LA-UR-12-25669

Approved for public release; distribution is unlimited.

Title: A Simulation Based Analysis of Distributed Coverage and Dynamic Relocation for Mobile Sensor Networks Using Potential Fields

Author(s): Thulasidasan, Sunil

Intended for: Report

A Simulation Based Analysis of Distributed Coverage and Dynamic Relocation for Mobile Sensor Networks Using Potential Fields

Sunil Thulasidasan
Computational Sciences Division
Los Alamos National Laboratory
Los Alamos, NM 87545
sunil@lanl.gov

***Abstract*— We simulate and study a distributed algorithm for the coverage problem in mobile sensor networks, in particular, one that attempts to solve the blanket coverage problem when very little information is available regarding terrain topology. We use virtual forces based on potential fields to achieve coverage and obstacle avoidance. Additionally, a distributed dynamic relocation algorithm – triggered by event occurrences – is also simulated which achieves convergence to a formation that surrounds the event of interest, in the absence of prior information regarding formation geometry. The entire scheme is purely distributed and completely based on the concept of potential fields [5], [4]. Control laws are formulated based on potential fields; convergence to equilibrium, graph connectivity properties and sensitivity to parameters are studied. Discussion of results and directions for future work are provided at the end.**

I. INTRODUCTION

This paper examines the problem of coverage in sensor networks via distributed algorithms. In particular, we consider algorithms that attempt to solve the coverage problem when very little information is available regarding terrain topology. In such cases, nodes start out from some initial configuration – possibly close together – and then spread out or *diffuse* through the network according to some control law; the desired outcome is maximal coverage area while maintaining network connectivity. A well-known technique in such cases is one that uses a potential-field approach to sensor node deployment [5]. Here, virtual fields are constructed such that nodes are repelled by other nodes and obstacles in close proximity. This forces the nodes to spread out through the network, and by applying an additional viscous damping force, convergence to some static equilibrium configuration is achieved. The algorithm does not require models of the environment, localization, and in the basic implementation, even communication between the nodes. All that is assumed is that nodes are equipped with sensors that allow measurement of range and bearing of nearby nodes and obstacles. In addition, coverage is not explicitly engineered into the control law; rather, coverage is an emergent property of the network.

II. RELATED WORK

Extensions of this idea are described and explored in [2], [1], [4], [7]. [4] considers the case when nodes, initially placed in some random configuration, react to an event – say an environmental event like a fire – through an attractive potential, while coverage and equilibrium are achieved through repulsive and damping forces respectively. Further, the nodes might also be required to surround a region of interest, and thus the control laws should be able to achieve a formation whose geometry is not known before-hand. In [7]and [1], strategies for node diffusion from regions of higher density to those of lower density are described, and simulation results are presented.

Note that the main problem being considered here is sensor self deployment in a *purely distributed* fashion with the aim of maximizing coverage while preserving network connectivity. Also of importance is the ability of sensor nodes to dynamically relocate during the occurrence of an event, and surround the event without having any prior information regarding formation geometry. In what follows we present some of the assumptions we make in Section III; the basic algorithm and control laws are described in Section IV and simulation results and discussion are presented in Section V. We conclude with directions for future investigation in Section VI.

III. ASSUMPTIONS

In this section we describe the assumptions (some of which are simplifying for feasibility sake) we make. Our goal is to achieve a placement of nodes that maximizes the total detection area without paying attention to holes in the coverage area. Thus we are dealing with the *blanket coverage* issue [3]. Further, we are implementing a purely distributed algorithm, since we assume that nodes do not have any information regarding their absolute positions; all they are capable of determining is their relative range and bearing, as well as communicating with other nodes within their communication radius. The communication radius itself is assumed to be *twice* the sensing radius. All nodes are homogenous in this problem, and have the same sensing and communication radii and other physical constraints. In the deployment phase we do not address the problem of maintaining a certain node density. During relocation, in our simulations, we relocate *all* nodes to the region of interest; presumably this is wasteful in real life, but we make that simplifying assumption here. All nodes are subject to maximum velocity and acceleration constraints implemented via clipping. Any event within the sensing range of a node is assumed to be detected by that node with probability 1. We also do not consider the (very likely) problem of sensor nodes being damaged by the events they are monitoring, other environmental factors or random failure; indeed in our simulation, the only probabilistic elements are the initial placement of nodes, obstacles and events.

IV. The Virtual Force Algorithm

The basic goal of our algorithms is for the nodes to initially deploy to a static monitoring configuration, starting from a dense initial formation, such that coverage is maximized, while preserving network connectivity. Upon detecting an event, nodes within sensing range are attracted to the event, while other nodes follow their neighbors; our goal in this case is to surround the event of interest (we assume that the sensing nodes can localize the event location through triangulation or some other approach). To achieve these goals we use virtual force algorithms based on potential fields. We briefly describe these algorithms and the basic equations of motion here. These largely follow the algorithms described in [5] and [4]. In the basic potential field method, each node is subject to a force $\mathbf{F}$ which is the gradient of some scalar potential field U. Thus

$$F = -\nabla U$$

Following [4], we divide the potential field into three components, resulting in the following three forces: the repulsive force between nodes F^n, the repulsive force from obstacles F^o and the attractive force from events F^e. Thus the total virtual force F_i from the potential fields acting on node i at time t is given by

$$F_i(t) = F_i^e(t) + F_i^o(t) + F_i^e(t)$$

To ensure that nodes reach a static equilibrium during the deployment phase, we also add a viscous damping term. The equation of motion is then given by Newton's second law:

$$\ddot{\mathbf{x}} = (\mathbf{F} - \nu\dot{\mathbf{x}})/m$$

where $\ddot{\mathbf{x}}$ is the acceleration and m is the mass of each node. Since each node is subject to a maximum velocity and acceleration, these are clipped such that $||v|| \leq v_{max}, ||\dot{v}|| \leq \dot{v}_{max}$. In the absence of events, $F^e = 0$, and nodes maintain their equilibrium state.

A. Control Law

In our simulation, we use double integrator dynamics based on the following equations: The forces on node i during the deployment phase can be written as

$$F_i^o = -\mu_o \sum_k \frac{1}{r_{ki}^2}\frac{\mathbf{r_{ki}}}{r_{ki}}, \quad F_i^n = -\mu_n \sum_{j\in N(i)} \frac{1}{r_{ji}^2}\frac{\mathbf{r_{ji}}}{r_{ji}} \qquad (1)$$

where in the first force term $\mathbf{r_{ki}}$ denotes the displacement vector from node i to obstacle k; $\mathbf{r_{ji}}$ denotes the displacement vector from node i to neighbor j defined by sensing radius- During the relocation phase, we have an additional attractive force $F_i^e = \mathbf{r_j}$ where $\mathbf{r_j}$ denotes the displacement vector from i to the event. . Our double integrator control law during node deployment then becomes

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{F^n} + \mathbf{F^o} - \nu\mathbf{v} \end{bmatrix}$$

where $\mathbf{F^n}$ can be expressed in compact notation as $\mathbf{F^n} = \mu_n I_2 \otimes L_w$, where L_w is the weighted Laplacian, the weights depending inversely on the inter-node distance as given in (1).

During event relocation, edge tension dynamics as described in [6] are implemented to preserve network connectivity during node movement.

V. Experimental Results and Discussion

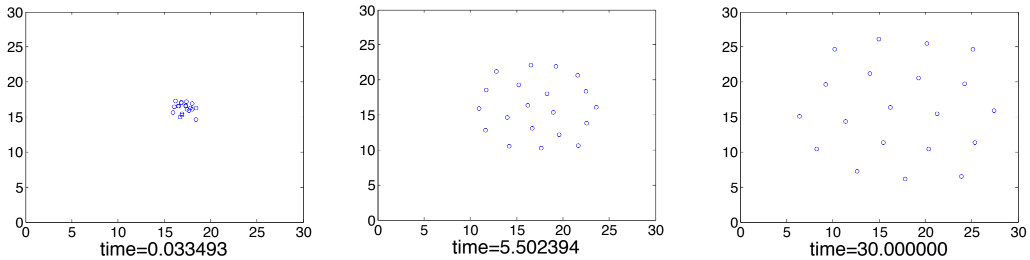
In this section we describe the simulation experiments and results of an implementation of the virtual-force/potential based coverage as well as the dynamic event-based relocation algorithms. All simulations were performed in MATLAB. We simulate networks of up to 50 nodes, covering a region of approximately 1000 square meters. The following simulation parameters were varied in our experiments:

- n: Network size $(10, 20, 30, 50)$
- n_o: Number of obstacles $(5, 10, 15)$
- μ_n: Node-repulsion coefficent $(4, 8)$
- μ_o: Obstacle repulsion coefficient $(0.01, 0.05, 0.1)$.
- ν: Viscous damping coefficient $(1, 2, 4)$.
- Δ : Sensing radius $(5m, 10m)$

The parameters were chosen based on simple initial experiments to narrow down the parameter space. While there might have been better parameter choices the simulation results indicated that these choices were reasonable. The communication radius R_c is assumed to be 2Δ based on literature survey. Further, as mentioned in section IV, we clip the maximum velocity and acceleration to $1m/s$ and $1.5m/s^2$ respectively, and each node is assumed to have unit mass. Due to space constraints we only present a small subset of the results that are representative of the general performance of the algorithm. The outcome of a very basic set up is shown in Figure 1, where the sensor nodes are deployed in a region free of obstacles, without event relocation. The only forces in this case are node mutual repulsion and the viscous damping force. As expected, the sensors deploy in a radially outward manner and reach their static deployment configuration fairly quickly. The inter-node spacing at equilibrium is roughly the value of the sensing radius.

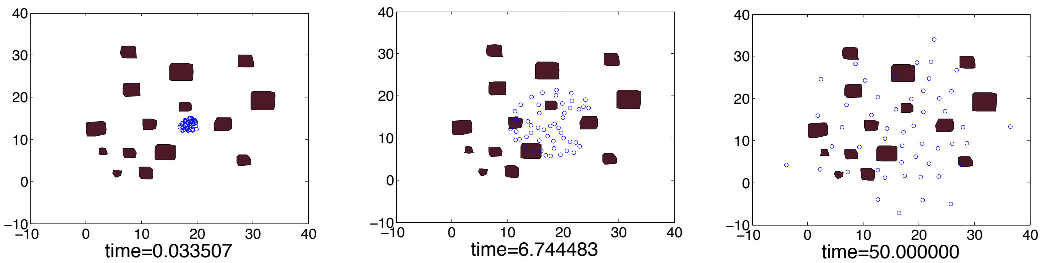
A. Obstacle Avoidance

A more realistic and interesting case is presented in Figure 2. In this scenario, 50 sensor nodes are initially deployed in an environment consisting of 15 obstacles of various sizes. The nodes initially start out in a densely packed configuration and rapidly deploy into the neighboring regions (see animation attached to report). Obstacle avoidance is not perfect – initially in a densely packed configuration, the combined repulsive forces of neighboring nodes can frequently end up pushing a node inside the object boundary; this can



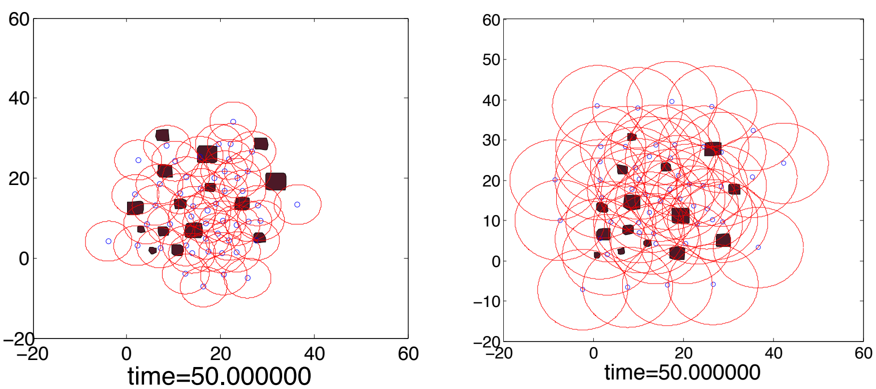
(a) (b) (c)

Fig. 1. A basic potential field set up of 20 nodes diffusing in time. Here $\Delta = 5, \mu_o = 0.1, \mu_n = 0.8, \nu = 1$, where μ_o, μ_n and ν are the obstacle repulsion, node repulsion and viscous damping coefficients respectively. Δ is the sensing radius



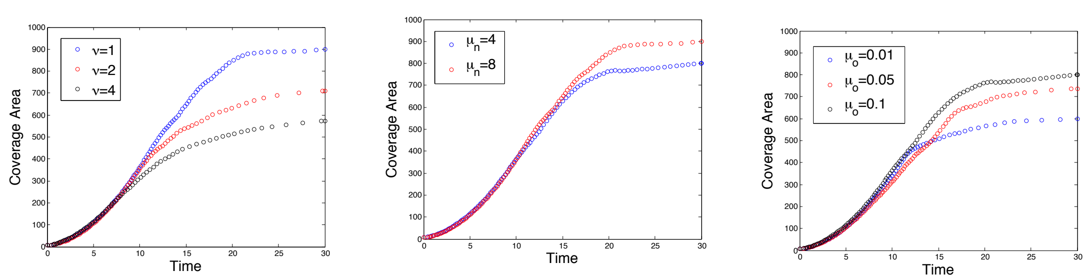
(a) (b) (c)

Fig. 2. Sensor deployment with obstacles in a 50-node setup, diffusing in time with $\Delta = 5, \mu_o = 0.1, \mu_n = 0.8, \nu = 1$.



(a) (b)

Fig. 3. Coverage area at the end of the deployment phase for two different sensing radii, $\Delta = 5$ and $\Delta = 10$. Other parameters are $\mu_o = 0.1, \mu_n = 0.8, \nu = 1$.



(a) (b) (c)

Fig. 4. Change in coverage area in a 50-node setup for different damping coefficients (a), node-repulsion coefficient (b) and obstacle repulsion coefficient (c)

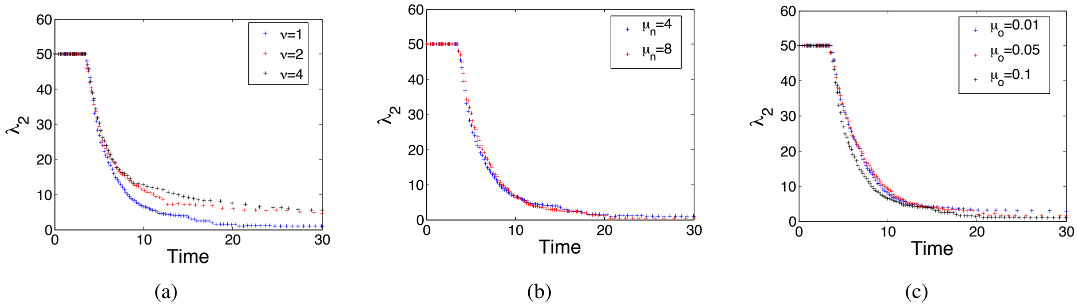


Fig. 5. Evolution of second eigenvalue (λ_2) of the communication graph Laplacian in a 50-node setup for different damping coefficients (a), node-repulsion coefficients (b) and obstacle repulsion coefficients (c). In all cases, the minimum $\lambda_2 > 0$, indicating the graph was connected.

be avoided by tweaking the obstacle repulsion coefficient. Nevertheless, obstacle avoidance worked fairly well, and the final deployment configuration in this case is shown in part (c) of Figure 2. The region covered by the 50 sensors is shown in Figure 3, one for sensing $\Delta = 5m$ and the other for $\Delta = 10m$. The red circles indicate the coverage area of each sensor. Since node repulsive forces are only active when nodes are within Δ of each other, a larger sensing radius results in a larger dispersal as shown. Acceleration and velocity clipping prevents nodes from gaining very high speeds when the internode separation is close to Δ. Note that the resulting dynamics then will not be described by the virtual force equation, but we ignore that aspect here (the authors of [5] and [4] also proceeded in a similar manner).

The coverage area at the equilibrium configuration is also sensitive to other experimental parameters, particularly the viscous damping coefficient ν and the node and obstacle repulsion coefficients, μ_n and μ_o respectively. A few of these results which indicate rate of change in coverage area as well as the final coverage area with respect to the aforementioned parameters are shown in Figure 4. For higher ν, one expects a greater retardation force, resulting in a more tightly packed configuration, and vice versa for the repulsion coefficients; this intuition agrees with results shown in Figure 4. Note that the coverage area is still growing, ever so slightly, at the end of the simulation time, because of the asymptotic approach to stabilization. In real life, to conserve sensor-node energy, the nodes would be engineered to halt once velocity falls below a certain critical point. Also of interest is that the discrete time stepping nature of the simulations often result in oscillatory behavior of the nodes near the equilibrium point. Again, these can again be solved by introducing velocity *dead-bands*.

B. Connectivity

As mentioned before, while solving the blanket coverage problem, it is also critical that the resulting formation be a connected network, especially if nodes are to be able to redeploy to areas that are not directly in their sensing range. Thus we would also like to study the graph connectivity of the equilibrium configuration and its sensitivity to various parameters. These results are shown in Figure 5, where we plot the second smallest eigenvalue of the Laplacian of the communication graph (recall that the communication radius is set to twice the sensing radius) against the force parameters. From spectral graph theory, we know that a zero second eigenvalue (λ_2) indicates a disconnected component [6]; the results in Figure 5 indicate that the graph always remains connected, although λ_2 dips to values less than 1 in the equilibrium configuration for some of the experiments. For a low retarding coefficient (ν) as well as for high repulsive coefficients (μ_n, μ_o) it is conceivable that many nodes will be pushed to the outer peripheries of the coverage region, often ending up with only one neighboring node. Recalling that λ_2 bounds the edge-cut set of a graph from below, it is not surprising that in these cases $\lambda_2 < 1$.

C. Event-based Relocation

Finally, we present results of the dynamic event-based redeployment algorithm. A redeployment sequence for two different scenarios is shown in Figure 6. Nodes remain at rest in their deployment configuration till an event occurs, denoted by the red polygon. At this point, nodes within sensing range (shown as red circles) assume the role of leaders; nodes outside the sensing range become followers. The connected nature of the deployment network assures that all nodes are able to receive information through the appropriate routing mechanisms (routing was not addressed in our experiments). In the simulation, the event itself is modeled as a stubborn node that does not update its position. This results in a "goal/leader/follower" interaction dynamics which is a directed interaction graph, where (in the classic rendezvous case) all agents converge to the position of the stubborn node (the event). In our case, the repulsive forces from the event and other nodes are also present, which results in the convergence to a formation surrounding the event of interest, as shown in figure. The actual shape of the formation is difficult to predict, depending as it does on the starting configuration and obstacle placement. Nevertheless, an event-surrounding formation is achieved in most cases. During the redeployment phase, network connectivity is maintained through edge tension dynamics. Note that, we do not consider the case of sensor nodes themselves being damaged by the event. Such situations can be overcome by achieving a desired node density in the coverage region so that the network is still resilient to the failure of a few nodes.

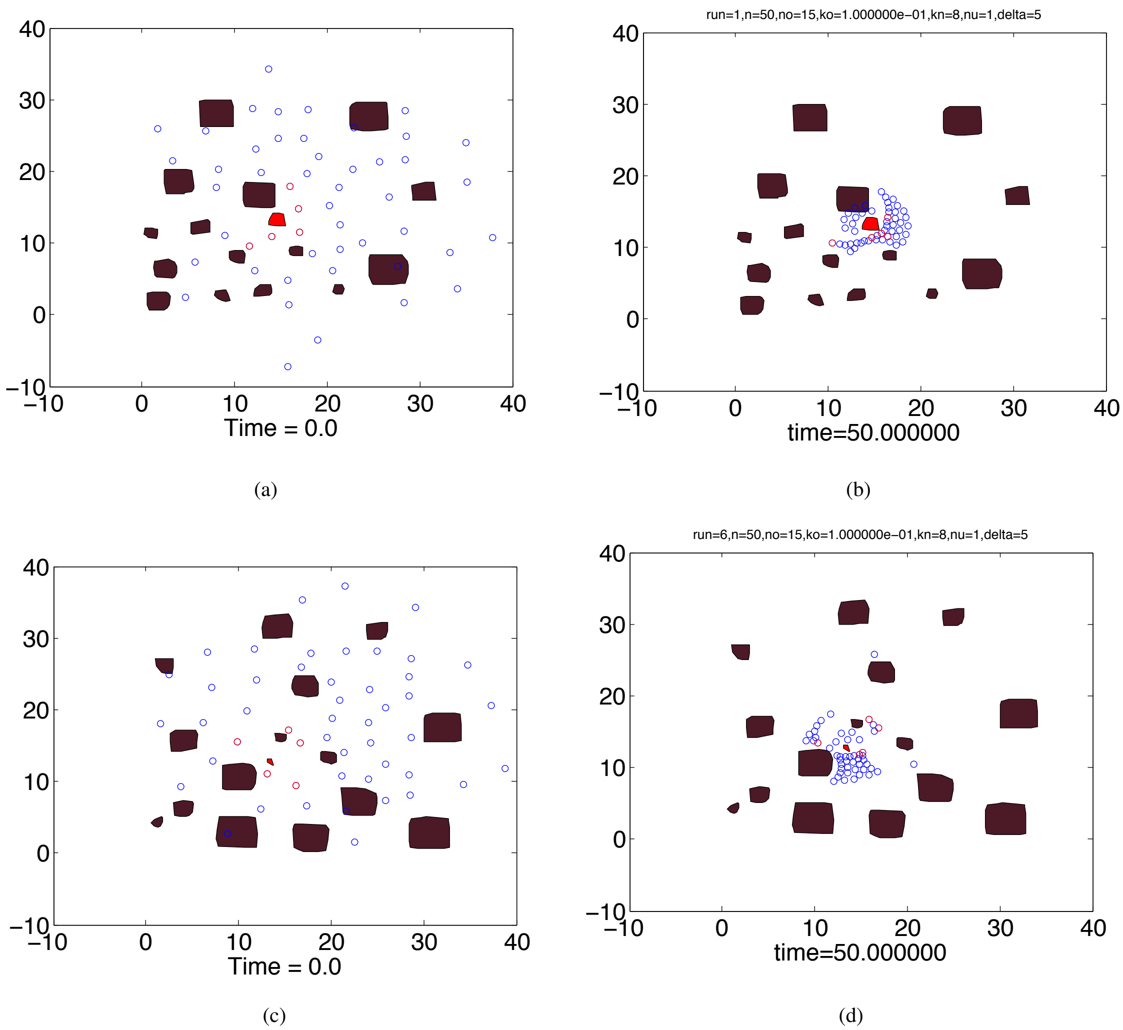


Fig. 6. Figures illustrating dynamic relocation based on event occurences for two different deployment scenarios (one per row). The first figure is the starting deployment. When an event (red polygon) occurs, the nodes closest to the event sense it (red circles). These are the leader nodes, and all other nodes converge to surround the event.

In our experiments, events happening at the periphery of the coverage region might potentially go un-noticed due to relatively sparse coverage on the peripheries.

VI. Conclusions and Future Work

Dynamic sensor node redeployment is an active area of research, and our experiments provide a number of interesting directions for further investigation. One of the interesting issues that was not considered was the formulation of control laws for maintaining network connectivity, whereby nodes are allowed to disconnect from neighbors, provided that the network as a whole remains connected. [8] proposes such control laws for maintaining connectivity, where formally, the set of desired states $\mathcal{X}_{c_n}$ corresponding to connected graphs $\mathcal{C}_n$ is such that

$$\mathcal{X}_{\mathcal{C}_n} = \{x(t) \in \mathbb{R}^{nm} \mid \lambda_2(x(t)) > 0\}$$

If $x(0) \in \mathcal{X}_{\mathcal{C}_n}$, then the control law assures that $x(t) \in \mathcal{X}_{\mathcal{C}_n} \forall\, t \geq 0$. Another major issue that was not considered here was the energy constraint on sensor nodes. Energy depleting behavior such as oscillations around equilibrium could be easily clipped and jostling for position during event relocation could also be mitigated. Further, not all nodes need to redeploy during an event; the number of redeploying nodes could be made sensitive to the intensity of the event, node energy levels and other such considerations. We intend to explore some of these issues in future work.

References

[1] Ruay-Shiung Chang and Shuo-Hung Wang. Self-deployment by density control in sensor networks. *Vehicular Technology, IEEE Transactions on*, 57(3):1745 –1755, may 2008.

[2] Jiming Chen, Shijian Li, and Youxian Sun. Novel deployment schemes for mobile sensor networks. *Sensors*, 7(11):2907–2919, 2007.

[3] D W Gage. Command control for many-robot systems. *Control*, 10(June):28–34, 1992.

[4] M. Garetto, M. Gribaudo, C.-F. Chiasserini, and E. Leonardi. A distributed sensor relocatlon scheme for environmental control. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, pages 1 –10, oct. 2007.

[5] Andrew Howard, Maja J Mataric, and Gaurav S Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. pages 299–308, 2002.
[6] Mehran Mesbahi and Magnus Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.
[7] M. Tariq, Zhenyu Zhou, Yong-Jin Park, and T. Sato. Diffusion based self-deployment algorithm for mobile sensor networks. In *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, pages 1 –5, sept. 2010.
[8] M. M. Zavlanos and G. J. Pappas. Potential fields for maintaining connectivity of mobile networks. *Trans. Rob.*, 23(4):812–816, August 2007.