Deploying Long-Lived and Cost-effective hybrid sensor networks

Wen Hu, Chun-Tung Chou, Sanjay Jha
School of Computer Science and Engineering
The University of New South Wales, Australia
{wenh, ctchou, sjha}@cse.unsw.edu.au

Nirupama Bulusu
OGI School of Science and Engineering
Oregon Health and Sciences University
nbulusu@cse.ogi.edu

July 2, 2004

Abstract

In this paper, we study the problem of network deployment in hybrid sensor networks, consisting of both resource-rich and resource-impoverished sensor devices. The resource-rich devices, called microservers, are more expensive but have significantly greater bandwidth and energy capabilities compared to the low-cost, low-powered sensors. Such hybrid sensor networks have the potential to support the higher bandwidth communications of broadband sensor networking applications, as well as the fine-grained sensing that is made possible by smaller sensor devices. We propose several techniques to investigate some fundamental questions on hybrid sensor network deployment — *for a given number of microservers, what is the maximum lifetime of a sensor network and the optimal microserver placement? What benefit can additional microservers add to the network, and how financially cost-effective is it to introduce these microservers?* For our investigation, we propose a cost model for energy usage in hybrid sensor networks, which is then formulated into an integer linear optimization problem and solved optimally. The integer linear problem solution does not scale with network size thus we introduce an approximation algorithm using tabu-search technique. Our studies show that network life time can be extended by more than 60% by adding an extra microserver to the network; the network life time of optimized microservers' placement can be more than 500% longer than the worst case life time. We

also propose a normalized cost model that balances the benefits with deployment costs, and show how to achieve an optimal deployment.

1 Introduction

This paper investigates the problem of *network deployment* in hybrid sensor/actuator networks. By hybrid sensor networks, we mean those networks consisting of both resource-rich and resource-impoverished sensor devices. The resource-rich devices, called microservers, are more expensive but have significantly greater bandwidth and energy capabilities compared to the low-cost, low-powered sensors. Such hybrid sensor networks have the potential to support the higher bandwidth communications of broadband sensor networking applications, as well as the fine-grained sensing possible by smaller sensor devices.

In the past couple of years, sensor networks research has addressed the development of sensor platforms[1], application domains, and communication paradigms[2][3][4][5]. Although previous work has considered optimal sensor network deployment [6][7][8], network deployment has not been previously considered in the context of hybrid sensor networks.

1.1 Motivation: Hybrid Sensor Networks

Historically, large scale networks have evolved to encompass myriad types of network devices. The Internet today combines different devices such as routers, servers and hosts. Even the routers can be classified into different categories (e.g., into core routers and edge routers). For large scale sensor networks that may have thousands of nodes in the future, it is more realistic to have hierarchical models of network devices rather than flat ones. Such a sensor network involves a hybrid of resource-rich specialized nodes in conjunction with small sensor devices [9]. The resource-rich nodes provide service such as (i) long-range data communications, (ii) persistent data storage, or (iii) actuation. Examples of actuation would be re-charging or replacing small nodes whose energy has been depleted, imagers which can take photos or video when activated by sensors, sprinklers used for precision agriculture which can sprinkle water in badly parched areas etc. The resource-rich node can act as a data sink, and we call it a *micro-server*. [1]

[1] The term micro-server was suggested by Deborah Estrin.

1.2 Data anycast: A communication paradigm for hybrid sensor networks

The key challenge in building Ad-Hoc multi-hop sensor networks from small, low-powered sensor nodes are scalability and energy-efficient mechanisms for data dissemination. Previously proposed data routing protocols[2][3][4][5] for sensor networks have not been designed to leverage the capabilities of hybrid devices. By exploiting resource-rich devices, the communication burden on smaller, energy, bandwidth, memory and computation-constrained sensor devices can be reduced. Consequently, these protocols may not be best suited for several applications of such hybrid sensor networks, which involve a multitude of mutually cooperative microservers.

Our thesis is that an *anycast* service, which routes sensor data to the nearest available microserver, rather than to a single designated server, can provide significant improvements to the aforementioned data dissemination protocols for such applications and networks. The intuition is that you only care for the service, not which server provides it. The anycast service should be useful for several hybrid sensor applications.

Consider the case of mobile soldiers operating in a battlefield. The soldiers may be equipped with more powerful data transmitters (out of band higher-range radios) than sensors. It may be more effective to forward the information (e.g. enemy detection, land mine presence, convoy vehicles) to the nearest available soldier, who can forward it to the other soldiers, instead of sending it to all soldiers in the field. In a disaster recovery operation, several biochemical sensors may have been scattered, and multiple imagers (aerial or robotic) may be navigating the terrain. When biochemical sensors detect a toxic plume, this message just needs to go to the nearest imager (rather than a specific imager) which can act accordingly.

1.3 The Problem: Hybrid Sensor Network Deployment

In this paper, we investigate some fundamental questions on hybrid sensor network deployment to support anycast communication.

- *Given a number of microservers, how does the placement of them affect the life time of network?* We propose a cost model for hybrid sensor networks, and formulate it into an integer linear programming optimization problem. We then introduce an approximation algorithm using tabu-search technique.

- *What is the benefit of introducing additional microservers into network? Is it cost effective to introduce these extra microservers?* To achieve optimal financial cost, we show that the number of microservers are different when the cost ratio of sensor and microserver are different.

The rest of this paper is organized as follows. Section 2 provides an overview of the anycast communication model which motivates the network deployment problem described in the paper. Section 3 proposes an integer linear programming formulation of the network deployment problem. Section 4 introduces a tabu-search algorithm to solve the problem efficiently. Section 5 presents an analysis to compare the life-time differences and a cost analysis of different scenarios. Section 6 discusses our conclusions.

2 Tree-Based Data Anycast

In this section, we provide an overview of our anycast mechanism which motivates the network deployment problem addressed in this paper.

We assume a *hybrid* sensor network which consists of both resource-rich micro-server nodes and low-power sensor nodes. Further we assume that there are multiple micro-servers (sinks) interested in the same data. Data needs to only reach one sink, thus motivating an anycast service. We assume that sensor network applications can handle small amount of data loss; and therefore anycast does not need to explicitly provide reliable data delivery.

We want to provide an anycast service that is scalable, self-organizing, robust, simple and energy-efficient. To implement this, we adopted a shared tree approach. Corresponding to each event source, a *shortest-path tree* rooted at the source is constructed. Sinks form the leaves of the tree. Sinks can dynamically join or leave the anycast tree. Although this approach requires more network state, it is a good approach to handling dynamics, as it simultaneously maintains paths to all sinks. By eliminating the need to discover paths to alternate sinks each time a sink leaves, it can reduce worst-case latency (when sinks fail) and does not require synchronization among sinks. Figure 1 illustrates how the structure of each anycast tree evolves when two sinks join and leave a sensor network. Details of the anycast mechanism are described in paper [10].

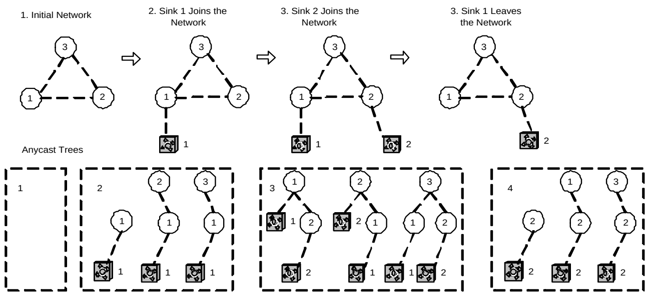

Figure 1: Illustration of the anycast mechanism. The lower, boxed pictures show the structure of each anycast tree as two sinks join and leave a sensor network.

An important metric in determining the performance of the anycast scheme is the number and placement of microservers (resource-rich nodes), relative to low-powered sensor nodes. The number of microservers must be sufficient to meet system lifetime objectives, as well as other application-governed objectives (e.g., message delivery latency), without exceeding resource cost thresholds. Moreover, the number of microservers chosen depends on parameters such as the occurrence pattern (frequency, spatial distribution) of sensor events in the system. In the next section, we propose a problem formulation for resource provisioning, *i.e.,* placement of microservers and sensors, incorporating all these factors.

3 Cost Model and Optimization

In this section, we propose a cost model for energy usage in a hybrid sensor network. Without loss of generality, we assume that the coverage area of the network is a rectangle whose area is A. We divide the area into a number of grids, the area of which (a)is chosen according to the transmission range of the sensor, such that at least one sensor is required per grid to maintain network connectivity and coverage. The total number of grids is n $= \lceil A/a \rceil$.

Therefore, the problem can be formulated as follows:
Given a set of locations $N = \{1, ..., n\}$, where should we deploy a number of sensors and microservers? For each location $i \in N$, a number of events r_i

happens within each time unit. To sense an event, it costs e_1 units of energy per sensor and E_1 units of energy per microserver. To forward the data packets of an event, it requires e_2 units of energy for a sensor and E_2 units of energy for a microserver. To maintain network coverage and connectivity, there must be one sensor or microserver at each location.

Each sensor can store B^{micro} units of energy, and each microserver can store B^{macro} units of energy.

We further define an indication function γ_{ij}^k as follows:

$$\gamma_{ij}^k = \begin{cases} 1 & \text{if the transmission for device (sensor or microserver) at grid } i \text{ to} \\ & \text{device at grid j uses device at grid k as an intermediate} \\ & \text{forwarder, includes } i \text{ but excludes } j \\ 0 & \text{otherwise} \end{cases}$$

The values of γ_{ij}^k depend on the network's routing algorithm (e.g., tree-based anycast) and can be calculated in advance. d_{ij} is the distance (hop-count) between grid i and grid j. Similarly, it can be calculated in advance.

Moreover, we define the following decision variables: x_i as:

$$x_i = \begin{cases} 1 & \text{if the device at grid } i \text{ is a normal sensor} \\ 0 & \text{otherwise (microserver)} \end{cases}$$

;and z_{ij} as:

$$z_{ij} = \begin{cases} 1 & \text{if sensor at grid } j \text{ is the closest microserver to sensor at grid } i \\ 0 & \text{otherwise} \end{cases}$$

.

Given the number of microservers (M), the objective of the optimization is to maximize the life time of the network by placing the microservers in optimal locations. Defined λ as $(\frac{1}{L})$ where L is life time of the network, the problem can be formulated as:

$$Minimize \quad \lambda \tag{1}$$

Subject to:

- The total energy that a sensor or a microserver at a grid (k) can consume within its life time L_k can not exceed the total energy they can store. ($\lambda_k = \frac{1}{L_k}$) Constraint (2) holds when the device at grid k is a sensor; otherwise, constraint (3) holds.

$$r_k e_1 x_k + \sum_{i=1}^{N}\sum_{j=1}^{N}(\gamma_{ij}^k r_i z_{ij}) e_2 x_k - B^{micro}\lambda_k \leq 0, \forall k \quad (2)$$

$$r_k E_1 - r_k E_1 x_k + \sum_{i=1}^{N}(r_i z_{ik}) E_2 (1 - x_k) - B^{macro}\lambda_k \leq 0, \forall k \quad (3)$$

- To keep the model linear, we define w_{ij}^k as $z_{ij}(1 - x_k)$. Constraints (4, 5, 6) limit z_{ij} equal to one only if microserver at grid j is the nearest one to the sensor at grid i (since anycast routes data to the nearest microserver). Therefore, constraint (7) actually is a redundant constraint, but we need it to make constraint (2) linear later.

$$d_{ij} w_{ij}^k \leq d_{ik} - d_{ik} x_k, \forall i, j, k \quad (4)$$

$$w_{ij}^k \leq z_{ij}, \forall i, j, k \quad (5)$$

$$z_{ij} - x_k \leq w_{ij}^k, \forall i, j, k \quad (6)$$

$$\gamma_{ij}^k z_{ij} - x_k \leq 0, \forall i, j, k \quad (7)$$

- There are M microservers in the network.

$$\sum_{i=1}^{N} x_i = N - M \quad (8)$$

- Only a microserver can be the end point (sink) of disseminated data.

$$z_{ij} - 1 + x_j \leq 0, \forall i, j \quad (9)$$

- A sensor only send packets to one microserver.

$$\sum_{j=1}^{N} z_{ij} = 1, \forall i \quad (10)$$

γ_{ij}^k	z_{ij}	x_k	$\gamma_{ij}^k z_{ij} x_k$	$\gamma_{ij}^k z_{ij}$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	1
1	1	1	1	1

Table 1: The values of $\gamma_{ij}^k z_{ij} x_k$ and $\gamma_{ij}^k z_{ij}$. They have different values only at row 7.

- Network life time equals to the life time of the sensor whose energy drys up first.

$$\lambda \geq \lambda_i, \forall i \tag{11}$$

- The scopes of variables x_i, z_{ij} and w_{ij}^k.

$$x_i \in \{0, 1\}, \forall i \tag{12}$$

$$z_{ij} \in \{0, 1\}, \forall i, j \tag{13}$$

$$w_{ij}^k \in \{0, 1\}, \forall i, j, k \tag{14}$$

From the optimization model, it is clear that the problem is that, given a number of microservers, trys to maximize the life time of network by placing microservers to optimal locations.

Constraints (2, 3) are not linear because they involve multiplication of decision variables x_k and z_{ij}. These constraints can however be replaced by equivalent linear constraints as follows.

Consider the values of $\gamma_{ij}^k z_{ij} x_k$ and $\gamma_{ij}^k z_{ij}$ (see Table 1):

The only difference between them is at row 7, and constraint (7) excludes this situation from occurring. Therefore, we can redefine constraint (2) as:

$$r_k e_1 x_k + \sum_{i=1}^{N} \sum_{j=1}^{N} (\gamma_{ij}^k r_i z_{ij}) e_2 - B^{micro} \lambda_k \leq 0, \forall k \tag{15}$$

Similarly, applying constraint (9), constraint (3) can be redefined as:

$$r_k E_1 - r_k E_1 x_k + \sum_{i=1}^{N} (r_i z_{ik}) E_2 - B^{macro} \lambda_k \leq 0, \forall k \tag{16}$$

4 A Tabu Search Algorithm

The model introduced in section 3 is a complicated combination problem that depends on the number of grids and the number of microservers. There are $N!/((N-M)!M!)$ combinations in total. From experiments, we find that the maximum number of grids that the commercial optimization package CPLEX [11] can handle efficiently is 20. Therefore, results produced by CPLEX are not very helpful for the deployment a reasonable size network.

Integer linear programming solution does not scale with network size, thus we introduce a tabu search [12] algorithm which provides an approximation to the optimal result of the model within reasonable computation-time.

4.1 Neighborhood

The neighborhood of a microserver's grid is defined as all other grids in the topology, namely:

$$N_k = \{1, 2, ..., k-1, k+1, ...n\} \tag{17}$$

4.2 Tabu Search

Our tabu-search algorithm (Figure 2) defines two tabu lists. The first one records the grids that microservers can not move to for a number of iterations I_t. The second one records the grids that microservers can not leave for another number of iterations I_f. The value of I_t and I_f should be large enough to avoid cycles (we tuned them as $3/4 \times N$ and $1/2 \times M$ respectively

```
int tsStable = 0;
int stabilityLimit = 500;

while(tsStale < stabiliyLimit) {

        if(bestGain(x, best, obj) >= 0) {   //intensification
                randomMoveOneOfTheBest(x);
        } else {                                        //diversification
                randomMoveAllMicroservers(x);
    }
    if(obj > best) {                    //better result found
        best = obj;
        tsStable = 0;
    } else {
        tsStable = tsStable + 1;
    }

    update_tabu_list(tabu_list_from, tabu_list_to);
}

bestGain(x, best, obj) {
        old = obj;
        soFarBest = -1;

        for each neighbour of current microservers {
                getlifetime(x, obj);

                if(obj > best) {           //aspiration level condition
                        update(x);
                        soFarBest = obj;
                } else if(intabulist(x)) {
                        continue;
                } else {
                        if(obj > soFarBest)
                                soFarBest = obj;
                }
        }

        return old - obj;
}
```

Figure 2: A Tabu-search Algorithm for Sensor Network Life-time Optimization Model.

Number of	Life-time		Computation Time	
Microservers	Cplex	Tabu-search	Cplex	Tabu-search
1	16901	16901	1m55.937s	0m0.155s
2	22641	22641	10m33.740s	0m0.511s
3	25531	25531	15m0.498s	0m0.991s
4	25531	25531	12m12.218s	0m2.217s
5	25531	25531	26m58.37s	0m8.754s
6	29268	29268	5m42.946s	0m40.711s

Figure 3: Results of CPLEX and Tabu-search algorithm at a 20 grid network

in our experiments).

The algorithm tries to find out a local maximum by calculating the life-time of each possible single move in intensification stage. While the gain is negative, the algorithm explores the unexplored area in diversification stage by random movement. Note that it will not move to recent locations since they are recorded in tabu-lists unless aspiration level condition is satisfied. The aspiration level condition is defined as a new best life-time found. The algorithm terminates when the objective function has not improved for the number of *stabilityLimit* iterations. The *stabilityLimit* parameter is defined as a large integer (e.g., 500) to ensure the robustness of the algorithm.

4.3 Algorithm Benchmark

To validate the tabu-search algorithm, we compared it with the commercial optimization package CPLEX at a 20 grid network (the maximum CPLEX can handle efficiently). Results (Figure 3) showed that our tabu-search algorithm achieved the same optimal results as CPLEX, but in a much more efficient manner.

5 Results and Analysis

In this section, we use our tabu-search algorithm to solve the mathematical model introduced in section 3, and study the behavior of a specific sensor network. We also propose a financial cost analysis model to determine the most cost effective combination of hybrid sensors in a network.

In this case, the network is divided into 100 equal size grids. The initial energy of a sensor is 6,000 Joule (equals to 2 AA batteries) and the initial energy of a microserver is 60,000 Joule; there are 5 events taking place at each grid within each time unit; it takes either 35 mJ for a sensor or 25 mJ for a microserver to sense/handle an event; it takes 6 mJ for either a sensor or a microserver to transfer the packets generated by an event [9].

Figure 4 plots both the optimal and the worst case life time of a 100 grid network by the number of microservers. Network life time improves by more than 60% after the second microserver is added. *The life time will not increase after the number of microserver reaches a threshold until there is a microserver at each grid.* [2]

The best microserver placements can extend network life-time by more than 500% comparing to the worst microserver placements (when there are nine microservers deployed). The life-time of random microserver placements should be somewhere between the best placements and the worst placements. Figure 5 shows the microservers' locations in the grid.

To consider the financial benefits of adding an additional microserver to the network, we use equation (18) to normalize the network life time against network cost.

$$L_M = \frac{L}{(N-M)c_s + Mkc_s} \tag{18}$$

where L is network life time, $N-M$ is the number of sensors, c_s is the cost of sensor, M is the number of microservers, and kc_s is the cost of microserver. k represents the ratio of the cost between a microserver and a sensor.

To achieve maximum financial benefits, the results $(\frac{L_M}{L_1})$(see Figure 6) show that different number of microservers should be deployed as the values of k change. For example, if $k = 10$, the life-time of network can be extended by more than 110% at the same normalized financial cost if six microservers are deployed comparing to just one microserver is deployed; if $k = 50$, the life-time of network can be extended by more than 20% at the same normalized financial cost if three microservers are deployed comparing to just one microserver is deployed. Not surprisingly, the benefits decrease as

[2]This is not surprising, given the prevalence of critical density threshold phenomena in wireless network design [13].

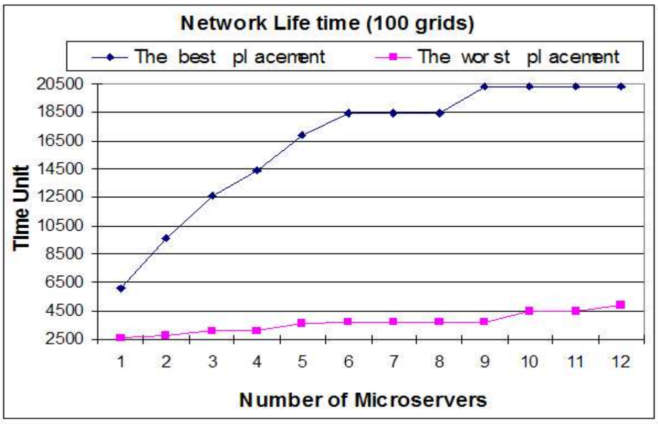

Figure 4: Network life time of a 100 grid network with different number of microservers

the value of k increases (while microserver becomes much more expensive than sensor).

6 Conclusions

In this paper, we considered the problem of network deployment for hybrid sensor networks, consisting of both resource-rich and resource-impoverished sensor devices. The resource-rich nodes are more expensive but provide significantly enhanced functionality (storage, memory, computation, energy, communication bandwidth, and other specialized functions). Such hybrid sensor networks have the potential to support the higher bandwidth communications of broadband sensor networking applications, as well as the fine-grained sensing possible by smaller sensor devices.

We proposed an integer linear programming formulation and introduce a tabu-search algorithm to answer some fundamental questions related to hybrid sensor network deployment — for a given number of microservers, what is the maximum lifetime of a sensor network and what is the optimal microserver placement? What benefit can additional microservers add to the network, and how cost-effective is it to introduce these microservers? We also propose a normalized cost model that balances the benefits with deployment costs. A case study showed how an optimal deployment can be

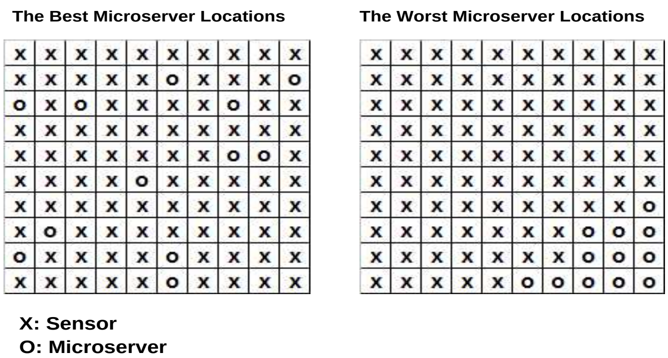

Figure 5: The best and worst placements of microservers in a 100 grid network

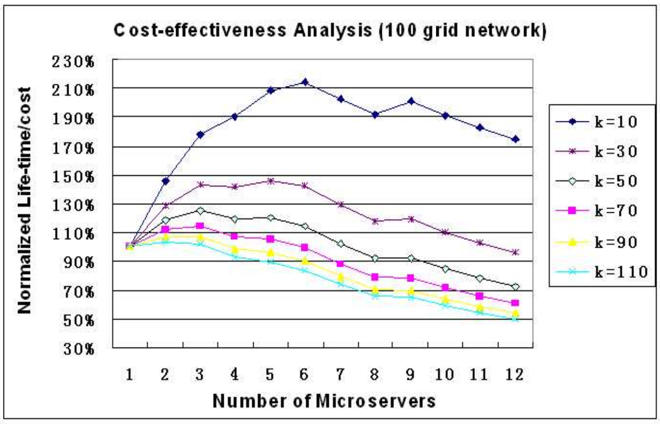

Figure 6: The cost normalized life time of a 100 grid network

achieved.

Our studies showed that network life time could be extended more than 60% by adding an extra microserver to the network; the network lifetime with optimized microserver placement can be 500% (or more) better than the worst case lifetime. We also proposed a cost model and showed that an optimal normalize-cost can be achieved.

References

[1] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister, "System architecture directions for networked sensors," in *Proceedings of the ninth international conference on Architectural support for programming languages and operating systems.* 2000, pp. 93–104, ACM Press.

[2] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong, "Tag: a tiny aggregation service for ad-hoc sensor networks," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 131–146, 2002.

[3] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 1, pp. 2–16, 2003.

[4] Fan Ye, Haiyun Luo, Jerry Cheng, Songwu Lu, and Lixia Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks," in *Proceedings of the eighth annual international conference on Mobile computing and networking.* 2002, pp. 148–159, ACM Press.

[5] Juan Liu, Feng Zhao, and Dragan Petrovic, "Information-directed routing in ad hoc sensor networks," in *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications.* 2003, pp. 88–97, ACM Press.

[6] 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," in *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, 2002.

[7] A. Howard, M. Matadd, and G. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *algorithm for mobile sensor networks. Autonomous Robots, Special Issue on Intelligent Embedded Systems*, 2001.

[8] Nojeong Heo and P.K. Varshney, "An intelligent deployment and clustering algorithm for a distributed mobile sensor network," pp. 4576 – 4581, 2003.

[9] Ram Kumar, Vlasios Tsiatsis, and Mani B. Srivastava, "Computation hierarchy for in-network processing," in *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications.* 2003, pp. 68–77, ACM Press.

[10] Wen Hu, Nirupama Bulusu, and Sanjay Jha, "A communication paradigm for hybrid sensor/actuator networks," in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2004)*, 2004.

[11] "http://www.ilog.com," .

[12] F. Glover and M. Laguna, "Tabu search," *In Handbook of Applied Optimization, P. M. Pardalos and M. G. C. Resende (Eds.)*, pp. 194 – 208, 2002.

[13] Bhaskar Krishnamachari, Stephen B. Wicker, and Ramon Bejar, "Phase transition phenomena in wireless ad-hoc networks," in *Proceedings of Symposium on Ad-Hoc Wireless Networks, IEEE Globecom*, 2001.