Iterative Node Deployment in an Unknown Environment

A. Gueye, S. Coleri Ergen and A. Sangiovanni-Vincentelli

***Abstract*—We consider the problem of deploying relay nodes to achieve connectivity with minimum cost in a sensor network of unknown radio propagation characteristics. For a network where a certain number of targets or sensing nodes have already been deployed in fixed and known positions, we aim at efficiently adding communication or relay nodes to guarantee connectivity with minimum cost, between any sensor node and a base station. The communication cost of a wireless link is defined as the expected number of retransmissions over that link and is modeled using an underlying Gaussian Process (GP) between the nodes. We propose an iterative sensor deployment approach that learns the parameters of the underlying GP while deploying the additional nodes in the best positions possible at each step. Our deployment algorithm is more powerful with respect to the ones found in literature since: 1) we do not assume fixed communication range, i.e., we do not assume that nodes can perfectly communicate within a fixed range and will not communicate at all outside that range (this assumption is not realistic for the wireless channel); 2) we do not assume the existence of a pilot deployment aimed at learning the radio propagation characteristics because of the high cost of the deployment process and of the sensor nodes themselves.**

***Index Terms*—Sensor Network, Deployment, Gaussian Processes**

I. Introduction

Wireless sensor networks (WSN) consist of spatially distributed, autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, voltage, current, at different physical locations. Many application areas such as environmental monitoring, health care, home automation and traffic control, benefit from WSNs. These networks typically include a base station (BS), several sensor nodes that generate data for transfer to BS and relay nodes that act as routers to guarantee connectivity between the sensor nodes and the BS.

The problem of node deployment in wireless sensor networks has been extensively studied in the sensor network community. In this activity, the goal is to find the best placement, particularly of the nodes which have a routing function, that optimizes some cost related to the specific application, such as number of sensors, communication cost, and coverage. Existing solutions are based on one of two approaches: a classical approach based on the assumption of a "communication range" model and a more recent approach that considers a "communication cost". The "communication range" assumption states that the network nodes can perfectly communicate within a fixed range, and cannot communicate outside that range [1], [2]. The "communication cost" model considers the probabilistic nature of the wireless communication channel, and introduces a communication cost that depends on the probability distribution of the channel. The probability distribution is assumed to be either known or that there exists a pilot node deployment from which the probability distribution can be learned [3], [4].

The original contribution of this paper is an iterative node deployment approach that eliminates the unrealistic "communication range" assumption and costly pilot deployment. The proposed method is based on iterating between placing a certain number of nodes and estimating the parameters indicative of the quality of the communications in the network. The communication cost is modeled using an underlying Gaussian Process (GP) as proposed in [4]. Starting from an initial placement of a limited number of nodes, the proposed iterative method makes a first estimation of the GP. Additional nodes are then incrementally placed at each iteration, their positions being determined based on the estimation of the GP parameters; the estimation of these communication quality parameters being updated after each placement of additional nodes, based on monitoring communication activities between the nodes in the network.

The remainder of this paper is organized as follows. Section II describes the system model. The proposed method is outlined in Section III with corresponding mathematical details given in Section IV. Section V provides the results of algorithm implementation for real-world data. Section VI concludes the paper.

II. System Model

Figure 1 gives a schematic description of the scenario under consideration. Wireless sensor networks comprise a BS, several sensor nodes and relay nodes. Each sensor node has one or more sensors to measure physical quantities such as temperature, voltage and a wireless transceiver to communicate with other nodes of the network. The sensor nodes are placed at various different locations of a selected environment depending on the monitoring requirements. The relay nodes

A. Gueye and A. Sangiovanni-Vincentelli are with the University of California at Berkeley, EECS Department, e-mail: {agueye|alberto}@eecs.berkeley.edu. S. Coleri Ergen is with the WSN Lab, sponsored by Pirelli and Telecom Italia, Berkeley, CA, e-mail: sinem.ergen@wsnlabberkeley.com.

978-1-4244-4148-8/09/$25.00 ©2009

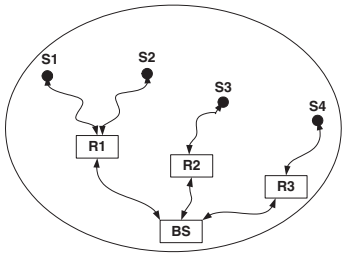


Fig. 1. WSN including sensor nodes, relay nodes, and base station

in the sensor network on the other hand act as wireless routers between the sensor nodes and the BS. The relay nodes have the function of guaranteeing the connectivity between the sensor nodes and the BS. The BS interfaces the wireless sensor network with an external network.

The communication channel is modeled using Gaussian Process (GP) for Classification ([5], [6], [7],[8]). GP is used because it allows modeling the correlations between link qualities among different locations in space and eases computation. This model has been justified in [4] for pilot deployment-based placement. An alternative method is modeling an underlying Rayleigh physical channel and mapping it to a logical channel for the packet loss. However, this option is too hard to analyze. In the GP model, the probability of successfully transmitting a data packet from the network node s (either a sensor node or a relay node) to the node t, is a function of an underlying Gaussian process W of the wireless communications field. Thus, given the underlying Gaussian $W_{s,t}$ of the link (s,t), the probability of successfully sending a data packet over that link is:

$$\theta_{s,t}(w) = \mathbb{P}(Y_{s,t} = 1 | W_{s,t} = w) = g(1, w) \qquad (1)$$

where $Y_{s,t}$ is the indicator random variable that takes value 1 if a data packet is successfully sent from node s to node t, and -1 otherwise (i.e. in case of transmission failure). The function $g(\cdot,\cdot)$ which lies in the interval [0,1] is called the *likelihood function*, which in this paper is assumed to be equal to the *probit* ([9]) or cumulative distribution of a standard normal random variable*: $g(y,w) = \Psi(yw)$, where:

$$\Psi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt \qquad (2)$$

As observed in [4], the communication cost $C(s,t)$ between the nodes s and t, which is defined as the expected number of retransmissions needed to successfully send the data packet, can be computed as:

$$C(s,t) = \int_w \frac{1}{\theta_{s,t}(w)} f_{W_{s,t}}(w) dw \qquad (3)$$

where $f_{W_{s,t}}(w)$ is the probability distribution function of the underlying random variable $W_{s,t}$, which is assumed to be Gaussian. Note that minimizing the total communication cost is equivalent to minimizing the total energy consumed in packet transmissions. The parameters of the distribution $f_W(\cdot)$ are however in general not known, so they have to be estimated by observing the communications between the nodes in the field.

*Other models such as the logit [10] and the step-function have been adopted in the literature

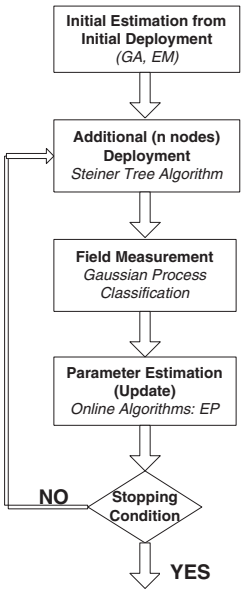


Fig. 2. Flowchart of the Iterative Deployment Algorithm

Considering the scenario and the model presented above, we are interested in the problem of deploying relay nodes to minimize the communication cost between the sensor nodes and the BS. We do not assume any prior knowledge about the statistical characteristics of the communication channel, neither do we assume the existence of a pilot deployment to learn such channel. Instead, we are looking for methods that solely use advantage of the communication in the field to learn the channel and find the best positions for the nodes to be deployed. We propose an iterative solution that is presented next.

III. The Iterative Sensor Deployment Algorithm

A schematic flowchart of the proposed method is shown in Figure 2. The proposed iterative procedure alternates between placing a fixed number of nodes, particularly nodes having a routing function (relay nodes), and updating the knowledge about the quality of the communications in the network. Hereinafter, an overview of the main steps of the method will be presented. The individual method steps will be described in detail in Section IV.

1) The first estimation of the wireless communication channel is computed based on the measurements from the initial node deployment using a Maximum Likelihood (ML) criterion.

978-1-4244-4148-8/09/$25.00 ©2009

2) Once the parameters of the underlying Gaussian process are estimated, n additional relay nodes are deployed to improve the communication cost between the sensor nodes and the BS.
3) After the placement of the additional nodes, communication activities between the nodes in the field are monitored and recorded. These new measurements are then used to update the estimate of the communication channel parameters.
4) The parameters of the underlying Gaussian process model are updated using Online Learning techniques.
5) Steps 2,3, and 4 are iterated until a stopping condition is reached. The stopping condition may be either reaching a predetermined maximum number of deployable relay nodes or attaining a minimum cost for sending the packets from the sensor nodes to the BS.

IV. Detailed Analysis of the Algorithm

We assume that a certain number of transmissions in the field have been observed. For the observed communication links, the values $\mathbf{Y} = [Y^{(1)}, \ldots, Y^{(n)}]$ are recorded, where $Y^{(i)} = (y_1^{(i)}, \ldots, y_m^{(i)})$ $i = 1, \ldots, N$ are independent, $y_j^{(i)}$ indicates whether the data packet in the i-th transmission in communication link j was received or not, m is the number of observed communication links. Observations are therefore available only from the first m links $(1, ..., m)$, links $(m + 1, ..., m + n)$ being not observed.

Given $\mathbf{Y}$, it would be desirable to compute the predictive communication cost $\bar{C}(s,t)$, which requires computing the posterior distribution of $W_{s,t}$ for any link (s,t). For that, one needs to first estimate the parameters $\Theta = (\mu, \Sigma)$ of the underlying Gaussian process given the observations $\mathbf{Y}$.

In this paper, a Maximum Likelihood (ML) criterion is used to estimate the parameters of the posterior distribution of $W = (w_1, \ldots, w_m, w_{m+1}, \ldots, w_{m+n})$ given $\mathbf{Y} = (Y^{(1)}, \ldots, Y^{(N)})$. The ML estimate can be computed as:

$$\hat{\Theta} = \arg\max_{\Theta} \left(\mathbb{E}_W \left[\mathbb{P}(\mathbf{Y}, \mathbf{W}|\Theta)\right]\right) \quad (4)$$

$$= \arg\max_{\Theta} \log\left(\mathbb{E}_W \left[\mathbb{P}(\mathbf{Y}, \mathbf{W}|\Theta)\right]\right) \quad (5)$$

where $\Theta = (\mu, \Sigma)$ are the parameters of the Gaussian process, mean and covariance functions, $\mathbf{W} = (W^{(1)}, \ldots, W^{(N)})$, where $W^{(i)} = (w_1^{(i)}, \ldots, w_m^{(i)}, w_{m+1}^{(i)}, \ldots, w_{m+n}^{(i)})$ are independent realizations of $W = (w_1, \ldots, w_m, w_{m+1}, \ldots, w_{m+n})$, and $y_j^{(i)}$ depends on the unobserved random variable $w_j^{(i)}$.

A. The Expectation Maximization (EM) Algorithm

Determining the exact ML estimate is computationally intractable so the Expectation-Maximization (EM) algorithm is used to compute an approximation. The EM algorithm ([11], [12]) iteratively maximizes a lower bound of the log-likelihood obtained by using Jensen's inequality:

$$\mathbb{E}_{\mathbf{W}}\left[\log\left(\mathbb{P}(\mathbf{Y}, \mathbf{W}|\Theta)\right)\right] \leq \log\left(\mathbb{E}_{\mathbf{W}}\left[\mathbb{P}(\mathbf{Y}, \mathbf{W}|\Theta)\right]\right) \quad (6)$$

The EM algorithm computes an approximation of the solution by alternating between an E-Step (Expectation Step), where an expectation of the likelihood is computed based on the current estimate of the parameters, and an M-Step (Maximization Step), where this expectation is maximized to update the parameter estimation.

In the E-Step, assuming a current estimate Θ_t of the parameters at time t, an expectation of the likelihood is computed as:

$$Q(\Theta, \Theta_t) = \mathbb{E}_{\mathbf{W}|\Theta_t, \mathbf{Y}}\left[\log\left(\mathbb{P}(\mathbf{Y}, \mathbf{W}|\Theta)\right)|\mathbf{Y}, \Theta_t\right] \quad (7)$$

where the expectation is taken with respect to the posterior distribution of W given Y and the prior distribution of W with parameters Θ_t . Once $Q(\Theta, \Theta_t)$ is computed, an update of the estimate is obtained in the M-Step by maximizing it over all the values of Θ .

$$\Theta_{t+1} = \arg\max_{\Theta}\left(Q(\Theta, \Theta_t)\right) \quad (8)$$

Theorem 1: If the posterior distributions $\mathbb{P}(W^{(i)}|\Theta, Y^{(i)})^{\dagger}$, can be approximated with Gaussian distributions with parameters $\hat{\Theta}_t^{(i)} = (\hat{\mu}_t^{(i)}, \hat{\Sigma}_t^{(i)})$, then the updates in the E-Step and the M-Step can be written as:

- **E-Step:**

$$\begin{aligned}\hat{\mu}_t^{(i)} &= \mu_t + \Sigma_t(\bar{Y}^{(i)})^T A^{(i)} \bar{\alpha}^{(i)}(\Theta_t) \\ \hat{\Sigma}_t^{(i)} &= \Sigma_t - \Sigma_t(\bar{Y}^{(i)})^T A^{(i)} \Omega^{(i)}(\Theta) A^{(i)} \bar{Y}^{(i)} \Sigma_t\end{aligned}$$

where

$$\Omega^{(i)}(\Theta) = (A^{(i)})^{-1} + \bar{\alpha}^{(i)}(\Theta_t)\left(\bar{\alpha}^{(i)}(\Theta_t)\right)^T - \Upsilon^{(i)}(\Theta_t)$$

The matrices $A^{(i)}$, $i = 1, \ldots, N$ are given by:

$$A^{(i)} = \left(I + \bar{Y}^{(i)} \Sigma_t (\bar{Y}^{(i)})^T\right)^{-1}$$

The vectors $\bar{\alpha}^{(i)}(\cdot)$ and the matrices $\Upsilon^{(i)}(\cdot)$ are given by

$$\bar{\alpha}^{(i)}(\Theta) = \frac{\int_{\mathcal{U}^{(i)}} U \exp\left\{-\frac{1}{2}U^T U\right\} dU}{\int_{\mathcal{U}^{(i)}} \exp\left\{-\frac{1}{2}U^T U\right\} dU} \quad (9)$$

$$\Upsilon^{(i)}(\Theta) = \frac{\int_{\mathcal{U}^{(i)}} UU^T \exp\left\{-\frac{1}{2}U^T U\right\} dU}{\int_{\mathcal{U}^{(i)}} \exp\left\{-\frac{1}{2}U^T U\right\} dU} \quad (10)$$

The sets over which the integrals are taken are given by:

$$\mathcal{U}^{(i)} = \left\{U \in \mathbb{R}^m | \left(\left(A^{(i)}\right)^{-1/2} U\right)_j > -y_j^{(i)} \mu_j, j = 1, \ldots, m\right\}$$

where $\bar{Y}^{(i)} = \left[diag(y_1^{(i)}, \ldots, y_m^{(i)}), 0_{m,n}\right]$ is defined to be the matrix composed of the $m - by - m$ diagonal matrix having the measurement data $y_j^{(i)}$ in its diagonal and the $m - by - n$ null matrix. The index $j = 1, ..., m$ designates the link number while the index $i = 1, .., N$ denotes the measurement instance.

†These distributions are not Gaussian. However, their projections on the Gaussian family are good approximations

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

- **M-Step:**
  The M-Step updates (Equation 8) $\Theta_{t+1} = (\mu_{t+1}, \Sigma_{t+1})$ are given by:

$$\mu_{t+1} = \frac{1}{N}\sum_{i=1}^{N} \hat{\mu}_t^{(i)} \quad (11)$$

$$\Sigma_{t+1} = \frac{1}{N}\sum_{i=1}^{N} \hat{\Sigma}_t^{(i)} + (\hat{\mu}_t^{(i)} - \mu_{t+1})(\hat{\mu}_t^{(i)} - \mu_{t+1})^T \quad (12)$$

Proof: Two ingredients have been used for the computation of the closed-form expressions given above: 1) the properties of Gaussian integrals [13], and 2) the observation that the probit function (the cdf of a standard normal random variable in Equation 2) can be rewritten as:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \delta(u \geq 0) e^{-(u-x)^2/2} du \quad (13)$$

Due to space limitation, the full proof is not available in this paper. The interested reader is referred to [14]. ■

Note that the function in Equation 5 is not concave so the EM algorithm might not converge to a global maximum. The EM algorithm is therefore run multiple times, each starting with a different initial point to try to obtain results close to a global maximum.

B. Deployment of Additional Relay Nodes

Once the parameters of the underlying Gaussian process have been learned and updated, n additional relay nodes are deployed to improve the communication cost between the sensor nodes and the BS. Choosing the new n positions that minimizes the communication cost is equivalent to solving a Minimum Cost Steiner Tree problem with the additional constraint that the number of Steiner points is at most n. Such problems are known to be NP-hard [15]. Several approximation algorithms have been proposed in the literature ([16], [17], [18],[19],[20]). As an illustration of our algorithm, we use a heuristic approximation proposed in [20].

The algorithm is based on constructing the complete undirected graph on the sensor nodes and the BS where the distance between any two nodes is equal to the cost of the shortest path in the original graph. Once a complete graph is obtained, a minimum spanning tree algorithm is run on it, then the resulting spanning tree is replaced by its corresponding subgraph in the original graph.This algorithm has a worst case time complexity of $\mathcal{O}(|S||V|^2)$ where S is the set of sensor nodes and V the set of all nodes in the network. The performance ratio of the algorithm is proven to be upper bounded by at most 2. If the number of relay nodes to be added at each step (n) is less than the number of Steiner points (s_t) added by the algorithm one should remove the relay nodes one at a time down to n. At each step, a node is removed such that the cost of the remaining graph is minimized. Removing the relay nodes decrease the performance of the algorithm. However, in most of the cases of interest, n is bigger than s_t and no removal is necessary as illustrated in Section V.

If the algorithm only makes use of the observed information, the choice of relay nodes may be biased towards the observed links. Therefore, in addition to the communication cost defined in Equation 3, an additional penalty is introduced in the cost of the links to force the algorithm to explore new positions, i.e. a constant penalty of $-\lambda$ is added if the link is observed. λ determines the degree of exploration compared to exploitation.

C. Online Learning: Expectation Propagation (EP) Algorithm

As new observations are made, the new value estimates of the Gaussian process parameters need to be computed. For this purpose, the EM methods for computing a Maximum Likelihood Estimator (MLE) can be used. However, to compute the best MLE, it would be necessary to consider both old and new observations.

Using MLE requires storing a large amount of data, which becomes an issue as more data become available. Furthermore, this method ignores the old estimates and computes completely new ones based on the entire data set. Online learning techniques ([21]) were therefore selected to generate approximated update algorithms that use the old estimate and need not store the old data.

Let $D^{(t)}$ be the data observed up to time t and Θ_t be the estimate of the parameters at time t computed from the data $D^{(t)}$. The goal is to find a way to update the parameter estimate Θ_{t+1} after a new observation $\mathbf{Y}^{t+1} = (Y_1^{t+1}, \ldots, Y_N^{t+1})$ is made. Here, superscripts denote the iteration number in the node deployment procedure and subscripts denote the instance number of observations within that iteration. $Y_i^{(t)}$ is a vector containing the information about whether there is a success or failure of transmission in each communication link at the i-th instance of t-th iteration.

To find the update, the GP model can be considered as a parameterized Bayesian model where the parameter (the Gaussian random process) has infinite dimension. Having an old posterior distribution of $P(W|D^{(t)})$, it is possible to compute the new posterior distribution $P(W|D^{(t+1)})$ as a function of $P(W|D^{(t)})$ by applying the Bayes' rule. Directly applying the Bayes' rule requires the knowledge about the different field realizations $W_1^{(t+1)}, \ldots, W_N^{(t+1)}$ that generate the observations $Y_1^{(t+1)}, \ldots, Y_N^{(t+1)}$ respectively. Trying to compute these values may be very demanding. Thus, the posterior distribution is derived based on the idea by Minka in [22]. The idea works as follows.

The prior distribution of W is assumed to have a Gaussian distribution, which may be computed from the initial deployment. The data at each iteration is used to update the posterior distribution which is approximated by another Gaussian distribution. The posterior distribution at the end of

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

number links	17	20	22	38	50
runtime (hours)	2.8030	4.1422	5.5876	13.7016	21.8164

TABLE I
SIMULATION TIME FOR DIFFERENT NUMBER OF LINKS ON 2 QUAD CORE INTEL XEON CPU AND 64GB MEMORY

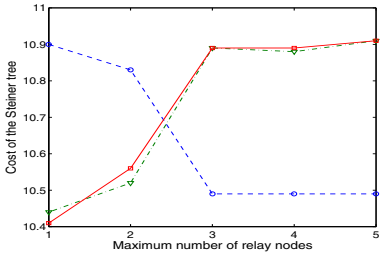


Fig. 4. Cost of the tree as a function of n (maximum number of relay nodes) and λ.

each iteration is used as a prior distribution for the following update.

D. Computation Cost

Deriving an analytical computational cost for this algorithm is challenging mainly because of the Gaussian integrals that are involved. Instead, we show the simulation time for deploying 2 nodes at a time in a network containing different number of links in Table I as an empirical complexity measure. The computation cost is almost linear in the number of links.

V. EXPERIMENTAL RESULTS

We tested the iterative deployment algorithm on the data collected from the sensor network at the Architecture department of Carnegie Mellon University [4] ‡. The samples were taken every 2 minutes, starting roughly at 5 pm, for a total measurement time of about 20 hours. Every two minutes, a packet is sent over each channel, and it is recorded whether it was successfully received (1) or not (0). Note that the data used in this paper is a subset of the measurement data gathered at CMU. In the scenario under consideration there is 1 base station (BS), 6 sensor nodes (S1-6), 7 possible relay positions (R0-6), one of which (R0) already hosting a relay node. Five out of the six remaining positions are to be chosen using the iteration deployment algorithm. For comparison purposes, we run simulations for $n = 1, 2, 3, 4, 5$ (the maximum number of possible relays per iteration).
Figure 3(a) shows the full network with all nodes and links. The *thickness* of the links in the figure captures the quality of the link. The thicker a link, the higher the number of successful transmissions. Figure 3(b) shows the initial deployment with the Base Station (BS), the sensor nodes $S1-S6$, connected via the relay node $R0$ whereas Figure 3(c) shows the result of the iterative deployment algorithm with 3 additional relay nodes. As seen in the figure, the resulting network contains the thickest, i.e. strongest, possible links.

Figure 4 plots the cost of the tree computed by the iterative algorithm as a function of n, the maximum number of nodes to be potentially deployed at each step, for different values of the λ, the penalty added for exploration. We observe that for $\lambda = 0$ (dashed line) the cost is a decreasing function of n whereas for $\lambda = 2, 4$ (dashed-dotted and dotted resp.) the cost is increasing with n: the maximum number of possible relays nodes per iteration. To better understand this observation, we have plotted (Figures 5(a), 5(b), and 5(c)) the total number of relay nodes that are *actually* deployed after each iteration for each value of n. When $\lambda = 0$, for small values of n, the amount of new information that is gained at each step of the algorithm is limited: Since the network does not tend to explore, the actual number of nodes deployed at each iteration decreases as n decreases. On the other hand, when $\lambda = 2, 4$, for small values of n, the additional penalty forces the network to explore more, increasing the number of relay nodes deployed at each iteration. For large values of n however since all the non-observed links are penalized in the same way, we cannot observe the tendency to explore.

VI. CONCLUSION

We presented an iterative algorithm for node deployment in a WSN within an unknown environment. The proposed solution is particularly useful in scenarios where sensor nodes have already been deployed in fixed and known positions, and additional nodes such as relay nodes have to be deployed to guarantee connectivity while maintaining a low communication cost between the sensor nodes and a base station.

We proposed an iterative approach that incrementally deploys the network nodes while learning the characteristics of the wireless communication channel based on an underlying Gaussian Process (GP) model for the communication cost of the wireless links. This approach eliminates the unrealistic assumptions about communication characteristics such as a fixed circular "communication range" and the cost of a pilot deployment. We demonstrated that the algorithm is capable of deploying the additional relay nodes at very good positions based on the real world communication data.

Future studies will be focused on the extension of the experimental results for larger networks and on the application of the method for more general networks such as base station deployment for WiFi or WiMax networks.

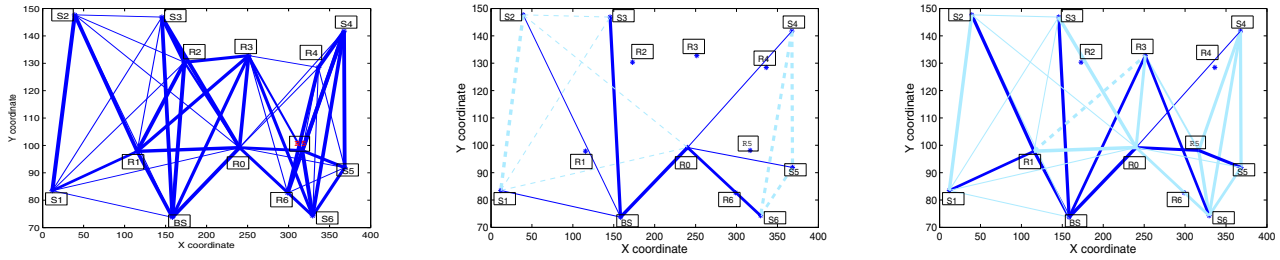
‡The authors would like to thank Andreas Krause for making the data available.

REFERENCES

[1] K. Chakrabarty, S. Iyengar, H. Qi, and E. Cho, "Grid Coverage of Surveillance and Target Location in Distributed Sensor Networks," *IEEE Transactions on Computers*, vol. 51, no. 12, pp. 1448–1453, 2002.

[2] K. Chakrabarty, H. Qi, S. Iyengar, and E. Cho, "Coding Theory Framework for Target Location in Distributed Sensor Networks," 2001. [Online]. Available: citeseer.ist.psu.edu/382274.html

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

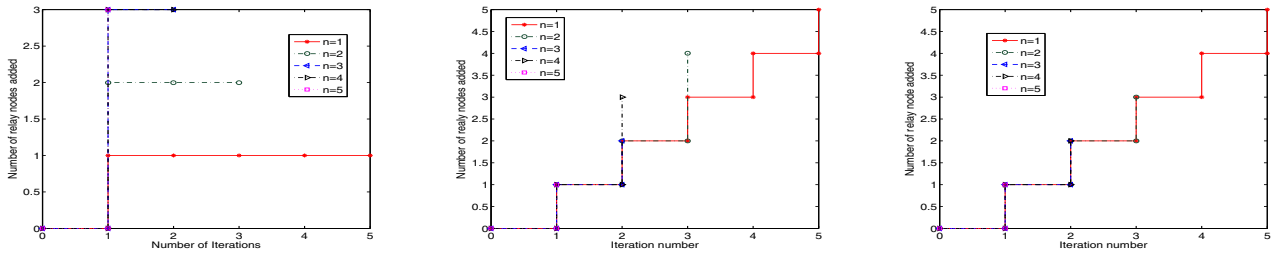


(a) Real-World Network: Complete Network 14 nodes, 50 links

(b) Real-World Network Initial Deployment: 1BS, 6SN, 1RN

(c) Output graph from iterative algorithm for the Real-World Network: 1BS, 6SN, 4RN

Fig. 3. Real-World Experiment results



(a) $\lambda = 0$

(b) $\lambda = 2$

(c) $\lambda = 4$

Fig. 5. Number of relay nodes added at each iteration for different values of n.

[3] S. Dhillon, K. Chakrabarty, and S. Iyengar, "Sensor Placement for Grid Coverage under Imprecise Detections," *In Proceedings of International Conference on Information Fusion*, vol. 12, 2002. [Online]. Available: citeseer.ist.psu.edu/dhillon02sensor.html

[4] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Near-optimal Sensor Placements: Maximizing Information while Minimizing Communication Cost," *ISPN'06*, 2006.

[5] C. K. I. Williams and D. Barber, "Bayesian Classification with Gaussian Processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1342–1351, 1998. [Online]. Available: citeseer.ist.psu.edu/williams98bayesian.html

[6] M. Kuss and C. E. Rasmussen, "Assessing Approximate Inference for Binary Gaussian Process Classification," *Machine Learning Reseach*, vol. 6, pp. 1679–1704, 2005.

[7] M. Seeger, "Gaussian Processes for Machine Learning," *International Journal of Neural Systems*, vol. 14, no. 2, pp. 1–38, 2004. [Online]. Available: citeseer.ist.psu.edu/seeger04gaussian.html

[8] C. Bishop and M. E. Tipping, "Bayesian Regression and Classification," *Advances in Learning Theory: Methods, Models and Applications*, vol. 190, pp. 267–285, 2003.

[9] J. H. Albert and S. Chib, "Bayesian Analysis of Binary and Polychotomous Response Data," *Journal of the American Statistical Association*, vol. 88, no. 422, 1993.

[10] J. Cramer, "The Origins of Logistic Regression," Jan 2003.

[11] S. Borman, "The Expectation Maximization Algorithm: A Short Tutorial." [Online]. Available: http://www.iro.umontreal.ca/ pift6266/A06/refs/EM_algorithm_tutorial.pdf

[12] J. Bilmes, "A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," *Technical Report, University of Berkeley, ICSI-TR-97-021, 1997.*, 1997. [Online]. Available: citeseer.ist.psu.edu/bilmes98gentle.html

[13] K. B. Petersen and M. S. Pedersen, "The matrix Cookbook," feb 2006, version 20051003. [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?3274

[14] A. Gueye, S. C. Ergen, and A. Sangiovanni-Vincentelli, "Iterative node deployment in an unknown environment," Department of Electrical Engineering and Computer Science, UC Berkeley, Berkeley, California, Tech. Rep., Sep 2008. [Online]. Available: http://www.eecs.berkeley.edu/˜ agueye/Iterative_Deployment

[15] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. Plenum Press, 1972, pp. 85–103.

[16] M. Karpinski and A. Zelikovsky, "New Approximation Algorithms for the Steiner Tree Problems," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 2, no. 30, 1995. [Online]. Available: citeseer.ist.psu.edu/karpinski95new.html

[17] D. E. Drake and S. Hougardy, "On Approximation Algorithms for the Terminal Steiner Tree Problem," *Information Processing Letters*, vol. 89, no. 1, 2004.

[18] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, , and M. Li, "Approximation Algorithms for Directed Steiner Problems," in *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, January 1998, pp. 192–200.

[19] H. J. Prömel and A. Steger, "A new approximation algorithm for the steiner tree problem with performance ratio 5/3," *J. Algorithms*, vol. 36, no. 1, pp. 89–101, 2000.

[20] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees," *Acta Informatica (Historical Archive)*, vol. 15, no. 2, pp. 141–145, June 1981.

[21] M. Opper, "A Bayesian Approach to Online Learning," pp. 363–378, 1998.

[22] T. Minka, "A Family of Algorithms for Approximate Bayesian Inference," *PhD Thesis, MIT Media Lab*, 2001. [Online]. Available: citeseer.ist.psu.edu/minka01family.html

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.