Maximum Lilkelihood Source Localization in Wireless Sensor Network Using Particle Swarm Optimization

T Panigrahi*, G Panda†, B Mulgrew‡, and Babita Majhi§
* Department of Electronics and Communication Engineering,
National Institute of Technology, Rourkela, India-769008
Email: panigrahit@nitrkl.ac.in
†School of Electrical Science, Indian Institute of Technology, Bhubaneswar, India-713002
‡Institute for Digital Communication, The Unversity of Edinburgh, UK
§The Department of Computer Application, The SOA Unversity, Bhubaneswar, India

Abstract—**Wireless sensor networks have been proposed as a solution to environment sensing, target tracking, data collection and other applications. Source localization is one of the important problem in wireless sensor network. In literature a decentralized approach using strong antena arrays at each node or sensor arrays at different positions are used to localize the sources. In this paper a purely co-operative method where every node will participate in estimation. The network does the bearing estimation by optimizing maximum likelihood function by forming random array among all the nodes. Particle swarm optimization is used to optimize ML function because it is more efficient compared to other evolutionary algorithm like GA. Finally the results are compared with most analyzed MUSIC algorithm.**

Index Terms—**Wireless sensor network, Maximum Likelihood estimation, MUSIC, particle swarm optimization.**

I. Introduction

Sensor signal processing in wireless sensor network deals with the problem of extracting information from a collection of measurements obtained from sensors distributed in space. The number of signals present is assumed to be finite, and each signal is parameterized by a finite number of parameters. Based on measurements of the sensor output, the objective is to estimate some desired parameters. This research area has attracted considerable interest for several years. A vast number of algorithms has appeared in the literature for estimating unknown signal parameters from the measured output of a sensor array [1]. In wireless sensor network where the sensor nodes are distributed arbitrarily in an geographical area has an important signal-processing task is source localization.

The functionality of the localization systems depends on the requirements and constraints at hand. Most localization methods depend on three types of physical variables measured by or derived from sensor readings for localization: time delay of arrival (TDOA), direction of arrival (DOA) and received sensor signal strength(RSS) or power. Widely used localization methods are time of arrival (TOA) and time difference of arrival(TDOA). Both these methodes provids accurate source localization in WSN [2]. The main disadvantages here is that both TOA and DOA need network synchronization. The DOA estimate can be obtained using array signal processing techniques. The DOA estimation is based on time difference between the sinsor or phase differnce in case of narrow band signals.

In literature we can find people were trying to estimate source location by measuring DOA with an antenna strong array at each sensor node [3], [4] or by taking group of sensor subarrays [5]. The processor of fusion is used to the DOA estimate from each subarray of sensors. Then the triangulation process of determining the intersection of these cross bearing DOA angles can be used to estimate the source location.

In this paper we doveloped efficient sensor network to estimate source DOA by forming a random array. The MLE technique is used here because of its superior statistical performance compared to other existing method. A likelihood function can be formulated easily if we know the observed parametric data [6]. The ML estimate is computed by maximizing the likelihood function or minimizing the negative likelihood function with respect to all unknown parameters, which may include the source DOA angles, the signal covariance, and the noise parameters. Since the ML function is multimodal, so direct optimization is seems to be unrealistic due to large computational burden. So main contribution is to reduce the dimensionality by taking some assumptions on signal, noise and array structure.

There are different optimization techniques available in literature for optimization of ML function like AP-AML [7], fast EM and SAGE algorithms [8] and a local search technique e.g. Quasi-Newton methods. All these techniques have several limitations because of multidimensional cost function which need extensive computation, good initialization is also crucial for global optimization and we can not guarantee that these local search techniques always have global converge.

Basically the ML function is a multimodal function. The gradient basd algorithms like simplex algorithm are available in literature which need good initialization to avoid to fall the solution in local minima. The evolutionary algorithms like genetic algorithm [9], particle swarm optimization and simulated annealing [10], [11] can be designed to optimize the ML function.

The feasibility of PSO to ML criterion for the accurate estimation of signal parameters is studied here. PSO is a recent evolutionary algorithm first introduced by Eberhart and Kennedy in 1995 [12], [13]. As an emerging technology, PSO has attracted a lot of attention in recent years, and has been successfully applied in many fields, such as phased array synthesis [14], [15], electromagnetic optimization [16], and etc. Most of the applications demonstrated that PSO could give competitive or even better results in a faster and cheaper way, compared with other heuristic methods such as GA.

Here each node in the network form an arbitrary array at fussion center to optimize ML function. After collecting spatially uncorrelated data it share to fussion center once with their relative postions to have global ML cost function. The global ML function is optimized by using PSO. We can choose node 1 is the central node or fussion center for the bearing estimation.

II. Data Model and Maximum Likelihood Estimation Problem

A major application of sensor array technology is estimations of parameters of the impinging signal to the array. Parameters to be identified include number of signals, magnitudes, frequencies, direction of arrival (DOA), distances and speeds of signals. Of all these parameters, the DOA estimation is has been paid most attention, especially in far-field signal applications, in which case the wave front of the signal may be treated planar, indicating that the distance is irrelevant. thus, the topic of current research is also focus on DOA estimation using far-field source consideration.

The parametric DOA algorithms are based on minimizing quadratic penalty functions. The penalty function came from some signal-noise model equation, which holds when the tried angle of arrival is exactly the actual incident angle. Because the exact incident angle is unknown, the model equation is supposed to be violated with a guessed angle. An optimal estimation of the incident angle may be obtained by minimizing a penalty function, which is usually a quadratic function on of the residue of the signal-noise model equation. This is known as maximum likelihood solution. Different ML algorithms have different likelihood functions, which came from different models of the signals to be estimated.

There are two categories of ML which are deterministic ML and Stochastic ML depending on the model of assumption on the signal waveform. Deterministic ML algorithms assumes that the signal waveform is deterministic but unknown, while the stochastic ML algorithms assumes that the signal waveform is Gaussian random processes. Both classes of ML algorithms assume zero mean Gaussian random noise.

Let us consider an array of *M* WSN nodes are distributed in an arbitrary geometry and received signals form N narrow band far-field signal sources at unknown locations at $[\theta_1, \ldots, \theta_N]^T$. The known nominal location of the m*th* sensor is given by $[x_m, y_m]$. The output of sensor nodes modeled by standard equation fo L snapshots as

$$\mathbf{x}(i) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(i) + \mathbf{n}(i), \qquad i = 1, 2, \ldots, L \tag{1}$$

where $\mathbf{s}(i)$ is the unknown vector of signal waveforms, $\mathbf{n}(i)$ is unpredicted noise process, , L denotes the number of data samples (snapshots). The matrix $\mathbf{A}(\boldsymbol{\theta})$ has the following special structure defined as

$$\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1), \ldots \mathbf{a}(\theta_N)] \tag{2}$$

where $\mathbf{a}(\theta)$ is called steering vector and $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_N]^T$ are the parameters of interest or true DOA's. The exact form of $\mathbf{a}(\theta)$ depends on the position of the nodes in sensor network, where then, mth element of the array manifold $\mathbf{A}(\boldsymbol{\theta})$ is

$$\exp\left\{ j\frac{2\pi}{\lambda}[x_m \sin\theta_n + y_m \cos\theta_m] \right\}$$

in which λ is the wavelength of the signal.

Further, the vectors of signals and noise are assumed to be stationary, temporarily white, zero-mean complex Gaussian random variables with the following second-order moments given by

$$\begin{aligned} E[\mathbf{s}(i)\mathbf{s}(j)^{\mathrm{H}}] &= \mathbf{S}\delta_{ij} \\ E[\mathbf{s}(i)\mathbf{s}(j)^{\mathrm{T}}] &= 0 \\ E[\mathbf{n}(i)\mathbf{n}(j)^{\mathrm{H}}] &= \sigma^2\mathbf{I}\delta_{ij} \\ E[\mathbf{n}(i)\mathbf{n}(j)^{\mathrm{T}}] &= 0 \end{aligned} \tag{3}$$

where δ_{ij} is the Kronecker delta, $(\cdot)^{\mathrm{H}}$ denotes complex conjugate transpose, $(\cdot)^{\mathrm{T}}$ denotes transpose, $E(\cdot)$ stands for expectation.

A. Maximum Likelihood Estimation

In many applications it is appropriate to model the signals as stationary stochastic processes, possessing a certain probability distribution. Then by far most commonly advocated distribution is the Gaussian one. Not only is this for the mathematical convenience of the resulting approach, but the Gaussian assumption is also often motivated by the Central Limit Theorem.

Under the assumptions taken above, the observation process, $\mathbf{x}(i)$, constitutes a stationary, zero-mean Gaussian random process having second-order moments

$$E[\mathbf{x}(i)\mathbf{x}(i)^{\mathrm{H}}] = \mathbf{R} = \mathbf{A}(\boldsymbol{\theta})\mathbf{S}\mathbf{A}^{\mathrm{H}}(\boldsymbol{\theta}) + \sigma^2\mathbf{I} \tag{4}$$

In most applications, no *a-priori* information on the signal covariance matrix and the number of sources are available. But here in this work we assume that the number of signals are known to us. The problem addressed herein is the estimation of $\boldsymbol{\theta}$ along with the parameter in $\mathbf{S}$ and σ^2(noise power) from a batch of L measured data $\mathbf{x}(1), \ldots, \mathbf{x}(L)$.

Under the assumption of additive Gaussian noise and complex Gaussian distributed signals we can have negative loglikelihood function [1], [17] is given as

$$\ell(\boldsymbol{\theta}, \mathbf{S}, \sigma^2) = \log|\mathbf{R}| + \mathrm{tr}\{\mathbf{R}^{-1}\hat{\mathbf{R}}\} \tag{5}$$

where $\hat{\mathbf{R}}$ is the sample covariance matrix and it defined as

$$\hat{\mathbf{R}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}(i)\mathbf{x}(i)^{\mathrm{H}} \tag{6}$$

The ML criterion function can be concentrated with respect to $\mathbf{S}$ and σ^2 by following [18]–[20]. The stochastic maximum likelihood (SML) estimates of the signal covariance matrix and the noise power are obtained by inserting the SML estimates of $\boldsymbol{\theta}$ in the following expressions

$$\hat{\mathbf{S}}(\boldsymbol{\theta}) = \mathbf{A}^{\dagger}(\boldsymbol{\theta}) \left(\hat{\mathbf{R}} - \hat{\sigma}^2 \mathbf{I} \right) \mathbf{A}^{\dagger \mathrm{H}}(\boldsymbol{\theta}) \tag{7a}$$

$$\hat{\sigma}^2(\boldsymbol{\theta}) = \frac{1}{M-N} \mathrm{Tr}\{\mathbf{P}_{\mathbf{A}}^{\perp}(\boldsymbol{\theta})\hat{\mathbf{R}}\} \tag{7b}$$

where $\mathbf{A}^{\dagger}$ is the psedo-inverse of $\mathbf{A}$ and $\mathbf{P}_{\mathbf{A}}^{\perp}$ is the orthogonal projection onto the null space of $\mathbf{A}^{\mathrm{H}}$ and are defined as

$$\mathbf{A}^{\dagger} = \left(\mathbf{A}^{\mathrm{H}}\mathbf{A}\right)^{-1} \mathbf{A}^{\mathrm{H}} \tag{8a}$$

$$\mathbf{P}_{\mathbf{A}} = \mathbf{A}\mathbf{A}^{\dagger} \tag{8b}$$

$$\mathbf{P}_{\mathbf{A}}^{\perp} = \mathbf{I} - \mathbf{P}_{\mathbf{A}} \tag{8c}$$

Therefore the concentrated form of the UML function now can be obtained bu using (7) in (5) as

$$f_{UML}(\boldsymbol{\theta}) = \log|\mathbf{A}(\boldsymbol{\theta})\hat{\mathbf{S}}(\boldsymbol{\theta})\mathbf{A}^{\mathrm{H}}(\boldsymbol{\theta}) + \hat{\sigma}^2(\boldsymbol{\theta})\mathbf{I}| \tag{9}$$

Stoica and Nehorai [21] proved that for uncorrelated sources, the statistical performances of CML and UML are similar; while for highly correlated or coherent sources, the performance of UML is significantly superior. Here we focus on the problem how to optimize the UML function by using PSO in cooperative way.

III. Particle Swarm Optimization PSO

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995 [12], [13], inspired by social behavior of bird flocking or fish schooling. In past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods. Another reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. Particle swarm optimization has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement.

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). A problem is given, and some way to evaluate a proposed solution to it exists in the form of a fitness function. A communication structure or social network is also defined, assigning neighbors for each individual to interact with. Then a population of individuals defined as random guesses at the problem solutions is initialized. These individuals are candidate solutions. They are also known as the particles, hence the name particle swarm. An iterative process to improve these particle solutions is set in motion. The particles iteratively evaluate the fitness of the particle solutions and remember the location where they had their best success. The individuals best solution is called the particle best or the local best known as *pbest*. Each particle makes this information available to their neighbors. Another best value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. when a particle takes all the population as its topological neighbors, the best value is a global best and is called *gbest*.

The swarm is typically modeled by particles in multidimensional space that have a position and a velocity. Consider a *D*-dimensional problem space and a swarm consisting of *P* particles. The position of the *i*th particle is a *D*-dimensional vector $x_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]$. The velocity of this particle is represented as $v_i = [v_{i1}, v_{i2}, \ldots, v_{iD}]$. These particles fly through hyperspace (i.e., $\mathbb{R}^D$) and have two essential reasoning capabilities: their memory of their own best position and knowledge of the global or their neighborhood's best. The best previous position of the *i*th particle, which gives the best fitness value, is denoted as $p_i = [p_{i1}, p_{i2}, \ldots, p_{iD}]$ and the best position found by any particle in the swarm is represented by $p_g = [p_{g1}, p_{g2}, \ldots, p_{gD}]$. In a minimization optimization problem, problems are formulated so that best simply means the position with the smallest objective value. Members of a swarm communicate good positions to each other and adjust their own position and velocity based on these good positions. So a particle has the following information to make a suitable change in its position and velocity:

At every iteration, the velocity and the position of each particle are updated according to the following equations:

$$\mathbf{v}_j^{n+1} = w^n \mathbf{v}_j^n + c_1 \mathbf{r}_1^n \odot (\mathbf{p}_j^n - \mathbf{x}_j^n) + c_2 \mathbf{r}_2^n \odot (\mathbf{p}_g^n - \mathbf{x}_j^n) \tag{10}$$

$$\mathbf{x}_j^{n+1} = \mathbf{x}_j^n + \mathbf{v}_j^{n+1} \tag{11}$$

where $\odot$ denotes element-wise product, $j = 1, 2, \ldots, P$, and $n = 1, 2, \ldots$, indicates the iterations, *w* is a parameter called the inertia weight, c_1 and c_2 are positive constants referred to as *cognitive* and *social* parameters respectively, $\mathbf{r}_1$ and $\mathbf{r}_2$ are D-dimensional vectors consisting of independent random numbers uniformly distributed between 0 and 1, which are used to stochastically vary the relative pull of $\mathbf{p}_i$ and $\mathbf{p}_g$ in order to simulate the unpredictable component of natural swarm behaviour.

IV. PSO for DOA Estimation in Wireless Sensor Network

Here we describe the formulation of the PSO algorithm for UML optimization to estimate source DOA's by wireless sensor network. At first the network starts by initializing a population of particles in the search space with random positions constrained between 0 and π in each dimension, and random velocities in between 0 and 1π. The *N*-dimensional position vector of the *j*th particle takes the form $x_j = [\theta_1, \ldots, \theta_N]$, where θ represents the DOAs of the sources. A particle position vector is converted to a candidate solution vector in the problem space through a suitable mapping. The score of the mapped vector evaluated by a likelihood function f_{UML} which is given in (9), is regarded as the fitness of the corresponding particle.

Every node in the network collect group of spatial and temporal complex uncorrelated data. To evaluate the likelihood function f_{UML} the network need the data from all the nodes. Therefore after collecting the data, each nodes sends to central node once per each experiments. During the evolution of algorithm, in every iteration update each particles velocity and position, then evaluate the global best.

The manipulation of a particle's velocity according to (10) is regarded as the central element of the entire optimization. Three components typically contribute to the new velocity. The first part refers to the inertial effect of the movement, which is just proportional to the old velocity and is the tendency of the particle to proceed in the same direction it has been travailing. The inertial weight w is considered critical for the convergence behaviour of PSO [22]. A larger w facilitates searching new area and global exploration while a smaller w tends to facilitate local exploitation in the current search area. In this study, w is selected to decrease during the optimization process, thus PSO tends to have more global search ability at the beginning of the run while having more local search ability near the end of the optimization.

Given a maximum value w_{max} and a minimum value w_{min}, w is updated as follows:

$$w^n = \begin{cases} w_{\text{max}} - \frac{w_{\text{max}} - w_{\text{min}}}{nK}(n-1), & \text{if } 1 \leq n \leq [nK] \\ w_{\text{min}}, & \text{for } [nK]+1 \leq n \leq K \end{cases} \quad (12)$$

where $[nK]$ is the number of iterations with time decreasing inertial weight, $0 < n < 1$ is a ratio, $\mathbf{K}$ is the maximum iteration number, and $[\cdot]$ is a rounding operator. Based on empirical practice [23] and extensive test runs, we select $w_{\text{max}} = 0.9, w_{\text{min}} = 0.4$, and $r = 0.4 \sim 0.8$.

The second and third components of the velocity update equation introduce stochastic tendencies to return toward the particles own best historical position and the group's best historical position. These paradigms allow particles to profit both from their own discoveries as well as the discoveries of the swarm as a whole, mixing local and global information uniquely for each particle on each iteration. Constants c_1 and c_2 are used to bias the particle's search towards the two best locations. These two parameters are not critical for the convergence of PSO. Following common practice in the literature [13], $c_1 = c_2 = 2$, although these values could be fine-tuned for the problem at hand. Since there was no actual mechanism for controlling the velocity of a particle, it is necessary to define a maximum velocity to avoid the danger of swarm explosion and divergence [24]. The velocity limit can be applied along each dimension at every node as

$$v_{jn}^k = \begin{cases} V_{\text{MAX}}, & \text{if } v_{jn}^k > V_{\text{MAX}} \\ V_{\text{MIN}}, & \text{if } v_{jn}^k < V_{\text{MIN}} \end{cases} \quad (13)$$

where $n = 1, \ldots, N$. In this work, we keep the limitation of V_{MAX} is set to the half value of the dynamic range, i.e., $V_{\text{MAX}} = 0.5$. The new particle position is calculated using (10). If any dimension of the new position vector is less than zero or more than one, it is clipped or adjusted to stay within this range.

The optimization iteration will be terminated if the specified maximum iteration number **K** is reached. The final global best position p_{kg} is taken as the ML estimates of source DOA.

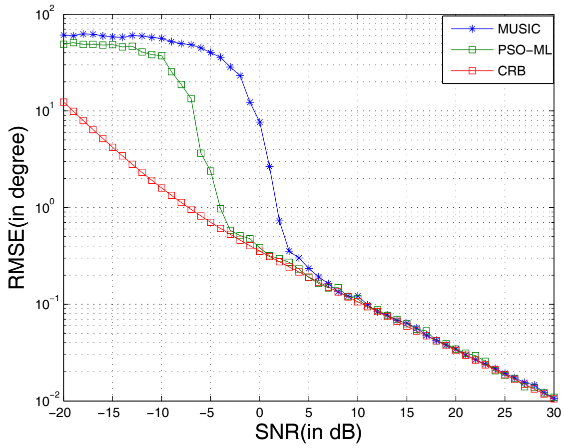

Fig. 1. DOA estimation RMSE values by global PSO and MUSIC

V. Simulation Results and Discussions

Here we present a numerical example to demonstrate the performance of PSO based DOA estimation using (9) against MUSIC [25] which is the best known and well investigated algorithm. The performances of those methods are compared in two ways: (a) the DOA estimation root-mean-squared error (RMSE), which is calculated as [26]

$$\text{RMSE} = \sqrt{\frac{1}{NN_{\text{run}}} \sum_{l=1}^{N_{\text{run}}} \sum_{n=1}^{N} \left(\hat{\theta}_n(l) - \theta_n\right)^2} \quad (14)$$

where N is the number of sources, $\hat{\theta}_n(l)$ is the estimate of the nth DOA achieved in the lth run, θ_n is the true DOA of the nth source; and (b) the ability to resolve closely spaced sources known as probability of resolution (PR). By definition, two sources are said to be resolved in a given run if both $|\hat{\theta}_1 - \theta_1|$ and $|\hat{\theta}_2 - \theta_2|$ are smaller than $|\theta_1 - \theta_2|/2$.

A. Example

In this example we consider $N = 8$ nodes with better connectivity to share ther observed information to the central node. We are assumed that each node knows its position co-ordinates for the determination of array manifold matrix $\mathbf{A}(\boldsymbol{\theta})$. Then It act as an arbitrary array and form its own steering matrix accordingly their position of the nodes. We assume that two equal power uncorrelated signal sources are impinging on the sensor network with true DOA's $[150° \; 158°]$ with respect to x-axis.

First the DOA is estimated by optimizing ML globally by PSO. We consider node#1 is a central node to compute f_{ML} and all the node shared their received data to it. After that we used MUSIC algorithm for the same estimation. Fig. 1 describes the DOA estimation RMSE obtained using global PSO-UML and MUSIC as a function of SNR. Fig. 2 shows the resolution probability for the same methods.

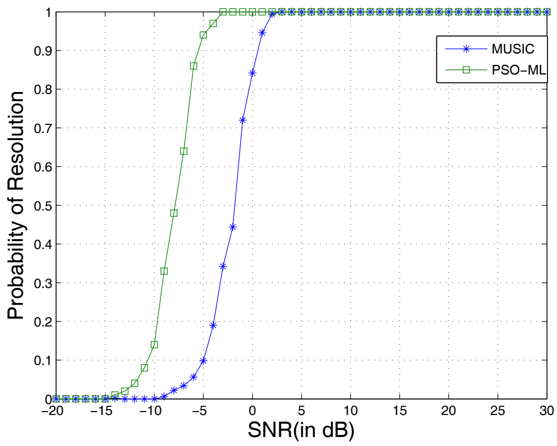

Fig. 2. DOA estimation PR values by global PSO and MUSIC

We can see from Fig. 1 and Fig. 2 the PSO-UML produce much superior performance compared to global MUSIC algorithm. The MUSIC algorithm gives better DOA estimation at high SNR, but at very low SNR the PSO-UML produce very good performance. The SNR simulated range is from -20dB to 30dB with 1dB step size. At lower SNR we can also ovbserve that around 5dB gain getting over gloval MUSIC algorithm. We have taken 100 Motecarlo trial are performed for each SNR. The array shape which is arbitrary here is remained shame that means the sensors are not changing their position through out the estimation process. Here very small number of (L = 20) snapshots taken for PSO-ML estimation, but to achive theoretical bound MUSIC algorithms need 2000 snapshots. So that MUSIC needs more communication among the nodes to have all the information at central processor for the estimation.

VI. Conclusion

Here we proposed a new method to estimate the direction of arrival of sources in wireless sensor network. In the network every node participate to estimate source bearing by optimizing the ML function after forming the whole network as an arbitrary array. This methods gives best performance even at low SNR compared to other algortithm. The PSO-ML estimates required very less number of snapshots than MUSIC, so that the data communication is reduced here which is the another advatage for energy efficient WSN.

References

[1] B. Ottersten, M. Viberg, P. Stoica, and A. Nehorai, "Exact and large sample ML techniques for parameter estimation and detection in array processing," *Radar Array Processing,Simon Haykin (ed.), Springer-Verlag.*, pp. 99–152, Feb 1993.

[2] L. Mailaender, "On the geolocation bounds for round-trip time-of-arrival and all non-line-of-sight channels," *EURASIP J. Adv. Signal Process*, vol. 2008, January 2008. [Online]. Available: http://dx.doi.org/10.1155/2008/584670

[3] R. L. Moses, O. L. Moses, D. Krishnamurthy, and R. Patterson, "A self-localization method for wireless sensor networks," *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 348–358, 2002.

[4] J. Chen, R. Hudson, and K. Yao, "Maximum-likelihood source localization and unknown sensor location estimation for wideband signals in the near-field," *Signal Processing, IEEE Transactions on*, vol. 50, no. 8, pp. 1843–1854, Aug 2002.

[5] J. C. Chen, K. Yao, and R. E. Hudson, "Source localization and beamforming," *IEEE signal Processing Magazine*, pp. 30–39, March 2002.

[6] P. Stoica and K. Sharman, "Maximum likelihood methods for direction-of-arrival estimation," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 38, no. 7, pp. 1132–1143, Jul 1990.

[7] I. Ziskind and M. Wax, "Maximum likelihood localization of multiple sources by alternating projection,," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 36, no. 10, pp. 1553–1560, Oct 1988.

[8] P. J. Chung and J. F. Böhme, "Doa estimation using fast em and sage algorithms," *Signal Process.*, vol. 82, no. 11, pp. 1753–1762, 2002.

[9] J. Rodriguez, F. Ares, E. Moreno, and G. Franceschetti, "Genetic algorithm procedure for linear array failure correction," *Electronics Letters*, vol. 36, no. 3, pp. 196–198, Feb 2000.

[10] P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated annealing: theory and applications (Mathematics and Its Applications)*. Springer publisher, 1987.

[11] P. J. M. Laarhoven and E. H. L. Aarts, Eds., *Simulated annealing: theory and applications*. Norwell, MA, USA: Kluwer Academic Publishers, 1987.

[12] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *in Proc. IEEE Conf. Neural Networks,*, vol. IV, IEEE service center, Piscataway, NJ, 1995, pp. 1942–1948.

[13] ——, "Particle swarm optimization: developments, applications and resources," in *in Proc. congress on evolutionary computation 2001,*, IEEE service center, Piscataway, NJ,Seoul, Korea., 2001.

[14] T. Panigrahi, A. Patnaik, S. Sinha, and C. Christodoulou, "Amplitude only compensation for failed antenna array using particle swarm optimization," in *Antennas and Propagation Society International Symposium, 2008. AP-S 2008. IEEE*, July 2008, pp. 1–4.

[15] D. Boeringer and D. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," *Antennas and Propagation, IEEE Transactions on*, vol. 52, no. 3, pp. 771–779, March 2004.

[16] J. Robinson and Y. Rahmat-Samii, "Particle swarm optimization in electromagnetics," *Antennas and Propagation, IEEE Transactions on*, vol. 52, no. 2, pp. 397–407, Feb. 2004.

[17] M. Li and Y. Lu, "Maximum likelihood DOA estimation in unknown colored noise fields," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 44, no. 3, pp. 1079–1090, July 2008.

[18] B. Friedlander and A. Weiss, "On direction finding with unknown noise covariance," vol. 2, pp. 780–784, Oct-2 Nov 1994.

[19] ——, "Direction finding using noise covariance modeling," *Signal Processing, IEEE Transactions on*, vol. 43, no. 7, pp. 1557–1567, Jul 1995.

[20] A. G. Jaffer, "Maximum likelihood direction finding of stochastic sources: A separable solution," in *in Proc. ICASSP 88,*, vol. 5, New York, April 1988, pp. 2893–2896.

[21] P. Stoica and A. Nehorai, "Performance study of conditional and unconditional direction-of-arrival estimation," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 38, no. 10, pp. 1783–1795, Oct 1990.

[22] K. E. Parsopoulos and M. N. Vrahatis, "Recent approaches to global optimization problems through particle swarm optimization," *Natural Computing*, vol. 1, pp. 235–306, 2002.

[23] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *EP '98: Proceedings of the 7th International Conference on Evolutionary Programming VII*. London, UK: Springer-Verlag, 1998, pp. 591–600.

[24] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 1, pp. 58–73, Feb 2002.

[25] R. Schmidt, "Multiple emitter location and signal parameter estimation," *Antennas and Propagation, IEEE Transactions on*, vol. 34, no. 3, pp. 276–280, MARCH 1986.

[26] M. Li and Y. Lu, "A refined genetic algorithm for accurate and reliable DOA estimation with a sensor array," *Wirel. Pers. Commun.*, vol. 43, no. 2, pp. 533–547, 2007.