

Research Article

Multiple Target Localization with Bistatic Radar Using Heuristic Computational Intelligence Techniques

Fawad Zaman,¹ Ijaz Mansoor Qureshi,² Ata Ur Rehman,¹ and Shujaat Ali Khan Tanoli¹

¹Department of Electrical Engineering, COMSATS Institute of Information Technology, Attock Campus, Attock 43600, Pakistan

²Department of Electrical Engineering, Air University, Islamabad Campus, Islamabad 44000, Pakistan

Correspondence should be addressed to Fawad Zaman; fawad@ciit-attock.edu.pk

Received 19 April 2015; Accepted 30 July 2015

Academic Editor: Ananda S. Mohan

Copyright © 2015 Fawad Zaman et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We assume Bistatic Phase Multiple Input Multiple Output radar having passive Centrosymmetric Cross Shape Sensor Array (CSCA) on its receiver. Let the transmitter of this Bistatic radar send coherent signals using a subarray that gives a fairly wide beam with a large solid angle so as to cover up any potential relevant target in the near field. We developed Heuristic Computational Intelligence (HCI) based techniques to jointly estimate the range, amplitude, and elevation and azimuth angles of these multiple targets impinging on the CSCA. In this connection, first the global search optimizers, that is, are developed separately Particle Swarm Optimization (PSO) and Differential Evolution (DE) are developed separately, and, to enhance the performances further, both of them are hybridized with a local search optimizer called Active Set Algorithm (ASA). Initially, the performance of PSO, DE, PSO hybridized with ASA, and DE hybridized with ASA are compared with each other and then with some traditional techniques available in literature using root mean square error (RMSE) as figure of merit.

1. Introduction

It is easy to deal with the far field targets because all the waves emitted by them are assumed to be plane waves and, by incorporating such assumption, one can characterize the signal model only by the Direction of Arrival (DOA) of the targets [1, 2]. On the other hand, it is comparatively difficult and challenging to localize the targets located close to the sensors array, that is, in near field. In this case, the plane wave assumption is no longer applicable and one has to deal with spherical waves. Thus, to localize the near field targets, the range information must also be incorporated into the signal model along with DOA [3, 4]. Near field targets localization, especially 3D (range, elevation angle, and azimuth angle), plays significant role in radar, cognitive radio networks, and array signal processing, since it is a preliminary step for adaptive beamformer to guide the main beam in preferred direction and simultaneously manage the nulls in the direction of jammers [5–7].

Several algorithms are available to address the issue of 3D near field source localization problem. In [8–10] different schemes are discussed but they not only are computationally expensive but also have the problem of pair matching between elevation and azimuth angles. To solve the pair matching problem, a two-stage separated steering vector-based algorithm is proposed in [11] but it has higher MSE and is computationally expensive as it requires more than 400 snapshots to achieve the results. It also fails to estimate the amplitude of sources which is also sometimes an important parameter to be estimated. Clearly the goal is to develop a scheme which must be able to jointly estimate the amplitude, range, elevation angle, and azimuth angle, should provide an improved MSE, and finally should be free of pair matching problem.

In order to achieve the desired goals Heuristic Computational Intelligence (HCI) techniques are available. In the last two decades, these techniques were being widely applied to a mixture of problems ranging from handy applications

in industry and commerce to leading scientific research [12–14]. These techniques include Differential Evolution (DE), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA). Besides the other useful properties, one of the most significant aspects of these techniques is their property of hybridization with any other heuristic or nonheuristic techniques. Due to this property, their reliability and applicability increase even more. In [15], an efficient scheme based on GA hybridized with Pattern Search (PS) is developed for the joint estimation of amplitude and 2D DOA of far field sources impinging on L shape arrays (1-L and 2-L shape arrays). Similarly, another useful algorithm is designed in [16] which is based on GA hybridized with Interior Point Algorithm (IPA) to jointly estimate the amplitude, range, and elevation angle of near field sources impinging on Uniform Linear Array (ULA).

In this paper, we assume a scenario of Bistatic Phase Multiple Input Multiple Output (MIMO) radar with colocated antennas which is capable of sending multiple beams with diverse wave forms on diverse targets. But, before getting into the mode of tracking targets by sending high gain sharp beams on the targets, it is required to estimate their spherical coordinates, that is, range and elevation and azimuth angles (r, θ, ϕ). To achieve this, the transmitter sends a coherent signal beam using any subarray of the transmitter such that a beam has a fairly wide solid angle so as to cover up any potential target in the space. The receiver in the first mode of this scenario is a CSCA of passive sensors consisting of two X and Y subarrays. The reflected signals from different targets will impinge on these X - Y arrays. Since we are interested in the ranges of the targets apart from 2D DOA, we treat the targets as near field in order to localize them completely. To jointly estimate the amplitude, range, and elevation and azimuth angles of the targets, we employ

- (a) PSO,
- (b) DE,
- (c) PSO hybridized with Active Set Algorithm (ASA),
- (d) DE hybridized with ASA.

In both hybrid schemes, the best individual results achieved through DE and PSO are given as starting point to ASA for further tuning. The performance criterion is made on the basis of mean square error (MSE) which is used as fitness function derived from maximum likelihood principle (MLP) [16]. This fitness function is able to automatically pair the amplitude, range, and elevation and azimuth angles of a specific target. The other properties of this fitness function are its ease in concept, ease in implementation, robustness against noise, and requirement of single snapshot. The performances of PSO, DE, Particle Swarm Optimization hybridized with Active Set Algorithm (PSO-ASA), and Differential Evolution hybridized with Active Set Algorithm (DE-ASA) are not only compared with each other but also with some traditional techniques available in literature [10, 11] using RMSE as figure of merit.

The rest of the paper is organized as follows: Section 2 evaluates the problem formulation, while the proposed HCI techniques are discussed in Section 3. In the same way, results

and discussion are carried out in Section 4, while Section 5 is dedicated to conclusion and future work directions.

2. Data Model for Near Field Targets

In this section, we developed a data model for K near field targets impinging on CSCA placed on the receiver of Bistatic radar. The CSCA is composed of two symmetric sub-ULAs placed along x -axis and y -axis, respectively, as shown in Figure 1. Each ULA carries $2 \times P$ passive sensors, while the reference sensor is common for them. For $K < 4P + 1$, the data model at m th and n th sensor in the x -axis and y -axis subarray, respectively, can be represented as

$$w_{m,0} = \sum_{k=1}^K s_k e^{j\psi_{xk}(m)} + \eta_{m,0}, \quad (1)$$

$$w_{0,n} = \sum_{k=1}^K s_k e^{j\psi_{yk}(n)} + \eta_{0,n}. \quad (2)$$

In (1) and (2), $\psi_{xk}(m)$ is the propagation delay for k th target between $(0, 0)$ th and $(m, 0)$ th sensors, while $\psi_{yk}(n)$ is the propagation delay for k th target between $(0, 0)$ th and $(0, n)$ th sensors. For k th near field target, the signal phases $\psi_{xk}(m)$ and $\psi_{yk}(n)$ are parameterized mathematically as intermediate parameters in the form given as

$$\psi_{xk}(m) = m\alpha_{xk} + m^2\beta_{xk}, \quad (3)$$

$$\psi_{yk}(n) = n\alpha_{yk} + n^2\beta_{yk}.$$

In (3), α_{xk} , β_{xk} , α_{yk} , and β_{yk} are the function of k th source amplitude (s), elevation angle (θ), azimuth angle (ϕ), and range (r). Mathematically, they can be expressed as

$$\begin{aligned} \alpha_{xk} &= \frac{-2\pi d \sin \theta_k \cos \phi_k}{\lambda}, \\ \beta_{xk} &= \frac{\pi d^2 (1 - \sin^2 \theta_k \cos^2 \phi_k)}{\lambda r_k}, \\ \alpha_{yk} &= \frac{-2\pi d \sin \theta_k \sin \phi_k}{\lambda}, \\ \beta_{yk} &= \frac{\pi d^2 (1 - \sin^2 \theta_k \sin^2 \phi_k)}{\lambda r_k}. \end{aligned} \quad (4)$$

In (1) and (2), $\eta_{m,0}$ and $\eta_{0,n}$ are Additive White Gaussian Noise (AWGN) added at m th and n th sensors, respectively. In vector form, the signal model can be represented as

$$\mathbf{w} = \mathbf{A}\mathbf{s} + \boldsymbol{\eta}, \quad (5)$$

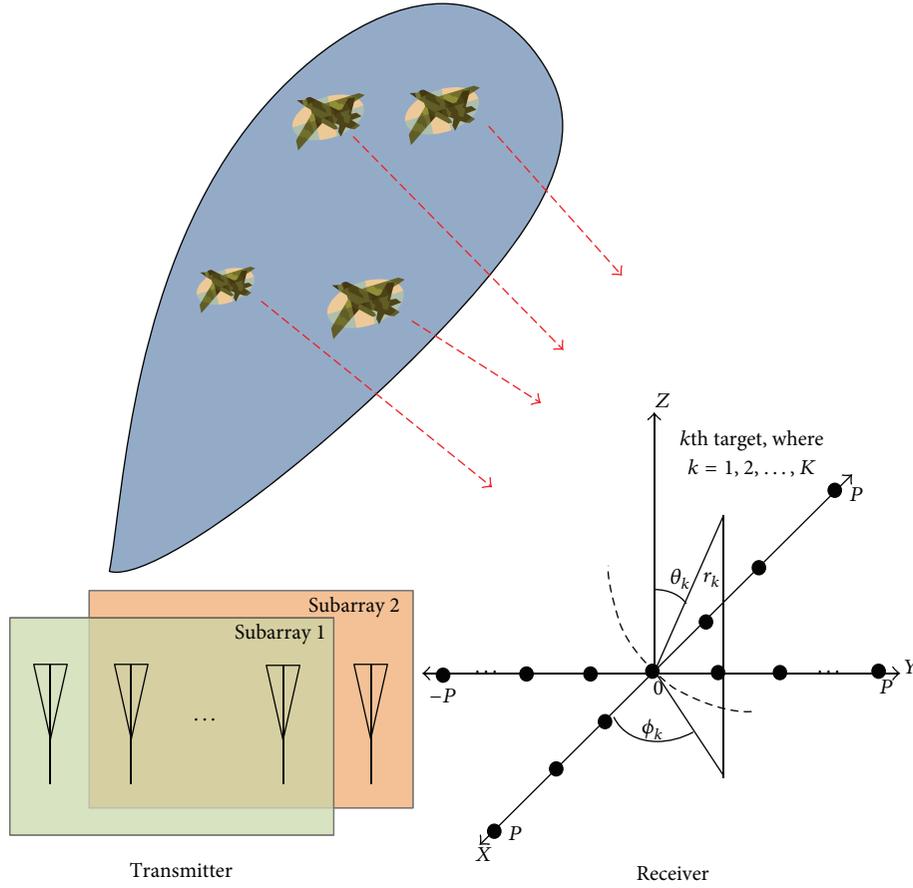


FIGURE 1: Schematic diagram for Bistatic radar.

where

$$\mathbf{s} = [s_1, s_2, s_3, \dots, s_K]^T,$$

$$\mathbf{w} = [w_{-P,0}, w_{-P+1,0}, \dots, w_{-1,0}, w_{1,0}, w_{2,0}, \dots, w_{P-1,0}, w_{P,0}, w_{0,-P}, w_{0,-P+1}, \dots, w_{0,-1}, w_{0,1}, w_{0,2}, \dots, w_{0,P-1}, w_{0,P}]^T,$$

$$\boldsymbol{\eta} = [\eta_{-P,0}, \eta_{-P+1,0}, \dots, \eta_{-1,0}, \eta_{1,0}, \eta_{2,0}, \dots, \eta_{P-1,0}, \eta_{P,0}, \eta_{0,-P}, \eta_{0,-P+1}, \dots, \eta_{0,-1}, \eta_{0,1}, \eta_{0,2}, \dots, \eta_{0,P-1}, \eta_{0,P}]^T, \quad (6)$$

$$\mathbf{A} = [a(\alpha_{x1}, \beta_{x1}, \alpha_{y1}, \beta_{y1}), a(\alpha_{x2}, \beta_{x2}, \alpha_{y2}, \beta_{y2}), \dots, a(\alpha_{xK}, \beta_{xK}, \alpha_{yK}, \beta_{yK})],$$

where

$$\mathbf{a}(\alpha_{xk}, \beta_{xk}, \alpha_{yk}, \beta_{yk}) = [e^{j((-P)\alpha_{xk} + (-P)^2\beta_{xk})}, e^{j((-P+1)\alpha_{xk} + (-P+1)^2\beta_{xk})}, \dots, e^{j(-\alpha_{xk} + \beta_{xk})}, e^{j(\alpha_{xk} + \beta_{xk})}, \dots, e^{j((P-1)\alpha_{xk} + (P-1)^2\beta_{xk})}, e^{j((P)\alpha_{xk} + (P)^2\beta_{xk})}, 1, e^{j((-P)\alpha_{yk} + (-P)^2\beta_{yk})}, e^{j((-P+1)\alpha_{yk} + (-P+1)^2\beta_{yk})}, \dots, e^{j(-\alpha_{yk} + \beta_{yk})}, e^{j(\alpha_{yk} + \beta_{yk})}, \dots, e^{j((P-1)\alpha_{yk} + (P-1)^2\beta_{yk})}, e^{j((P)\alpha_{yk} + (P)^2\beta_{yk})}]^T. \quad (7)$$

Now clearly the problem in hand is to accurately and jointly estimate the unknown parameters (amplitudes, ranges, and elevation and azimuth angles) of the reflected signals from targets.

3. Heuristic Computational Intelligence

Heuristic Computational Intelligence (HCI) is a subfield of artificial intelligence that can be used for continuous

and combinatorial optimization problems. The entire HCI algorithms are global optimization based methods which have stochastic or meta-heuristic characteristics which can be used for black box or gradient free problems. Basically, these kinds of algorithms make use of the iterative progress in a population and, to achieve the desired goal, the population is selected using parallel processing in random guided manner [17]. The major advantages of HCI over traditional optimization methods include wide range of applicability, conceptual simplicity, knowledge usage capability, hybridization with any other scheme, robustness against dynamic changes, and ability of self-optimization [18]. In literature, several HCI based algorithms are available but we are focusing on DE and PSO.

3.1. Particle Swarm Optimization (PSO). PSO algorithm is inspired from bird flocking and school fishing which was proposed by Kennedy and Eberhart in 1995 [19]. It can handle multimodal, discontinuous, and nonconvex optimization problems because it is a pseudooptimization method [20, 21]. The schematic diagram of PSO is shown in Figure 2, while its execution steps are given below.

Step 1 (initialization). Initialize the swarm randomly, that is, randomly producing Q particles. In this problem, each particle consists of $4 * K$ genes, where K represents the total number of available targets. Mathematically, it can be written as

$$\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \dots, \mathbf{b}_Q]^T,$$

$$\begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \vdots \\ \mathbf{b}_Q \end{bmatrix} = \begin{bmatrix} \theta_{1,1}, \theta_{1,2}, \dots, \theta_{1,K}, \phi_{1,K+1}, \phi_{1,K+2}, \dots, \phi_{1,2*K}, r_{1,(2*K+1)}, r_{1,(2*K+2)}, \dots, r_{1,(3*K)}, s_{1,(3*K+1)}, s_{1,(3*K+2)}, \dots, s_{1,(4*K)} \\ \theta_{2,1}, \theta_{2,2}, \dots, \theta_{2,K}, \phi_{2,K+1}, \phi_{2,K+2}, \dots, \phi_{2,2*K}, r_{2,(2*K+1)}, r_{2,(2*K+2)}, \dots, r_{2,(3*K)}, s_{2,(3*K+1)}, s_{2,(3*K+2)}, \dots, s_{2,(4*K)} \\ \theta_{3,1}, \theta_{3,2}, \dots, \theta_{3,K}, \phi_{3,K+1}, \phi_{3,K+2}, \dots, \phi_{3,2*K}, r_{3,(2*K+1)}, r_{3,(2*K+2)}, \dots, r_{3,(3*K)}, s_{3,(3*K+1)}, s_{3,(3*K+2)}, \dots, s_{3,(4*K)} \\ \vdots \\ \theta_{Q,1}, \theta_{Q,2}, \dots, \theta_{Q,K}, \phi_{Q,K+1}, \phi_{Q,K+2}, \dots, \phi_{Q,2*K}, r_{Q,(2*K+1)}, r_{Q,(2*K+2)}, \dots, r_{Q,(3*K)}, s_{Q,(3*K+1)}, s_{Q,(3*K+2)}, \dots, s_{Q,(4*K)} \end{bmatrix}. \quad (8)$$

The upper and lower bounds of θ , ϕ , r , and s are given as

$$\begin{aligned} \theta_{q,k} &\in R : 0 \leq \theta_{q,k} \leq \frac{\pi}{2} \\ \phi_{q,K+k} &\in R : 0 \leq \phi_{q,K+k} \leq 2\pi \\ r_{q,2K+k} &\in R : r_l \leq r_{q,2K+k} \leq r_u \\ s_{q,3K+k} &\in R : s_l \leq s_{q,3K+k} \leq s_u \end{aligned} \quad (9)$$

for, $q = 1, 2, \dots, Q$, $k = 1, 2, \dots, K$,

where r_l and r_u represent the lower and upper limits of ranges, while s_u and s_l are the upper and lower limits of amplitudes.

Step 2 (fitness function). We have used MSE as a fitness function to compute the fitness of every particle. For q th particle, it can be mathematically written as

$$FF(q) = \frac{1}{(1 + \xi(q))}, \quad (10)$$

where $\xi(q)$ is called MSE which is derived from maximum likelihood principle [16]. The total MSE for q th particle can be mathematically written as

$$\xi(q) = \xi_1(q) + \xi_2(q), \quad (11)$$

$$\xi_1(q) = \left(\frac{1}{2P_x} \right) \sum_{m=-P_x}^{P_x} |w_{m,0} - \widehat{w}_{m,0}^q|^2, \quad (12)$$

$$\xi_2(q) = \left(\frac{1}{2P_y} \right) \sum_{n=-P_y}^{P_y} |w_{0,n} - \widehat{w}_{0,n}^q|^2, \quad (13)$$

where P_x and P_y are the number of sensors placed along x -axis and y -axis, respectively. In (12) and (13), $w_{m,0}$ and $w_{0,n}$

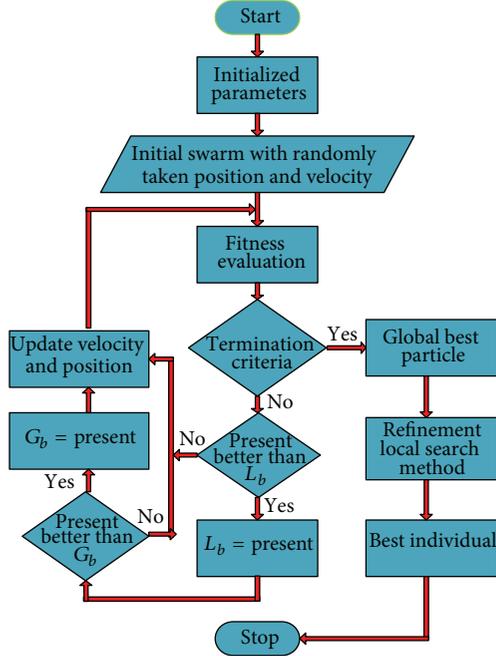


FIGURE 2: Generic flow diagram of Hybrid Particle Swarm Optimization.

are given in (1) and (2), respectively, while, $\widehat{w}_{m,0}^q$ and $\widehat{w}_{0,n}^q$ can be defined as

$$\begin{aligned} \widehat{w}_{m,0}^q &= \sum_{k=1}^K \widehat{b}_{3K+k}^q e^{j((-m2\pi d \sin(\widehat{b}_k^q) \cos(\widehat{b}_{3K+k}^q))/\lambda + (n^2 \pi d^2 (1 - \sin^2(\widehat{b}_k^q)) \cos^2(\widehat{b}_{3K+k}^q))/\lambda \widehat{b}_{2K+k}^q)}, \\ \widehat{w}_{0,n}^q &= \sum_{k=1}^K \widehat{b}_{3K+k}^q e^{j(-n2\pi d \sin(\widehat{b}_k^q) \sin(\widehat{b}_{3K+k}^q))/\lambda + (n^2 \pi d^2 (1 - \sin^2(\widehat{b}_k^q)) \sin^2(\widehat{b}_{3K+k}^q))/\lambda \widehat{b}_{2K+k}^q}. \end{aligned} \quad (14)$$

Now, store each particle as local best (l) while the one having maximum fitness function is stored as global best (g_b) particle.

Step 3a (update particle velocity). We used the following relation to update the velocity of each particle:

$$\begin{aligned} v_{qm}(n) &= v_{qm}(n-1) + \sigma_1(1-\gamma)(l_{qm} - b_{qm}(n-1)) \\ &\quad + \sigma_2\gamma(g_{bm} - b_{qm}(n-1)). \end{aligned} \quad (15)$$

In the right hand side of (15), the first term is called inertia or momentum which shows the previous velocity. The second term shows the private thinking and is also called as cognitive term of the particle while the last term explains the collective behavior of the population and is known as a social component. At the beginning, the value of $\gamma = 0.1$ to ensure that more weightage is given to local intelligence at the beginning; then there is gradual increase in the value of γ towards 0.9 which means that more weightage is given to collective intelligence at the end. In (15), both σ_1 and σ_2 are positive constants and for the ongoing problem $\sigma_1 = \sigma_2 = 1$.

The velocity is doubly bounded; that is,

$$\begin{aligned} -v_{\max} &\leq v_{qm} \leq v_{\max}; \\ \text{if } v_{qm}(n) > v_{\max}, & \quad v_{qm}(n) = v_{\max}; \\ \text{if } v_{qm}(n) < -v_{\max}, & \quad v_{qm}(n) = -v_{\max}. \end{aligned}$$

v_{\max} and v_{\min} are taken to be 2 and -2 , respectively, in order to avoid the premature convergence.

Step 3b (update particle position). The position of each particle can be updated as

$$b_{qm}(n) = b_{qm}(n-1) + v_{qm}(n). \quad (16)$$

Step 4a (choose local best particle). If fitness $b_q(n) > l(n)$, replace l with b .

Step 4b (choose global best particle). Similarly replace the previous global best g_b particle with $b_q(n)$ as a new global best particle if fitness $b_q(n) > g_b$.

Step 5 (termination). Terminate, if any of the following conditions is satisfied; otherwise go to Step 3a:

- (i) The predefined total number of iterations/flights is executed for the algorithm.
- (ii) The desired MSE is attained which is 10^{-7} .

Step 6 (hybridization). In this step, the best particle achieved through PSO is given to ASA for further improvement.

Step 7 (storage). Store the results for later on discussion and comparison.

3.2. Differential Evolution (DE). DE was introduced by Storn and Price in 1997 [22] which iteratively searches large spaces of candidate solution that improve the candidate solution with respect to specified measurement of quality. In other words, DE deals with the optimization problem in the following way: first it achieves the candidate solution and then, by using simple formula, it generates new solution by combining with the existing ones. As a result only those candidate solutions will survive which has best fitness or score. The block diagram of DE is shown in Figure 3, while its steps are given as follows.

Step 1 (initialization). The first step is similar to the one given for PSO in (8).

Step 2 (updating). Update all particles of the current generation "ge" from 1 to Q . Let us choose q th particle from (8), that is, $\mathbf{b}_h^{q:ge}$, where $q = 1, 2, \dots, Q$ and $h = 1, 2, \dots, 4 \times K$, while "ge" represents the particular generation. The main goal of DE is to find out the particles of next generation, that is, $\mathbf{c}^{q:ge+1}$, by adopting the following steps:

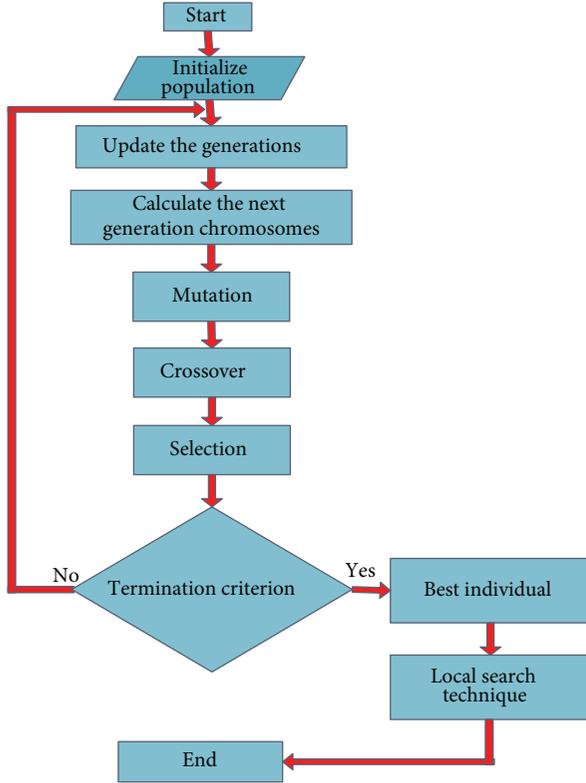


FIGURE 3: Generic flow diagram for hybrid DE.

(a) *Mutation*: select any three different numbers (particles) from 1 to Q , that is, (a_1, a_2, a_3) , by keeping the following conditions:

$$1 \leq a_1, a_2, a_3 \leq Q,$$

$$\text{where } \begin{cases} a_q \neq n_h & \forall q, h = 1, 2, 3, \\ n_q \neq q & \forall q = 1, 2, 3, \end{cases} \quad (17)$$

$$\mathbf{d}^{q,ge} = \mathbf{b}^{a_1,ge} + F(\mathbf{b}^{a_2,ge} - \mathbf{b}^{a_3,ge}),$$

where “ F ” is a constant whose value usually lies in the range 0.5 to 1.

(b) *Crossover*: the crossover can be performed as

$$\mathbf{o}_k^{q,ge} = \begin{cases} \mathbf{d}_h^{q,ge} & \text{if } \text{rand}() \leq \text{CR} \text{ or } h = h_{\text{rand}} \\ \mathbf{b}_h^{q,ge} & \text{otherwise,} \end{cases} \quad (18)$$

where $0.5 \leq \text{CR} \leq 1$ and h_{rand} is between 1 and $4 * K$ chosen at random.

(c) *Selection operation*: in this step, select the particle of next generation by using the following criterion of selection:

$$\mathbf{b}^{q,ge+1} = \begin{cases} \mathbf{o}^{q,ge} & \text{if } \text{err}(\mathbf{o}^{q,ge}) < \text{err}(\mathbf{b}^{q,ge}) \\ \mathbf{b}^{q,ge} & \text{otherwise,} \end{cases} \quad (19)$$

where the $\text{err}(\mathbf{b}^{q,ge})$ and $\text{err}(\mathbf{o}^{q,ge})$ are defined in (8). Repeat this for all particles.

Step 3 (stopping criteria). The stopping criteria of DE are based on any of the following conditions being fulfilled:

- (I) $\text{err}(\mathbf{b}^{q,ge+1}) < \epsilon$, where ϵ is a very small positive number OR.
- (II) The number of iterations is completed.

Else go back to Step 2.

Step 4 (hybridization). In order to further refine the results of DE, we give the best particle to ASA as starting point.

3.3. Active Set Algorithm (ASA). Active set methods are recursive procedures for solving sequence of equality-constrained quadratic subproblems effectively from majority of algorithms based on interior methods and simplex techniques due to the fact that they operate in two phases; first they focus on feasibility, while others focus on optimality. The applications in which their dominance is well established are quadratic programming including portfolio analysis, structural analysis, and optimal control and optimal mesh refinement in ODE or PDE constrained problems, sparse linear programming problems, and box constrained optimization problem [23–25]. Active set quadratic programming is exploited in many state-of-the-art optimization solvers such as QPOPT [26], SQOPT [27], and QPA (part of the GALAHAD library) [28]. Although the global optimization methods (PSO and DE) alone can produced excellent results, their rate of convergence usually decreases with the increase of iteration especially in nonlinear problems. Therefore, to get accurate results, we need an efficient local search optimizer like ASA. The ASA is mainly used in constrained optimization problems whose basic purpose is to transform the problem into an easier solvable problem [29]. The standard working procedure of ASA in the form of pseudocode is given as follows, while its flow diagram in terms of processing blocks is shown in Figure 4.

Generic Pseudocode of Active Set Algorithm

Start. Initialize ASA algorithm with random assignment of parameters.

Step 1. Determine a suitable starting point.

Step 2. Repeat till an optimal condition is achieved.

Step 3. Solve the equality problem defined roughly by ASA.

Step 4. Calculate the Lagrange multipliers for ASA and remove all those constraints having negative Lagrange multipliers.

End Repeat

The mathematical formulation of ASA algorithm is given as.

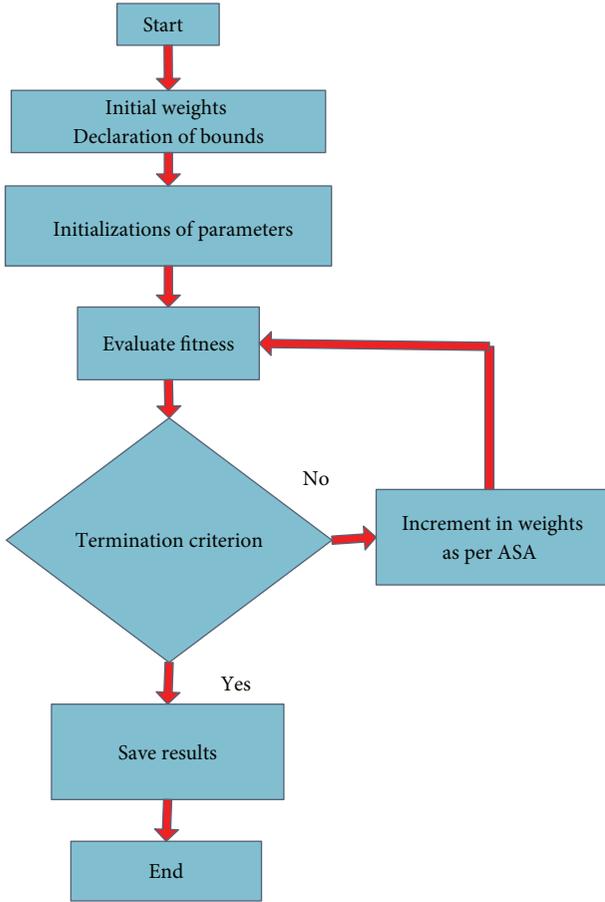


FIGURE 4: Generic flow diagram of Active Set Algorithm.

The constrained optimization problem can be mathematically written as

$$\min \{f(y)\} \quad \text{such that } l \leq y \leq u, \quad (20)$$

where l and u represent the lower and upper bounds, respectively, of the desired vectors. In early methods, penalty functions were used to transform the constrained problem into unconstrained one which is beyond or near the constrained boundary. In such a way, the constrained problems were solved by a sequence of parameterized unconstrained optimizations, which is obviously an inefficient way, that are now replaced by methods which focus on the solution of Karush-Kuhn-Tucker (KKT) equations such as ASA. The KKT equations are the essential condition for constrained optimization problem. In order to achieve the global solution point, the KKT equations are the sufficient and essential conditions to be satisfied. The KKT equations can be mathematically written as

$$\nabla f(y^*) + \sum_{i=1}^m \lambda_i \cdot \Delta G_i(y^*) = 0, \quad (21)$$

$$\lambda_i \cdot G_i(y^*) = 0, \quad i = 1, \dots, m_e, \quad (22)$$

where $\lambda_i \geq 0$, $i = m_e + 1, \dots, m$, are called Lagrange multipliers. Equation (21) represents the gradient cancellation

TABLE I: Parameters setting for ASA.

Parameters	Setting
Starting point	Best particles achieved by DE and PSO
Number of iterations	2000
Number of variables	$4 \times K$
Fitness limit	10^{-15}
Function tolerance	10^{-15}
Nonlinear constraints tolerance	10^{-15}
Maximum function evaluations	5000
X-tolerance	10^{-15}
Derivative approximate	Finite central difference

of active constraints and the objective function at the solution point where the Lagrange multipliers are necessary to balance the deviation in magnitude of the constrained gradients and objective function.

For ASA, we used a MATLAB built-in optimization toolbox where the parameter setting is provided in Table I.

4. Results and Discussion

In this section, several simulations are carried out to validate the performance of the proposed techniques based on PSO, DE, PSO-ASA, and DE-ASAs. For theoretical consideration among the optimization techniques, the strength of memetic computing techniques based on PSO-ASA and DE-ASAs is superior to the rest. Comparison of the results of these non-conventional meta-heuristic techniques is presented here in number of graphical and numerical illustrations to establish memetic optimization methodologies in terms of estimation accuracy, reliability, robustness, and convergence.

Simulation studies are present here in two parts; in part 1, we compared the performances of PSO, DE, PSO-ASA, and DE-ASA with each other in terms of estimation accuracy and convergence rate for different number of targets. In part 2, the performances of the two best techniques among them are compared with existing traditional algorithms [10, 11] by using MSE as a figure of merit. Every time, the number of sensors in both subarrays is taken to be same. In order to avoid the phase ambiguity, the interelement spacing between the consecutive elements $d = \lambda/4$ is required [10]. All the signals reflected back from targets are assumed to be statistically independent and having constant frequency. The received data at the output of each sensor are polluted by zero mean, unit variance AWGN. The values of elevation and azimuth angles are taken in degrees, while the values of ranges are taken in terms of wavelength (λ). All the simulations for PSO and DE are carried out for 1000 independent trials.

4.1. Estimation Accuracy. In this subsection, estimation accuracy of PSO, DE, PSO-ASA, and DE-ASA is discussed without adding any noise to the system. In Tables 2–4, the estimation accuracy of all schemes is provided for two, three, and four

TABLE 2: Estimation accuracy for 2 targets.

Scheme	θ_1°	ϕ_1°	$r_1(\lambda)$	s_1	θ_2°	ϕ_2°	$r_2(\lambda)$	s_2
Desired values	35.0000	73.0000	1.5000	4.0000	52.0000	105.0000	3.0000	1.0000
PSO	35.0061	73.0060	1.4951	4.0050	52.0060	105.0061	3.1032	1.0051
DE	35.0044	72.0058	1.5022	3.9977	51.0058	104.9955	3.1022	0.9976
PSO-ASA	34.9967	73.0036	1.5018	4.0019	52.0036	105.0033	3.1017	1.0020
DE-ASA	35.0015	72.0082	1.4991	3.9988	51.0082	104.9984	3.1008	0.9988

TABLE 3: Estimation accuracy for 3 targets.

Scheme	θ_1°	ϕ_1°	$r_1(\lambda)$	s_1	θ_2°	ϕ_2°	$r_2(\lambda)$	s_2	θ_3°	ϕ_3°	$r_3(\lambda)$	s_3
Desired values	80.0000	40.0000	7.0000	2.0000	65.0000	160.0000	1.0000	8.0000	25.0000	120.0000	3.0000	6.0000
PSO	80.3843	39.6156	6.8209	2.1789	64.6158	159.6156	1.3847	8.6157	24.6158	120.1789	2.8209	5.8210
DE	79.8028	40.1973	6.9026	1.9028	65.1974	160.1973	1.1979	8.1972	25.1974	119.9028	2.9026	6.0971
PSO-ASA	79.9264	40.0738	7.0192	2.0191	64.9261	160.0738	0.9261	7.9263	24.9261	120.0191	3.0192	6.0193
DE-ASA	80.0268	39.9731	6.9932	1.9931	64.9733	159.9731	1.0269	8.0260	24.9733	119.9931	2.9932	5.9932

TABLE 4: Estimation accuracy for 4 targets.

(a)

Scheme	θ_1°	ϕ_1°	$r_1(\lambda)$	s_1	θ_2°	ϕ_2°	$r_2(\lambda)$	s_2
Desired values	40.0000	120.0000	2.5000	3.0000	75.0000	50.0000	4.0000	9.0000
PSO	38.2202	121.8799	2.9797	3.8711	73.1884	48.1285	4.8686	9.9128
DE	41.2388	121.2490	2.8989	3.7010	73.8656	48.8375	4.6654	9.6655
PSO-ASA	40.4177	119.6279	2.6522	3.4805	75.4947	50.3942	4.4056	9.4060
DE-ASA	40.0709	120.0781	2.5791	3.0812	75.0869	50.0870	4.0867	9.0869

(b)

Scheme	θ_3°	ϕ_3°	$r_3(\lambda)$	s_3	θ_4°	ϕ_4°	$r_4(\lambda)$	s_4
Desired values	55.00000	160.0000	12.0000	6.0000	30.0000	220.0000	6.0000	4.0000
PSO	56.8282	161.8284	12.6582	5.2837	28.2814	218.1369	6.6182	4.8273
DE	53.8656	161.2659	12.4942	5.4766	28.7766	218.7751	5.6656	4.6659
PSO-ASA	55.4941	160.4658	12.2944	6.4041	30.5041	220.4041	6.32941	4.2946
DE-ASA	55.0870	160.0791	12.0912	6.0908	30.0988	220.0715	6.0870	4.0912

sources, respectively, for different values of elevation angle, azimuth angle, range, and amplitudes. For this, the CSCA consists of 9, 13, and 17 sensors for 2, 3, and 4 targets, respectively. One can see from Tables 2–4 that each scheme produced better estimation accuracy for two targets but their accuracy degraded for three and four targets due to the presence of local minima. From these results, the advantages of hybridization are quite obvious as the estimation accuracy of PSO alone is less as compared to DE alone but its accuracy increases than DE when it is hybridized with ASA. Overall the DE-ASA produced better estimation accuracy as compared to the other three techniques. The second and third best results are given by PSO-ASA and DE alone, respectively.

In Table 5, mean, variance, and standard deviation are carried out for all schemes in terms of RMSE. For this simulation, two targets are considered to be impinging on CSCA having nine sensors. It is apparent that again DE-ASA produced better results as compared to the other algorithms

TABLE 5: Mean, variance, and standard deviation in terms of RMSE for two targets.

Scheme	Mean	Variance	Standard deviation
PSO	$0.23E^{-02}$	$1.0E^{-03}$	$31.6E^{-03}$
DE	$5.45E^{-03}$	$1.0E^{-04}$	$10.0E^{-03}$
PSO-ASA	$3.63E^{-04}$	$1.0E^{-05}$	$3.1E^{-03}$
DE-ASA	$8.19E^{-06}$	$1.0E^{-07}$	$0.31E^{-03}$

while the second and third best results are once again given by PSO-ASA and DE alone.

4.2. Convergence. In this section, the convergence of each scheme for different number of targets is discussed. By convergence we mean the total number of times a particular scheme got its desire results. As shown in Figure 5, the convergence of each scheme decreases with the increase

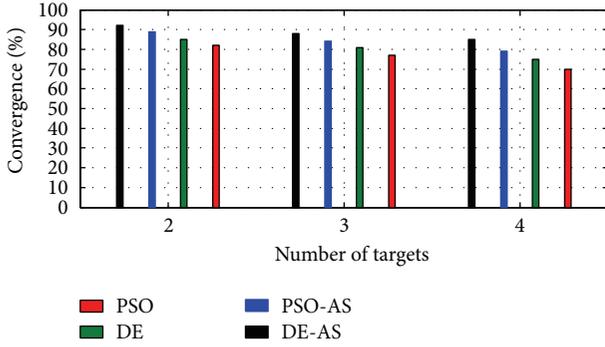


FIGURE 5: Convergence versus number of targets.

TABLE 6: Proximity effect of elevation angles for $s_1 = 1, s_2 = 3, s_3 = 5, r_1 = 1.5\lambda, r_2 = 3\lambda, r_3 = 4\lambda$, and $\phi_1 = 130^\circ, \phi_2 = 70^\circ, \phi_3 = 160^\circ$.

Scheme	θ_1°	θ_2°	θ_3°	%convergence
Desired values	35.0000	75.0000	60.0000	—
PSO-ASA	35.3843	75.3842	60.3844	89
DE-ASA	35.1791	75.1790	60.1793	95
Desired values	30.0000	65.0000	75.0000	—
PSO-ASA	30.3846	65.9832	75.9834	81
DE-ASA	30.1792	65.4301	75.4302	92
Desired values	30.0000	40.0000	50.0000	—
PSO-ASA	31.3965	41.4011	51.4013	68
DE-ASA	30.7692	40.7694	50.7690	87
Desired values	30.0000	35.0000	40.0000	—
PSO-ASA	32.3417	37.3518	42.3519	62
DE-ASA	31.1105	36.1107	41.1105	80

of unknowns (targets) in the problem. However, the convergence of hybrid schemes are less degraded and they maintained fairly good convergence every time. The first and second best schemes among them are DE-ASA and PSO-ASA as they maintained better convergence. The third best convergence is given by DE alone. From the above discussion, the first and second best schemes are DE-ASA and PSO-ASA, respectively, so, to summarize the discussion, we shall be limited to the discussion of these two hybrid schemes only in the upcoming discussion.

4.3. Proximity Effects. In this subsection, the results are carried out to assess the proximity effects of elevation and azimuth angles for DE-ASA and PSO-ASA. We considered 3 sources impinging on CSCA, composed of 13 sensors. As provided in Tables 6 and 7, both of the hybrid schemes produced fairly good estimation accuracy and convergence for closely spaced elevation and azimuth angles. However, the DE-ASA produced better results as compared to PSO-ASA.

TABLE 7: Proximity effect of azimuth angles for $s_1 = 1, s_2 = 3, s_3 = 5, r_1 = 1.5\lambda, r_2 = 3\lambda, r_3 = 4\lambda$, and $\theta_1 = 30^\circ, \theta_2 = 50^\circ, \theta_3 = 85^\circ$.

Scheme	θ_1°	θ_2°	θ_3°	%convergence
Desired values	25.0000	80.0000	240.0000	—
PSO-ASA	24.6841	79.6170	239.6254	90
DE-ASA	25.1790	79.8210	240.1793	93
Desired values	50.0000	80.0000	70.0000	—
PSO-ASA	49.6154	80.9832	70.9832	81
DE-ASA	49.8210	80.4305	70.4303	90
Desired values	50.0000	60.0000	70.0000	—
PSO-ASA	48.6031	58.5981	71.4014	70
DE-ASA	50.7694	60.7696	70.7692	85
Desired values	50.0000	55.0000	60.0000	—
PSO-ASA	52.4420	57.3519	57.6468	64
DE-ASA	51.1104	56.1105	58.8967	82

4.4. Estimation Accuracy for DOA on Reference Axis. Some of the elevation angles ($\theta = 0^\circ, 90^\circ$) and azimuth angles $\phi = (0^\circ, 90^\circ, 180^\circ)$ are considered to be critical angles where the performance of most algorithms degraded. We considered the three targets at $(\theta_1 = 0^\circ, \phi_1 = 90^\circ, r_1 = 3\lambda, s_1 = 1)$, $(\theta_2 = 90^\circ, \phi_2 = 0^\circ, r_2 = 1\lambda, s_2 = 2)$, and $(\theta_3 = 35^\circ, \phi_3 = 180^\circ, r_3 = 5\lambda, s_3 = 4)$. In Figure 6(a), the estimation accuracy of both hybrid schemes for elevation angles is evaluated on reference axis (0° and 90°). It can be seen that both hybrid schemes produced significant error (6° to 8°) especially for 90° . However, the DE-ASA produced comparatively less errors as compared to PSO-ASA.

In Figure 6(b), the estimation accuracy of both hybrid schemes is evaluated for azimuth angles at $0^\circ, 90^\circ$, and 180° . In this case, both schemes are up to the mark and produced negligible error. Once again the DE-ASA produced better results as compared to PSO-ASA.

4.5. Comparison with Other Techniques Using Root Mean Square Error (RMSE). In this subsection, we compared the RMSE of DE-ASA and PSO-ASA with existing nonheuristic techniques [10, 11] in the presence of noise. We considered two sources, $(\theta_1 = 25^\circ, \phi_1 = 55^\circ, r_1 = 3.5\lambda, s_1 = 1)$ and $(\theta_2 = 60^\circ, \phi_2 = 170^\circ, r_2 = 1\lambda, s_2 = 3)$, where the CSCA consist of 17 sensors. In Figures 7(a)–7(c), both hybrid schemes produced lower RMSE as compared to the algorithms described in [10, 11]. The RMSE of both hybrid schemes is lower for the target located near to the array as compared to the target located comparatively away from the array as shown in Figure 7(c). Among all of them, the DE-ASA produced lower RMSE while the second best RMSE is obtained by the PSO-ASA scheme.

One of the other features of both hybrid schemes is that they can also be used for the amplitude estimation which is missing in [10, 11]. In Figure 7(d), RMSE is shown against SNR for the amplitude of the targets. Once again the DE-ASA scheme maintained a lower RMSE for both targets.

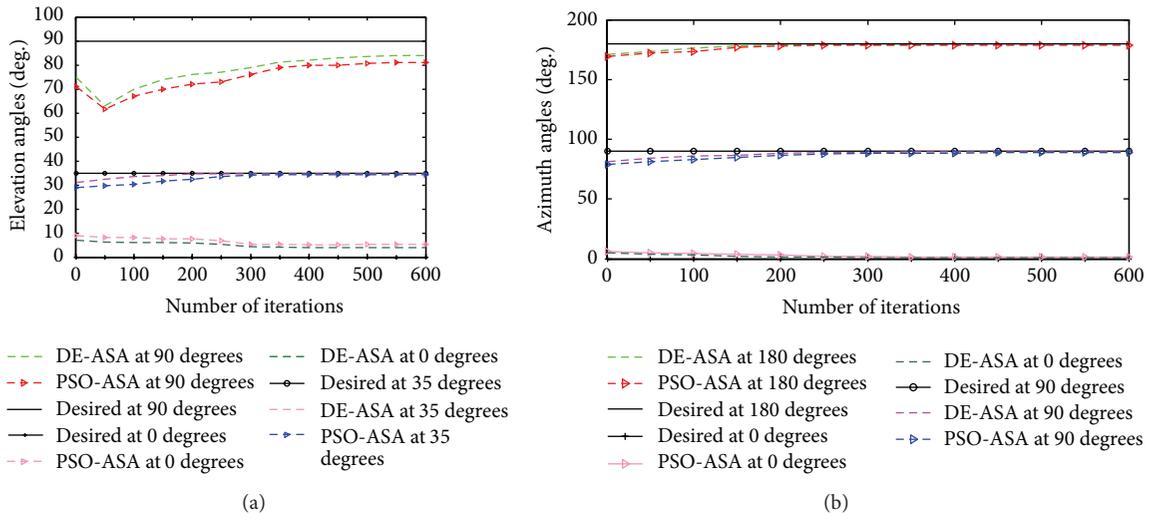


FIGURE 6: (a) Elevation angle estimation on reference axis. (b) Elevation angle estimation on reference axis.

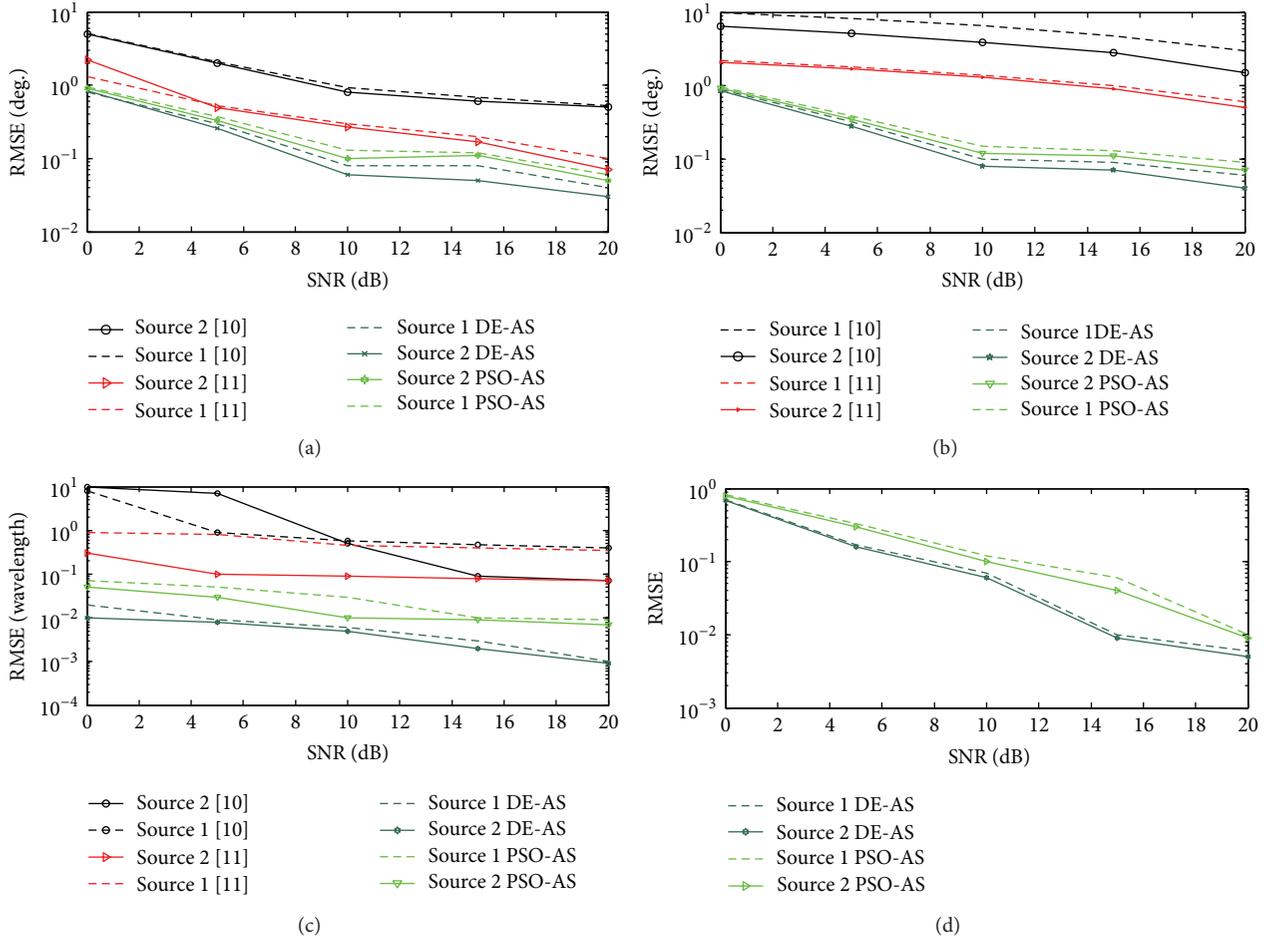


FIGURE 7: (a) Root mean square error of elevation angles versus SNR. (b) Root mean square error of azimuth angles versus SNR. (c) Root mean square error of ranges versus SNR. (d) Root mean square error of amplitudes versus SNR.

5. Conclusion and Future Work Direction

In this work, we developed Heuristic Computational Intelligence techniques to estimate jointly the amplitude, range, and elevation and azimuth angle of multiple targets impinging on the passive CSCA. In these computational techniques, first Particle Swarm Optimization (PSO) and Differential Evolution (DE) were used alone and then, to improve the results further, they were hybridized with ASA. It has been found that the hybrid DE-ASA schemes produced better results as compared to PSO and DE in terms of estimation accuracy and convergence and have also shown better results as compared to nonheuristic techniques in terms of RMSE.

Besides all the advantages, the proposed schemes have also few limitations:

- (i) All the schemes fail if the number of targets increases to that of the number of sensors in the CSCA.
- (ii) The performance of the proposed schemes degraded when the number of sources is more than four.

In the future, one can use such kind of techniques in the following areas:

- (i) This kind of model for finding range, amplitude, and elevation angle can effectively be used for spectrum sensing in the cognitive radio domain in which the source shall be mainly primary user station. With each snapshot, one can find these updated parameters.
- (ii) One can use such kind of algorithms for null steering, sidelobe reductions, and main beam steering.

Conflict of Interests

All the authors of this paper declare that there is no conflict of interests regarding its publication.

References

- [1] S. O. Al-Jazzar, A. Muchkaev, A. Al-Nimrat, and M. Smadi, "Low complexity and high accuracy angle of arrival estimation using eigen value decomposition with extension to 2D AOA and power estimation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, article 123, 2011.
- [2] R. Roy and T. Kailath, "ESPRIT—estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.
- [3] J. Liang, X. Zeng, B. Ji, J. Zhang, and F. Zhao, "A computationally efficient algorithm for joint range-DOA-frequency estimation of near-field sources," *Digital Signal Processing*, vol. 19, no. 4, pp. 596–611, 2009.
- [4] Y. Wu, L. Ma, C. Hou, G. Zhang, and J. Li, "Subspace-based method for joint range and DOA estimation of multiple near field sources," *Signal Processing*, vol. 86, no. 8, pp. 2129–2133, 2006.
- [5] S. Darzi, T. S. Kiong, M. T. Islam, M. Ismail, S. Kibria, and B. Salem, "Null steering of adaptive beamforming using linear constraint minimum variance assisted by particle swarm optimization, dynamic mutated artificial immune system, and gravitational search algorithm," *The Scientific World Journal*, vol. 2014, Article ID 724639, 10 pages, 2014.
- [6] T. S. Kiong, S. B. Salem, J. K. S. Paw, K. P. Sankar, and S. Darzi, "Minimum variance distortionless response beamformer with enhanced nulling level control via dynamic mutated artificial immune system," *The Scientific World Journal*, vol. 2014, Article ID 164053, 9 pages, 2014.
- [7] S. Leng, W. Ser, and C. C. Ko, "Adaptive beamformer derived from a constrained null steering design," *Signal Processing*, vol. 90, no. 5, pp. 1530–1541, 2010.
- [8] N. Kabaoglu, H. A. Cirpan, E. Cekli, and S. Paker, "Deterministic maximum likelihood approach for 3-D near field source localization," *International Journal of Electronics and Communications*, vol. 57, no. 5, pp. 345–350, 2003.
- [9] J. Liang, S. Yang, J. Zhang, L. Gao, and F. Zhao, "4D near-field source localization using cumulant," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, Article ID 017820, 10 pages, 2007.
- [10] R. N. Challa and S. Shamsunder, "Passive near-field localization of multiple non-Gaussian sources in 3-D using cumulants," *Signal Processing*, vol. 65, no. 1, pp. 39–53, 1998.
- [11] J. Liang, D. Liu, X. Zeng, W. Wang, J. Zhang, and H. Chen, "Joint azimuth-elevation/(-range) estimation of mixed near-field and far-field sources using two-stage separated steering vector-based algorithm," *Progress in Electromagnetics Research*, vol. 113, pp. 17–46, 2011.
- [12] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE, Piscataway, NJ, USA, 2nd edition, 2000.
- [13] D. B. Fogel and L. J. Fogel, "Using evolutionary programming to schedule tasks on a suite of heterogeneous computers," *Computers and Operations Research*, vol. 23, no. 6, pp. 527–534, 1996.
- [14] P. J. Angeline, G. M. Saunders, and J. B. Pollack, "An evolutionary algorithm that construct recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 54–64, 1994.
- [15] F. Zaman, I. M. Qureshi, A. Naveed, J. A. Khan, and R. M. A. Zahoor, "Amplitude and directional of arrival estimation: comparison between different techniques," *Progress in Electromagnetics Research B*, vol. 39, pp. 319–335, 2012.
- [16] F. Zaman, I. M. Qureshi, A. Naveed, and Z. U. Khan, "Joint estimation of amplitude, direction of arrival and range of near field sources using memetic computing," *Progress In Electromagnetics Research C*, vol. 31, pp. 199–213, 2012.
- [17] H.-P. Schwefel, *Evolution and Optimum Seeking*, Sixth-Generation Computer Technology Series, John Wiley & Sons, New York, NY, USA, 1995.
- [18] D. B. Fogel, "The advantages of evolutionary computation," in *Proceedings of the Biocomputing and Emergent Computation (BCEC '97)*, Skovde, Sweden, September 1997.
- [19] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, Perth, Australia, December 1995.
- [20] F. Zaman, *Estimation of direction of arrival for adaptive beamforming [Ph.D. dissertation]*, 2013.
- [21] S. H. Ling, K. Y. Chan, F. H. Leung, F. Jiang, and H. Nguyen, "Quality and robustness improvement for real world industrial systems using a fuzzy particle swarm optimization," *Engineering Applications of Artificial Intelligence*, 2015.

- [22] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [23] S. W. Sloan, "A steepest edge active set algorithm for solving sparse linear programming problems," *International Journal for Numerical Methods in Engineering*, vol. 26, no. 12, pp. 2671–2685, 1988.
- [24] W. W. Hager and H. Zhang, "A new active set algorithm for box constrained optimization," *SIAM Journal on Optimization*, vol. 17, no. 2, pp. 526–557, 2006.
- [25] P. E. Gill and E. Wong, "User's guide for SNCTRL, an interface for SNOPT 7," Numerical Analysis Report 11-1, Department of Mathematics, University of California, San Diego, Calif, USA, 2011.
- [26] P. E. Gill, W. Murray, and M. A. Saunders, "User's guide for QPOPT 1.0: a fortran package for quadratic programming," Report SOL 95-4, Department of Operations Research, Stanford University, Palo Alto, Calif, USA, 1995.
- [27] P. E. Gill, W. Murray, and M. A. Saunders, "User's guide for SQOPT version 7: software for large-scale linear and quadratic programming," Numerical Analysis Report 06-1, Department of Mathematics, University of California, San Diego, Calif, USA, 2006.
- [28] N. I. M. Gould, D. Orban, and P. L. Toint, "GALAHAD, a library of thread-safe fortran 90 packages for large-scale nonlinear optimization," *ACM Transactions on Mathematical Software*, vol. 29, no. 4, pp. 353–372, 2003.
- [29] M. A. Z. Raja, "Stochastic numerical treatment for solving Troesch's problem," *Information Sciences*, vol. 279, pp. 860–873, 2014.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

