Simulated Annealing based Localization in Wireless Sensor Network

Anushiya A Kannan, Guoqiang Mao and Branka Vucetic
School of Electrical and Information Engineering,
University of Sydney, NSW 2006, Australia.
Email: {anushiya, guoqiang, branka}@ee.usyd.edu.au

***Abstract*— In this paper, we describe a novel localization method for ad hoc wireless sensor networks. Accurate self-organization and localization is an essential characteristic of high performance ad hoc wireless sensor networks. Many researchers have approached the localization problem from different angles. DV-hop method and convex optimization based semi-definite programming method are some of the well-known methods. In this paper, we propose a localization method based on the simulated annealing technique. Simulated annealing (SA) is used to estimate the approximate solution to combinatorial optimization problems. In this paper we show that our simulated annealing based localization (SAL) method can be used in ad hoc sensor networks to estimate the location of nodes accurately. The SAL can be viewed as an extension of the gradient search method. However, unlike the gradient search method which only allows downhill move, the SAL scheme can bring the convergence out of the local minima as it allows both up hill and down hill moves in a controlled fashion. We have tested our SAL scheme on a distributed sensor network of 200 nodes whose distance measurements are corrupted by Gaussian noise. Simulation results show that this novel scheme gives accurate and consistent location estimates of the nodes, which is better than the performance offered by other well-known schemes such as DV-hop method and convex optimization based semi-definite programming method.**

I. INTRODUCTION

Recent advances in integrated circuit design, embedded systems and novel sensing materials have enabled the development of low cost, low power, multi functional sensor nodes. These nodes with wireless interfaces and on-board processing capacity bring the idea of wireless sensor network into reality. It changes the way information is collected, specially in situations where information is hard to capture and observe.

Cheap, smart sensors, networked through wireless links and deployed in large numbers, with automatic localization capabilities provide unprecedented opportunities in many applications. Some of them include monitoring patients and assisting disabled patients in the health sector; monitoring and controlling homes and cities; monitoring bush fire, water quality surveillance etc. in the environment; monitoring humidity, temperature etc. in the agriculture. In addition, there are a broad spectrum of applications in the defense related area, where smart sensors offer new capabilities for reconnaissance and surveillance as well as other tactical applications.

In a sensor network, there will be a large number of sensor nodes densely deployed at positions which may not be predetermined. The information gathered and transferred by these micro-sensors will be meaningless unless they are tagged with the location where the information is obtained from. This makes self-organization and localization capabilities a fundamental requirement in sensor networks.

The rest of the paper is organized as follows: section II summarizes similar efforts in sensor network localization; section III presents a brief review of the simulated annealing method; section IV describes our proposed SAL approach and the related constrain setting and section V presents simulation results. We conclude this paper in section VI with intended future work as an extension to our proposed SAL method.

II. RELATED WORK

The primary function of a location estimation method is to estimate the coordinate of sensor nodes with respect to a set of anchor nodes with known global location information. Location estimation has been done in many different ways in the literature. Here we focus on a category of localization methods which estimate coordinate based on distance measurements.

A. Distance Measurement

Distance measurement may be done by using signal propagation time or received signal strength (RSS) information.

Different ways of using signal propagation time in distance measurement include time of arrival (ToA), round trip time of flight (RToF) or time difference of arrival (TDoA) of a signal [1]. In these propagation time based methods, expected high accuracies of distance measurements between nodes demands a stringent synchronization resolution. Furthermore, high propagation speed of wireless signals will magnify a small error in time measurement into a large error in distance measurements.

Since most of the mobile units in the market already have RSS indicator built in them, RSS method looks attractive for range measurements. But the accuracy of this method can be very much influenced by multi-path, fading, non-line of sight (NLoS) conditions and other sources of interference.

B. Coordinate Estimation

In the literature, different approaches are considered for coordinate estimation of the sensor nodes. The main aim of these methods is to estimate the coordinates of the sensor nodes with minimum error.

Niculescu [2] proposed a distributed, hop by hop localization method (APS). It uses a similar principle as that of GPS. Unlike in GPS, not all sensor nodes will have direct communication with the anchors in a sensor network. They can only communicate with their one-hop neighbors which are within their transmission range. So if a node wants to communicate with the anchor, it has to use the hop-by-hop propagation method. Each non-anchor node maintains an anchor information table and exchanges updates only with its neighbors. APS method starts with all anchors broadcasting hop counts which are set to zero. When the sensor nodes receive them, they will update the anchor information table and broadcast it again with hop count increased by one. Once all the anchors have received the distance of the other anchors in terms of number of hops, they estimates the average distance for one hop. This is broadcasted by the anchors as a correction factor to the entire network in hop-by-hop fashion. On receiving the correction factor, a non-anchor node may estimate distances to anchor nodes and perform triangulation to get its estimated coordinate.

Savarese [3] proposed a method whose function has two phases, HOP-TERRAIN and refinement. The HOP-TERRAIN phase is a similar method to that of Niculescu's, allowing all nodes to arrive at initial location estimates. The refinement phase of the method is an iterative method. This uses the results of the HOP-TERRAIN phase and the distance measurements of the immediate neighbors to do the least square triangulation. To mitigate error propagation, a confidence level, which is a value between $(0, 1)$, was associated with each node's estimated location. Nodes, like anchors, that have high faith in their location estimate select a high confidence value. A node that observes poor conditions (e.g., few neighbors) select a low confidence value. These are used to weigh the equations when solving the system of linear equations and the higher confidence value neighbor has more impact on the outcome of the triangulation performed by its neighbors.

Savvides [4] extended the single hop technique of GPS to multi-hop operation as in Niculescu's thus waiving the line of sight requirement with anchors. Non-anchor nodes collaborate and share information with one another over multiple hops to collectively estimate their locations. To prevent error accumulation in the network, they have used least squares estimation with a Kalman filter to estimate locations of all the non-anchor nodes simultaneously. In order to ensure that the solution is unique, they have introduced a method called computation trees. To avoid converging at local minima, they have used a geometrical relationship to obtain an initial estimate that is close to the final estimate. His algorithm is based on the assumptions that the distance measurement between the nodes and their neighbors are accurate.

Doherty [5] has approached the problem using convex optimization based on semi-definite programming. The connectivity of the network has been represented as a set of convex localizing constraints for the optimization problem. Pratik [6] extended this technique by taking the non-convex inequality constraints and relaxed the problem to a semi-definite program. Tzu-Chen Liang improved pratik's method further by a gradient-based local search method [7]. All these semi-definite programming methods requires rigorous centralized computation.

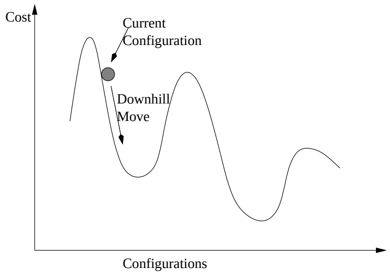

Fig. 1. Configuration space: balls and hills

In this paper, we propose to use simulated annealing technique to optimize a cost function in order to estimate the location of the sensor nodes. Initially all nodes, except the anchors, are assigned a random estimate of the location. A cost function, which represents the quantitative measure of the "goodness" of the coordinate estimate, was formulated using the estimated coordinates of the nodes and their measured distances with respect to their neighbors. Simulated annealing technique is used to optimize the cost function. This technique is very similar to iterative improvement method. But unlike the iterative improvement method, our proposed SAL method also allows uphill moves in a controlled fashion. Since simulated annealing accept bad moves in a controlled fashion, it is possible to jump out of the local minima and converge to a global minimum.

III. Simulated Annealing Technique

Simulated annealing theory explains how a defect free crystal can be formed from a liquid. At a low temperature the atoms of a liquid are in a highly ordered state, like that as a crystal lattice. But low temperature alone is not sufficient for finding the ground state of the substance. At high temperatures many atomic rearrangements happen. If it is cooled in an un-controlled manner, it may form a glass or a crystal with defects. Care must be taken to cool the liquid slowly and carefully to form a defect free crystal. The liquid must attain thermal equilibrium at each temperature while cooling. This is the efficient technique, called simulated annealing, modeled in early days to make the atoms to arrange themselves in a highly ordered state at each given temperature by Metropolis [8].

This principle of simulated annealing technique with an analogous set of "controlled cooling" operations was used in combinatorial optimization problems, such as minimizing functions of very many variables, to obtain a highly optimized, desirable solution by Kirkpatrick [9]. The "balls and hills" diagram [10] in Fig. 1 illustrates an optimization problem in one dimension. The cost surface is defined by including all the possible values of the cost function $f(x)$, taken over all legal configuration of x. In a normal gradient search method, the current configuration is perturbed only in the direction of reducing cost. Each new perturbation moves to a configuration downhill from the previous one. This may results in the solution trapped in a local minima. Simulated annealing allows perturbations to move uphill in a controlled fashion. Because each perturbation can transform one configuration into a worse configuration, it is possible to jump out of local minima and potentially fall into a more downhill path. However, because the uphill moves are carefully controlled; when we get closer to a good, final solution, we need not worry about getting out of it by an uphill move to some far worse one.

Even though simulated annealing could organize randomly placed variables into an optimized meaningful solution, it does not guarantee to get the optimum answer; but will give an acceptable answer in a reasonable time. Obtaining a good estimate for an optimization problem depends on the ability to simulate how the system reaches thermodynamic equilibrium at each fixed temperature in the schedule of decreasing temperatures.

IV. Simulated Annealing based Localization

The location estimation problem has a natural analogy with the simulated annealing algorithm. All the non-anchor nodes are initialized with random locations (x_i,y_i) within the boundaries. All the nodes have the ability to measure the distances between them and their one hop neighbors, which impose constraints on the possible values of the estimated coordinates. The measured distance is the true distance between the non-anchor node and its one hop neighbor, together with Gaussian noise describing the uncertainty of the distance measurement. A cost function (CF), which represents the quantitative measure of the "goodness" of the coordinate estimate is defined in Eq.1.

$$\begin{aligned} CF &= \sum_{i=1}^{N} \sum_{j \epsilon N_i} (\hat{d_{ij}} - d_{ij})^2 \\ \hat{d_{ij}} &= \sqrt{(\hat{x_i} - \hat{x_{ij}})^2 + (\hat{y_i} - \hat{y_{ij}})^2} \end{aligned} \tag{1}$$

Where:
N-total number of non-anchor nodes
N_i-set of neighbors of node i
$\hat{d_{ij}}$-estimated distance of perturbed node i with its neighbor j
d_{ij}-measured distance of perturbed node i with its neighbor j
($\hat{x_i}$,$\hat{y_i}$)-estimated coordinate of perturbed node i
($\hat{x_j}$,$\hat{y_j}$)-estimated coordinate of one hop neighbor j of perturbed node i

$$\Delta(CF) = CF_{new} - CF_{old} \tag{2}$$

Our aim is to optimize the cost function using simulated annealing technique to get the optimal solution without getting trapped in a local minimum. When the simulated annealing algorithm initially starts, the system is in a high energy state due to the random initial estimates of the coordinates of the non-anchor nodes.

In each step of the algorithm, a non-anchor node is chosen to be perturbed. The coordinate estimation ($\hat{x_i}$,$\hat{y_i}$) of the chosen node is given a small displacement in a random direction. The new value of the cost function is calculated for the new location estimation. If the change in cost function $\Delta(CF)$, is less than or equal to zero,(i.e. $\Delta(CF) \leq 0$) then the perturbation is accepted and the new configuration is used as the starting point of the next step.

The case $(\Delta(CF) > 0)$ is treated probabilistically: the probability that the displacement is accepted is $P(\Delta(CF)) = exp(-\Delta(CF)/T)$. Here T is a control parameter, which by analogy with the application of SA is known as the system "temperature" and P is a monotonically increasing function of T. A random probability (RP) uniformly distributed in the interval $(0, 1)$ is selected and compared with $P(\Delta(CF))$. If it is less than $P(\Delta(CF))$ then the perturbation is accepted and the new configuration is used as the starting point of the next step. If not, the perturbation is rejected and the original configuration is used as the starting point of the next step.

Initially, the "temperature" T is set to a high enough value to permit aggressive, essentially random search of the configuration space. At a higher "temperature" the probability of accepting a large uphill move is high. This could help the system jump out of local minimum. With the increase in the number of iterations, the system temperature decreases. When the temperature decreases, the probability of accepting a bad move decreases. The temperature is simply a control parameter. The idea is to employ a cooling method to moderate the acceptance of uphill moves over the course of the solution. Most of the uphill moves are allowed at higher temperatures. As the temperature cools, fewer uphill moves are allowed. In SAL the initial "temperature" was set such that the probability of accepting a bad uphill move is about 80%

A. Basic Components of Simulated Annealing based Localization (SAL)

The simulated annealing method needs four basic components [10]. They are set as follows in SAL:

- **Configurations:** This is a model of what a legal placement is. In SAL only planer situation is considered. (But augmenting to 3D is straight forward.) Nodes are placed randomly and uniformly in a square region of side length 10.
- **Move set:** This is a set of allowable moves that will reach all feasible configurations and one that is easy to compute. A simple move set is taken as a random direction in the planer multiplied by a small change in distance (Δd) in that direction. In order to bias the generation of random moves at lower temperatures, we empirically restrict the change in distance as the temperature cools by introducing a shrinking factor $\beta < 1$ where $(\Delta d)_{new} = \beta * (\Delta d)_{old}$.
- **Cost function:** This represents the quantitative measure of the "goodness" of the coordinate estimate of the localization configuration. It is defined in Eq.1.

```
$T$ = initial temperature
$(\Delta d)$ = initial move distance
WHILE (final temperature not met and change in cost function, $\Delta(CF)$, is not acceptably small)
{
    FOR $i = 1$ to $(q * N)$
    {
        pick a node to perturb
        DO $p$ times
        {
            generate a random perturbation to a node's estimated location
            evaluate the change in cost function, $\Delta(CF)$
            if $(\Delta(CF) \leq 0)$
                //downhill move $=>$ accept it
                accept this perturbation and update the configuration system
            else
                //uphill move $=>$ accept with probability
                pick a random probability $P(\Delta(CF)) = exp(-\Delta(CF)/T)$
                if$(\Delta(CF) \leq P(\Delta(CF)))$
                    accept this perturbation and update the configuration system
                else
                    reject this perturbation and keep the old configuration system
        }
    }
    $T_{new} = \alpha * T_{old}$
    $(\Delta d)_{new} = \beta * (\Delta d)_{old}$
}
```

Fig. 2. Simulated Annealing Algorithm

- **Cooling schedule:** It is used to anneal the problem from a random solution to a good, frozen solution. It is taken as the simplest one: $T_{new} = \alpha * T_{old}, \alpha < 1$, where the initial starting temperature and cooling rate are determined empirically to give a good result.

To get the optimum performance out of the simulated annealing technique, it is necessary to cool carefully and slowly, allowing it to come to thermal equilibrium at each temperature. At each temperature, $p * (q * N)$ perturbations is done in order to get the system into equilibrium in that particular temperature step. Here p is the number of perturbations given to a particular non-anchor node at a particular temperature and $(q * N)$ is the number of non-anchor nodes perturbed at a particular temperature were N is the number of nodes in the sensor network and q is a reasonably large number to make the system go into thermal equilibrium.

Two criteria can be used to stop the SAL simulation: when the change in the cost function ($\Delta(CF)$) is smaller than a predefined small number or when the predefined final temperature is reached. The structure of the simulated annealing algorithm is shown in Fig.2.

V. Simulation Results

In order to evaluate our SAL method, we ran many experiments on SAL using visual studio .NET. In this paper, we have chosen a sensor network with 200 nodes including the anchor nodes. They were uniformly distributed in a square region of $10*10$. The values of p and q in Fig.2 are chosen as 10 and 2 respectively, which defines the number of iterations done in our SAL method. The measured distance between the neighboring nodes, which is used in the cost function CF, is blurred by introducing a Gaussian noise into the true distance as shown in equation Eq.3.

Transmission Range	1.0	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0
Connectivity	5-6	7-8	10-11	13-14	17-18	20-21	24-27	28-29	33-34	37-39	40-42

TABLE I
The Relationship between the Transmission Range and The Connectivity

$$\begin{aligned} \hat{d}_{ij} &= d_{ij} * (1.0 + \text{Gaussian Noise}() * \text{Noise Factor}) \\ d_{ij} &= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \end{aligned} \quad (3)$$

Where:
(x_i, y_i) and (x_j, y_j) are original coordinates of two different sensor nodes.
$d_{ij}$is true distance and $\hat{d}_{ij}$ is measured distance between the two sensor nodes.

In our simulations noise factor is taken as 10%. The Gaussian noise has a mean of 0 and a standard deviation of 1. If the true distance between two nodes d_{ij} is smaller than the transmission range then the nodes are said to be neighbors. The connectivity (average number of neighbors per node) is controlled by specifying the transmission range. A relationship of transmission range vs. connectivity is tabulated in Table I. To allow for easy comparison between different scenarios, errors in location estimates are normalized to the transmission range (i.e 5% location error means absolute location error is $0.05 *$ transmission range). For every set of simulation, the number of nodes are fixed and the transmission range is varied in order to vary the connectivity. In Fig.3 to Fig.5, the original and estimated sensor locations are shown and an error offset line has been drawn between original and estimated locations. Fig.3 is for the case when there is no error in the distance measurements.

As shown in Fig.3, when no noise is introduced, that is in ideal situations, simulated annealing based localization (SAL) estimate the location of the nodes 100% accurately. Fig.4 shows a SAL simulation results with noise, for 10% anchors and a transmission range of 1.8. From this figure we can see that the SAL gives a very accurate result. But this is not the case always. When nodes are placed uniformly, all non-anchor nodes are not uniquely localizable all the time. These nodes which cannot be uniquely localized introduces a situation called flip ambiguity. If a node's neighbors are placed in positions such that they are approximately on the same line, and this node can be reflected across this line with no change in the cost function, then it is said to have flip ambiguity. In

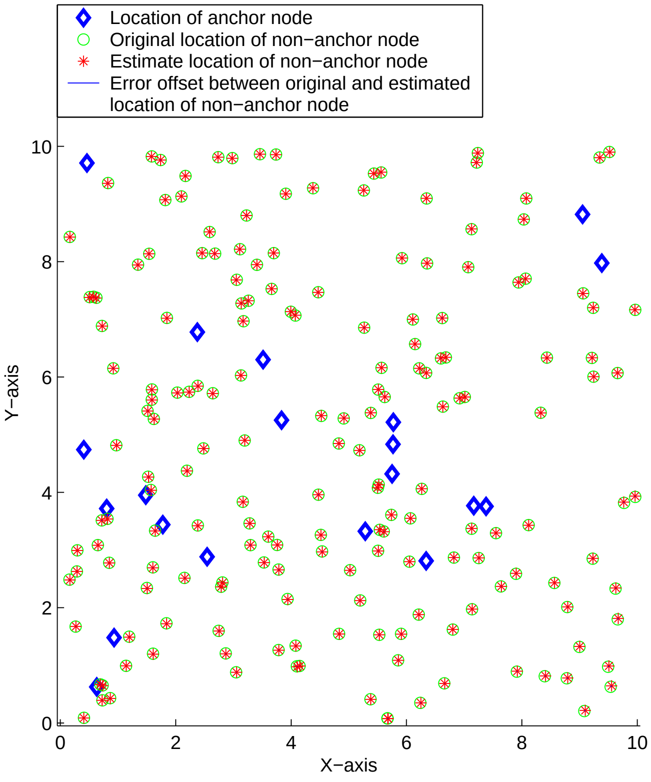

Fig. 3. SAL with 20 Anchors and no Noise

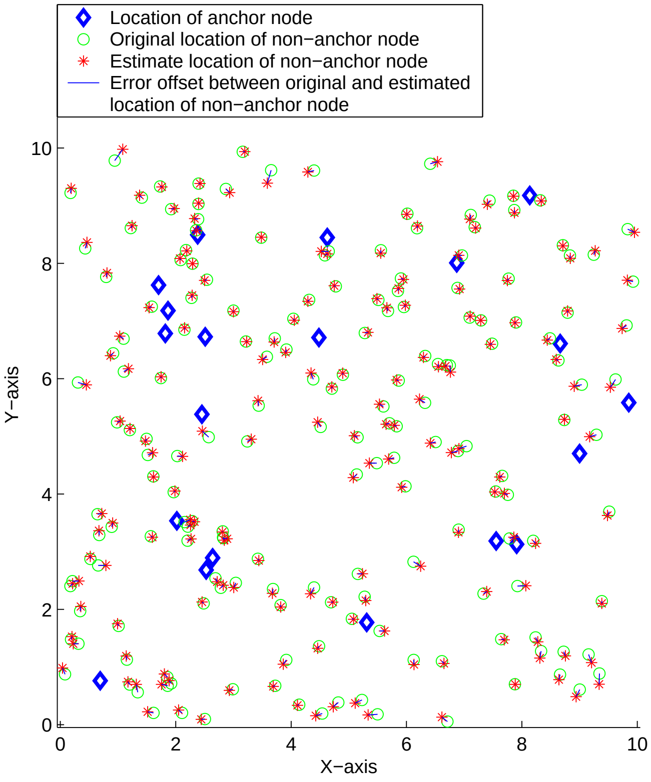

Fig. 4. SAL with 20 Anchors and 10%noise

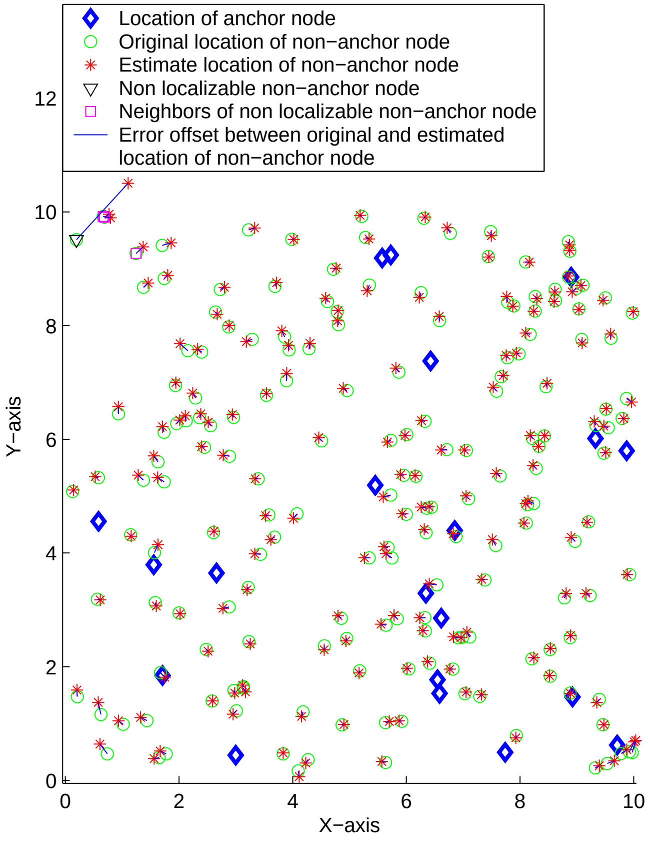

Fig. 5. SAL with 20 Anchors and 10% noise - flip ambiguity

this situation, this particular node is not uniquely localizable. Fig.5 highlights this flip ambiguity situation. In this figure non-uniquely localizable node is marked as a triangle and its one hop neighbors are marked as squares. If we look closely at the node that is not localized correctly, we could see that the neighbors of the node create a symmetrical line against which the node could be flipped while maintaining the same value for the cost function CF. This phenomena of flip ambiguity is discussed in [11]–[13]. But a good method to solve this problem is yet to be found. When this situation arises in SAL, most of those nodes which are not uniquely localizable have large errors, which makes the average error increases significantly. As the problem caused by flip ambiguity has gone beyond the capability of the class of localization algorithms based on distance measurements, we eliminate those nodes which can not be uniquely localized when computing the average localization error. From the simulations, it is also noted that flip-ambiguity is more common with less connectivity situations. When the connectivity increases, the flip-ambiguity occurs less frequently. The number of anchor nodes also play a part in the flip-ambiguity, but not to the extent of the connectivity.

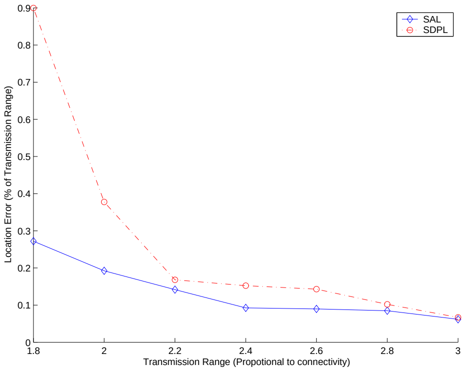

Fig. 6. Location error of uniformly distributed sensor nodes with 5% anchor

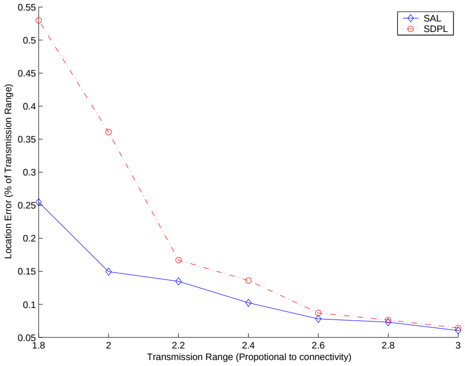

Fig. 7. Location error of uniformly distributed sensor nodes with 10% anchor

$$\text{location error} = \frac{1}{(n-m)} * \frac{\sum_{i=m+1}^{n} (x_i - \hat{x_i})^2 + (y_i - \hat{y_i})^2}{r^2} * 100\% \tag{4}$$

Where:
n - number of nodes in the sensor network
m - number of anchor nodes in the sensor network
$(\hat{x_i},\hat{y_i})$ - estimated location of non-anchor node
(x_i,y_i) - original location of non-anchor node
r - transmission range

From the literature review, it is observed that semi-definite programming relaxation with gradient-based local search localization (SDPL) [7] gives much better performance than the other methods reported in the literature. Thus we compare the performance of our proposed SAL algorithm with SDPL results. Fig.6 shows the SAL and SDPL simulation results of changing transmission range while having 5% anchor nodes and Fig.7 shows the simulation results of changing transmission range while having 10% anchor nodes. For each point in Fig.6 and Fig.7, ten simulations are performed with different random seeds and the average error is shown. The location error is calculated as in equation Eq.4. It is reported in percentage, relative to transmission range. In SAL when the connectivity is 17 or above, the mean square location error goes below 0.3%, no matter how many anchor nodes are present. When the connectivity is less, the number of anchor nodes plays an important role in flip ambiguity and about 20% of the simulations are affected by it. From Fig.6 and Fig.7, we could see that SAL performs better than SDPL.

VI. Conclusion and Future Work

We presented a novel Simulated Annealing based Localization method to determine the location of the non-anchor nodes in a uniformly distributed sensor network. Localization is based on iterative gradient search method, but allowing perturbations to move uphill in a controlled manner. We have shown via simulations that our proposed method gives better accuracy than the semi-definite programming localization. SAL has the following properties: it does not require special infrastructure or setup; it does not propagate error in localization to the other nodes; even though the simulation is done in a centralized system, it can be easily mapped to distributed system.

SAL system localizes the nodes accurately except when there are flip ambiguities. The performance of the localization system using distance measurement are affected by this flip ambiguity. However it is an objective of our future research to solve the flip ambiguity problem. When there are no flip ambiguities (i.e, all the non-anchor nodes can be uniquely identifiable), SAL system localizes all the nodes very well (location error of $\leq 0.3\%$ of transmission range).

Simulated annealing was done on centralized manner to analyze the theoretical idea. In the future, we would be investigating its implementation in a distributed manner, which will improve its scalability and reduce its complexity.

References

[1] M. Vossiek, L. Wiebking, P. Gulden, J. Wieghardt, C. Hoffmann, and P. Heide, “Wireless local positioning,” *IEEE Microwave Magazine*, vol. 4, no. 4, pp. 77–86, 2003.

[2] D. Niculescu and B. Nath, “Ad hoc positioning system (aps),” in *IEEE GLOBECOM ’01*, vol. 5, 2001, pp. 2926–2931.

[3] C. Savarese and J. Rabaey, “Robust positioning algorithms for distributed ad-hoc wireless sensor networks,” in *Proceedings of the General Track: 2002 USENIX Annual Technical Conference*, 2002, pp. 317–327.

[4] A. Savvides, H. Park, and M. B. Srivastava, “The bits and flops of the n-hop multilateration primitive for node localization problems,” in *International Workshopon Sensor Networks Application*, 2002, pp. 112–121.

[5] L. Doherty, K. pister, and L. El Ghaoui, “Convex position estimation in wireless sensor networks,” in *IEEE INFOCOM 2001*, vol. 3, 2001, pp. 1655–1663.

[6] P. Biswas and y. Ye, “Semidefinite programming for ad hoc wireless sensor network localization,” in *Information Processing in Sensor Networks, 2004. IPSN 2004*, 2004, pp. 46 – 54.

[7] T. C. Liang, T. C. Wang, and Y. Ye, “A gradient search method to round the semidefinite programming relaxation solution for ad hoc wireless sensor network localisation,” Sanford University, formal report 5, 2004. [Online]. Available: http://www.stanford.edu/ yyye/formal-report5.pdf

[8] M. A. E. . . N.Metropolis, A.Rosenbluth, *J.Chem. Phys.*, 1953.

[9] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[10] R. Rutenbar, “Simulated annealing algorithms: an overview,” *IEEE Circuits and Devices Magazine*, vol. 5, no. 1, pp. 19–26, 1989.

[11] D. Moore, J. Leonard, D. Rus, and S. Teller, “Robust distributed network localization with noisy range measurements,” in *Second Int. Conf. on Embedded Networked Sensor Systems (SenSys 2004)*, Nov. 2004.

[12] T. Eren, D. Goldenburg, W. Whiteley, Y. Yang, A. Morse, A. B.D.O, and B. P.N, “Rigidity, computation, and randomization in network localisation,” in *IEEE INFOCOM 2004*, vol. 4, 2004, pp. 2673 – 2684.

[13] D. Goldenburg, W. Krishnamurthy, A.and Maness, Y. Yang, and A. Young, “Network localization in partially localizable networks,” in *IEEE INFOCOM 2005*, 2005.