A Novel Hybrid Approach for Optimizing the Localization of Wireless Sensor Networks

Halima LAKHBAB^{1,*}

¹LIMSAD Laboratory, Faculty of Sciences Ain Chock - Casablanca, Hassan II University, Morocco

Abstract. Wireless sensor networks are used for monitoring the environment and controlling the physical environment. Information gathered by the sensors is only useful if the positions of the sensors are known. One of the solutions for this problem is Global Positioning System (GPS). However, this approach is prohibitively costly; both in terms of hardware and power requirements. Localization is defined as finding the physical coordinates of a group of nodes. Localization is classified as an unconstrained optimization problem. In this work, we propose a new algorithm to tackle the problem of localization; the algorithm is based on a hybridization of Particle Swarm Optimization (PSO) and Simulated Annealing (SA). Simulation results are given to illustrate the robustness and efficiency of the presented algorithm.

1 Introduction

Research on ad hoc wireless sensor networks receives a growing interest over the last few years. Sensor networks usually consist of a large number of sensors which are densely deployed in a geographical area. Such networks facilitate monitoring and controlling of physical environments from remote locations with better accuracy. Information gathered by the sensors is only useful if the positions of the sensors are known. One of the solutions for this problem is Global Positioning System (GPS). However, this approach is prohibitively costly, both in terms of hardware and power requirements. The Sensor Networks Localization Problem (SNLP) can be formally presented as follows: Assuming that the accurate positions of some nodes are known, how to use them and partial pair-wise distance measurements to estimate the positions of all sensor nodes in the sensor network? Particle swarm optimization algorithm is a population-based algorithm where several artificial particles search for good solutions. PSO was originally introduced by Kennedy and Eberhart in 1995 [1]. In this approach, each particle, of the population, called swarm, adjusts its trajectory toward its own previous best position, and toward the previous best position attained by any member of its topological neighborhood. Thus, the particles profit from the discoveries and previous experience of all other companions during the search for promising regions of the landscape. Simulated Annealing [2, 3] is a probabilistic technique for finding the global minimum of a cost function that may possess several local minima. SA it is commonly said to be the oldest among the metaheuristics and surely one of the first algorithms which had an explicit strategy to avoid local optima. SA exploits an analogy between the way in

which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more general system; the SA algorithm employs a random search which not only accepts changes that decrease the objective function, but also some changes that increase it. The probability of accepting a trial point is decreased during the search.

Several techniques have been proposed over the last few years in the literature to approximate the sensor network localization problem and its variants. Global optimization techniques [5], popular metaheuristic methods, such as simulated annealing SA [4], evolutionary algorithm [6] or particle swarm optimization [7–9]. Semidefinite Programming relaxations is also proposed [10, 11]. The semidefinite programming approach is further extended to a gradient search technique in [12]. In [13], we have proposed a combination of PSO method and a gradient based method, to approach the solution of the SNLP.

In the present work, the best characteristics of the PSO and of SA are combined to develop a novel hybrid algorithm, for tackling the localization problem. In the following section mathematical modeling of SNLP is presented. Our hybrid approach is outlined in section 3. In section 4, numerical results are presented. The paper is concluded in section 5.

2 Mathematical Modeling of the SNLP

The sensor network localization problem with some distance information can be studied in the framework of graph theory. Given a graph $G(V, E)$ and a set of non-negative weights $\{d_{ij} : (i, j) \in E\}$ on its edges, the goal is to place the vertices of G in $\mathbb{R}^p$, ($p = 2, 3$) such that the Euclidean distance between every pair of adjacent vertices (i, j) equals to the prescribed weight d_{ij} . The vertices of

*e-mail: halimalakhabab@yahoo.fr

G correspond to the sensors, the edges correspond to the communication links, and the weights correspond to the distances between sensors.

To solve this problem, we assume only a few nodes are equipped with GPS hardware. These nodes are called anchor nodes and they know their positions without communicating with other nodes.

A two-dimensional sensor network example, and its graph realization are presented in figures (1) and (2).

Figure 1. A two-dimensional sensor network example

Figure 2. A graph (realization) corresponding to the sensor network in "Figure 1"

In the SNLP, we have $m \geq 0$ anchor points $a_1, \dots, a_m \in \mathbb{R}^p$ whose locations are known, and $n > 0$ sensor points $x_1, \dots, x_n \in \mathbb{R}^p$ whose locations we wish to determine. Furthermore, we are given the Euclidean distance values $\bar{d}_{kj}$ between a_k and x_j for some $(k, j) \in E$ and d_{ij} between x_i and x_j for some $(i, j) \in E$ with $i < j$. Specifically, let $N_a = \{(k, j) \in E : \bar{d}_{kj} \text{ is specified}\}$ and $N_x = \{(i, j) \in E : i < j, d_{ij} \text{ is specified}\}$.

The goal of SNLP is to find the coordinates of $x_1, \dots, x_n \in \mathbb{R}^p$ such that:

$$\begin{aligned} \|a_k - x_j\| &= \bar{d}_{kj} \quad \forall (k, j) \in N_a \\ \|x_i - x_j\| &= d_{ij} \quad \forall (i, j) \in N_x \end{aligned} \quad (1)$$

The SNLP defined by (1) can be naturally formulated as a nonlinear global minimization problem, where the objective function is given by

$$f(x_1, \dots, x_n) = \sum_{(i,j) \in N_x} (\|x_i - x_j\| - d_{ij})^2 + \sum_{(j,k) \in N_a} (\|x_j - a_k\| - \bar{d}_{jk})^2 \quad (2)$$

Here, all constraints are substituted into the objective function.

This function is everywhere infinitely differentiable and has a large number of local minimizers. Also, it is easy to see that 0 is the global minimum of f .

3 The HyPSOSA Method

3.1 Particle Swarm Optimization (PSO) method

PSO is a powerful and promising optimization method which has a wide range of applications in many areas, especially in Wireless Sensor Networks Localization Problem [7–9]. Like the other evolutionary computation techniques, PSO is a population-based search algorithm inspired by social behavior of bird flocking. PSO is initialized with a population (or a swarm) of random solutions, called particles where each particle has its own velocity. Particles spread in the multi dimensional search space, representing a possible solution, with velocities which are dynamically adjusted according to their historical behaviors. Therefore, the particles have a tendency to fly toward a better region of search space over the course of search process. The basic PSO algorithm consists of three steps, namely, generating particle's positions and velocities, velocity update, and finally, position update. A particle changes its position from one move (iteration) to another based on velocity updates. First, the positions, x_k^i , and velocities, v_k^i , of the initial swarm of particles are randomly generated using upper and lower bounds on the design variables values, in the second step, velocities and positions of all particles are updated to persuade them to achieve better objective or fitness values which are functions of the particles current positions in the design space at time k . The fitness function value of a particle determines which particle has the best global value in the current swarm, p_k^g , and also determines the best position of each particle over time, p_k^i , i.e. in current and all previous moves. The velocity update formula is given by

$$v_{k+1}^i = \omega v_k^i + c_1 r_1 (p_k^i - x_k^i) + c_2 r_2 (p_k^g - x_k^i) \quad (3)$$

Here, and c_1 and c_2 are constants, and r_1 and r_2 are random numbers uniformly distributed in $[0, 1]$.

At the final step, PSO update the position by using (4).

$$x_{(k+1)}^i = x_k^i + v_k^i \quad (4)$$

3.2 Simulated Annealing method

The simulated annealing (SA) is a stochastic strategy for selecting the minimal configuration of states of a system. It was first put forward by Metropolis and successfully applied to the optimization problems by Kirkpatrick [2, 3]. In the basic SA algorithm, a random value is considered to be the initial solution in the search space. Then the new solution value of the objective function is calculated and compared with the current solution. A move towards the new solution is made from the current solution if it provides a better result than the previous iteration. The probability of accepting a new solution is given as follows:

$$P = e^{(-\Delta f)/T} \quad (5)$$

Where Δf is the energy difference of the two states and T is a control parameter called temperature, which controls the magnitude of the perturbations of the objectif function f .

To avoid getting trapped at a local minimum point, the rate of reduction of T should be slow. Thus, at the start of SA random and impractical moves may be accepted, but in the end only improving ones are likely to be allowed, which can help the procedure not to converge at a local minimum. A stop criterion is used to determine the points to stop the temperature cooling and finish the optimization.

3.3 The proposed hybrid Approach

In the present method, the best characteristics of the PSO, that has strong global-search ability, are combined with the good local search characteristics of the SA, to develop a novel hybrid algorithm; the proposed algorithm is called **HyPSOSA**. Other applications of hybrid PSO and SA algorithm can also be found in [14–18].

In every iteration of PSO, a random number is generated and is compared with r_k

$$r_k = 1 - e^{-\gamma k}$$

Where γ is a constant and k is the current number of iteration.

If the random number is less than r_k , a local search strategy by SA, around the worst current particle in the swarm, is detonated. We note x_k^{SA} , the solution given by SA.

If $f(x_k^{SA}) < f(p_k^g)$, then we take $p_k^g = x_k^{SA}$.

4 Numerical Results

Simulation of the SNLP and its performance evaluation is done in Scilab 5.5.2.

PSO parameters are set as follows:

- Population size = 50
- Maximum number of iterations = 50
- Acceleration constants $c_1 = c_2 = 1.7$

The linearly decreasing method is adopted for the inertia factor, which is

$$w_k = w_{max} - \frac{(w_{max} - w_{min}) \times k}{Nmax}$$

where $Nmax$ is the maximum number of iterations, w_{max} and w_{min} are maximum and minimum weights, respectively, and k is the current iteration number.

We implement the SA algorithm with the following parameters: the initial temperature is set to 50 and the maximum number of iteration is 20. We set the constant γ to 0.05.

Figure (4) gives the node deployments in the square area, the anchor node rate is 10% from the 100 nodes. In this figure, we also illustrate the communications links between nodes¹.

The figure (5) shows the localizations given by HyPSOSA method for the previous example. The diamonds and the circles indicate the true positions of the anchors and the sensors, respectively.

Next, we consider three problems: S14A6, S27A3 and S90A10 that consist of 14 sensors and 6 anchors, 27 sensors and 3 anchors, 90 sensors and 10 anchors, respectively. In the table (1) we report the smallest value in the objectif function f_{best} reached by the algorithm for 10 runs and the average of minimums f_{mean} .

¹ $R = 0.3$: node communication radius

Figure 3. Flow chart of the HyPSOSA algorithm

Figure 4. Neighbor relationship diagram

Figure 5. Example S90A10. True location and HyPSOSA localization

Table 1. Results of PSO and HyPSOSA for 3 examples

	f_{best}		f_{mean}	
	PSO	HyPSOSA	PSO	HyPSOSA
S14A6	0.0000017	$2,894.10^{-8}$	0.0003665	0.0000076
S27A3	$3,427.10^{-10}$	$4,387.10^{-11}$	0.0000017	$5,748.10^{-8}$
S90A10	$1,613.10^{-9}$	$4,035.10^{-10}$	0.0000287	0.0000011

Note that the average number of function evaluations by PSO is 2550 for the tree example, and by HyPSOSA is 3182, 2761 and 3196 for S14A6, S27A3 and S90A10 respectively.

We conclude that the minimums and the average of minimum given by HyPSOSA, for the tree examples, are better than those given by PSO.

5 Conclusion

In the literature many methods and systems are proposed and described to estimate the location of nodes in Sensor Network Localization Problem. In this paper we presented a new hybrid approach for tackling this problem by combining a Particles Swarm Optimization and Simulated Annealing. Experiment results show the efficiency of the proposed method.

Note that HyPSOSA can potentially be used in other minimization problem.

References

- [1] J. Kennedy, R. Eberhart, Particle Swarm Optimization, Proc. IEEE Int. Conf. Neural Networks, 1942-1948 (1995)
- [2] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equations of state calculations by fast computing machines, J. of Chemical Physics **21**(6), 1087-1091 (1953)
- [3] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Sc. **220**, 671-680 (1983)
- [4] A. A. Kannan, G. Mao, B. Vucetic, Simulated annealing based localization in wireless sensor network, Proc. IEEE Conf. LCN'05, 513-514 (2005)
- [5] J. Moré, Z. Wu, Global continuation for distance geometry problems, SIAM J. Opt. **7**, 814-836 (1997)
- [6] G.F. Nan, M.Q. Li, J. Li, Estimation of node localization with a real-coded genetic algorithm in WSNs, Proc. Int. Conf. on Machine Learning and Cybernetics **2**, 873-878 (2007)
- [7] R. C. Abreu, J. E. C. Arroyo, A Particle Swarm Optimization Algorithm for topology Control in Wireless Sensor Networks, Inter. Conf. Chilean (2011)
- [8] J.J. Gnana Chandran, S. P. Victor, An Energy Efficient Localization Technique Using Particle Swarm Optimization in Mobile Wireless Sensor Networks, American J. of Sc. Res., 33-48 (2010)
- [9] R. V. Kulkarni, G. K. Venayagamoorthy, A. Miller, C. H. Dagli, Network-centric Localization in MANETs based on Particle Swarm Optimization IEEE Swarm Intelligence Symp., 1-6(2008)
- [10] J. Aspnes, W. Whiteley, Y. R. Yang, IEEE Trans. mobile computing, A theory of network localization, **5**(12), 1663-1678 (2006)
- [11] P. Biswas, Y. Ye, Proc. 3rd Int. Symp. on Information Processing in Sensor Networks, 46-54 (2004)
- [12] T.-C. Liang, T.-C. Wang, Y. Ye, A gradient search method to round the semidefinite programming relaxation solution for ad hoc wireless sensor network localization, Technical report, Stanford University (2004)
- [13] H. Lakhbab, S. El Bernoussi, Int. J. of Math. Analysis **6**, (2012)
- [14] F.Javidrad, M.Nazari, A new hybrid particle swarm and simulated annealing stochastic optimization method, Appl. Soft. Comp. **60**, 634-654 (2017)
- [15] M. Basu, P. Deb, G. Garai, Hybrid of Particle Swarm Optimization and Simulated Annealing for Multidimensional Function Optimization, Inter. J. Information Technology **20**(1),34-45 (2014)
- [16] M. Bahrepour, E. Mahdipour, R. Cheloi, M. Yaghoobi, Super-sapso: a new sa-based pso algorithm, Appl. of Soft Comp. **58**, 423-430 (2009)
- [17] G. Yang, D. Chen, G. Zhou, A new hybrid algorithm of particle swarm optimization, Lecture Notes in Comp. Sc. **4115**, 50-60 (2006)
- [18] N. Sadati, M. Zamani, H. Mahdavian, Hybrid particle swarm-based simulated annealing optimization techniques Proc. IEEE Industrial Electronics Conf. 644-648 (2006)