swarm into two types, open-loop and feedback. We propose one closed form open-loop control law and two feedback control laws. We solve the energy cost optimization problem by a heuristic. The simulation results show that our proposed linear programming based feedback control law and closed form feedback control law outperform the baseline. Note that there is no significant difference between the performance of the closed form feedback control law and linear programming based feedback control law. And there is also no
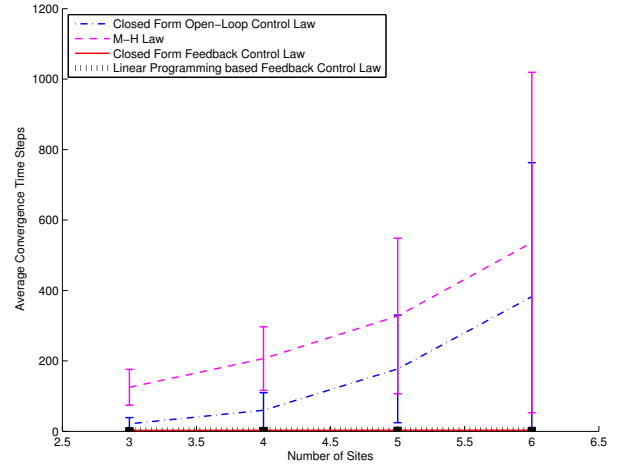

Fig. 5. Relationship between average convergence time steps and number of sites. The feedback control laws outperform open-loop control laws. Note that the closed form feedback control law and linear programming based feedback control law overlap with each other because of the scale of y - axis. Fig. 7 shows the difference of these two methods using appropriate scale in the same situation as this figure.
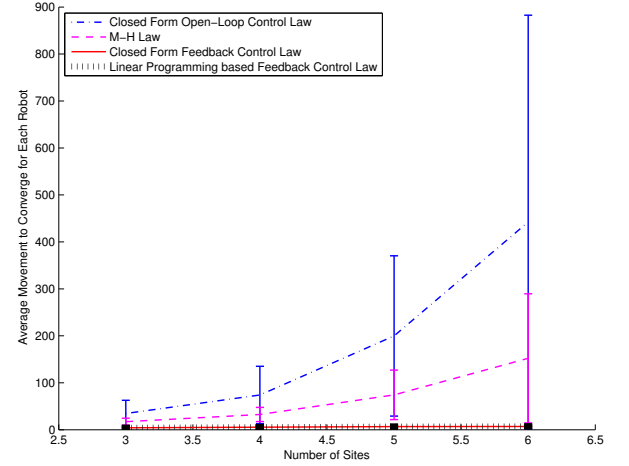

Fig. 6. Relationship between average movement to converge for each robot and number of sites. The feedback control laws outperform open-loop control laws. Note that the closed form feedback control law and linear programming based feedback control law overlap with each other because of the scale of y - axis. Fig. 8 shows the difference of these two methods using appropriate scale in the same situation as this figure.

significant change of the performance for both control laws as the number of robots increases.

REFERENCES

[1] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[2] L. Odhner and H. Asada, "Stochastic recruitment: Controlling state distribution among swarms of hybrid agents," in *American Control Conference, 2008.* IEEE, 2008, pp. 4226–4231.

[3] S. Berman, Á. Halász, M. A. Hsieh, and V. Kumar, "Optimized stochastic policies for task allocation in swarms of robots," *Robotics, IEEE Transactions on*, vol. 25, no. 4, pp. 927–937, 2009.

[4] B. Acikmese and D. S. Bayard, "A markov chain approach to probabilistic swarm guidance," in *American Control Conference (ACC), 2012.* IEEE, 2012, pp. 6300–6307.
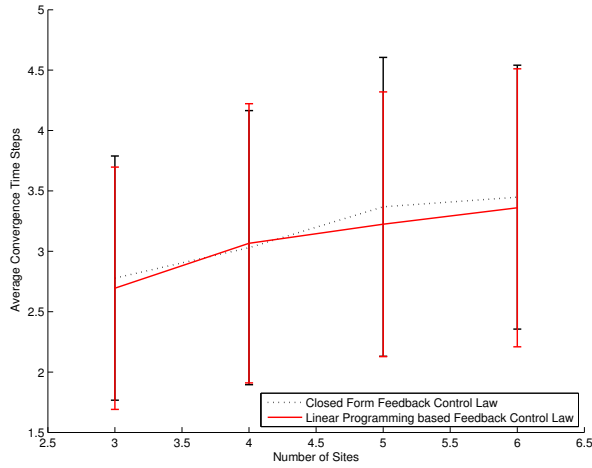
Fig. 7. Relationship between average convergence time steps and number of sites. The average convergence time does not change much for different number of sites.
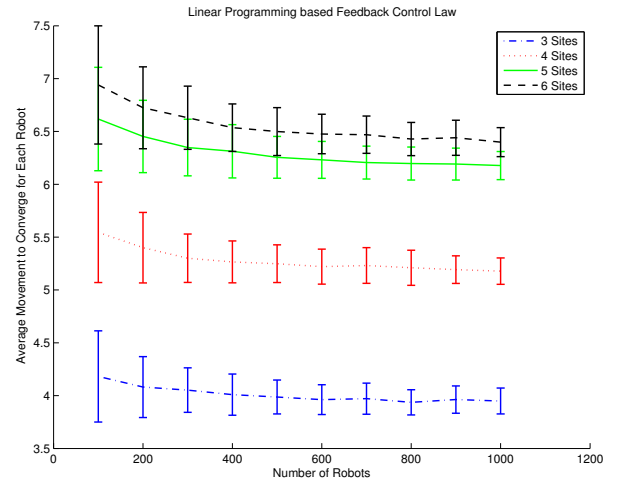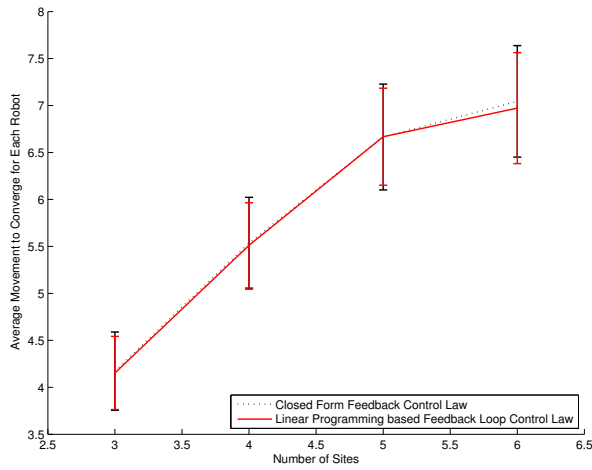

Fig. 8. Relationship between average movement to converge for each robot and number of sites. The average movement does not change much for different number of sites.
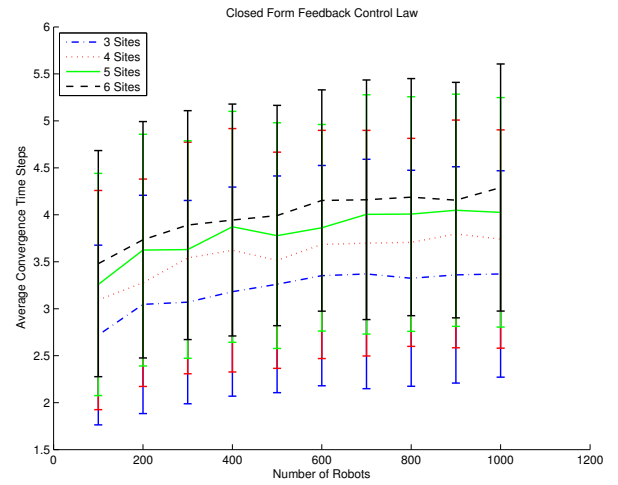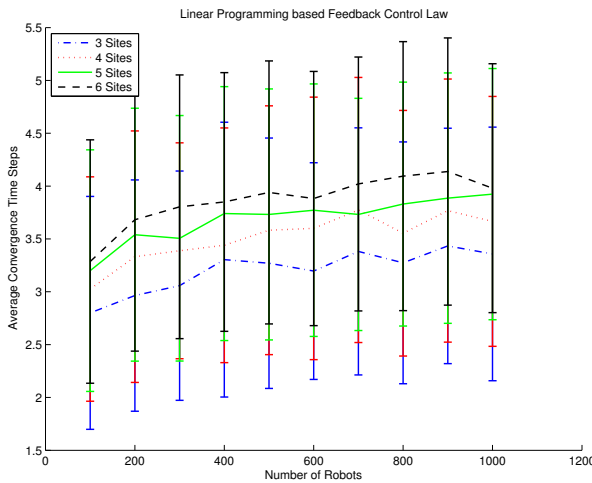

Fig. 9. Average convergence time steps using linear programming based feedback control law.

Fig. 10. Average movement to converge for each robot using linear programming based feedback control law.

Fig. 11. Average convergence time steps using closed form feedback control law.

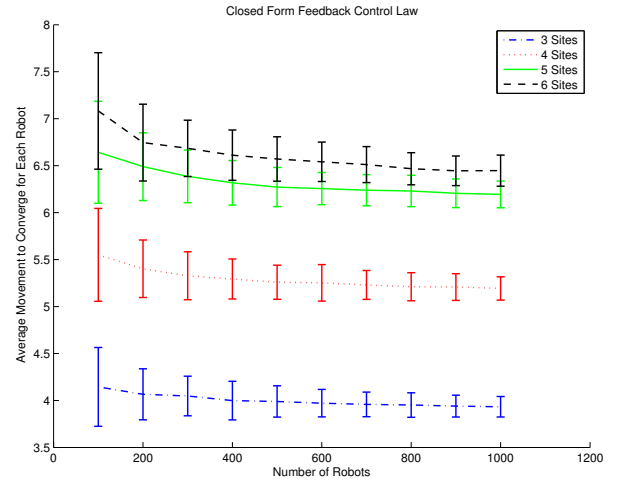
Fig. 12. Average movement to converge for each robot using closed form feedback control law.

[5] T. W. Mather, C. Braun, and M. A. Hsieh, "Distributed filtering for time-delayed deployment to multiple sites," in *Distributed Autonomous Robotic Systems*. Springer, 2013, pp. 299–312.

[6] T. W. Mather and M. A. Hsieh, "Macroscopic modeling of stochastic deployment policies with time delays for robot ensembles," *The International Journal of Robotics Research*, vol. 30, no. 5, pp. 590–600, 2011.

[7] M. A. Hsieh, Á. Halász, S. Berman, and V. Kumar, "Biologically inspired redistribution of a swarm of robots among multiple sites," *Swarm Intelligence*, vol. 2, no. 2-4, pp. 121–141, 2008.

[8] Á. M. Halász, M. A. Hsieh, S. Berman, and V. Kumar, "Dynamic redistribution of a swarm of robots among multiple sites." in *IROS*, 2007, pp. 2320–2325.

[9] L. Odhner and H. Asada, "Stochastic recruitment: A limited-feedback control policy for large ensemble systems," *Robotics: Science and Systems IV*, 2008.

[10] L. Odhner, J. Ueda, and H. H. Asada, "Stochastic optimal control laws for cellular artificial muscles," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 1554–1559.

[11] I. Chattopadhyay and A. Ray, "Supervised self-organization of homogeneous swarms using ergodic projections of markov chains," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, no. 6, pp. 1505–1515, 2009.

[12] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.