applied sciences

MDPI

Article

An Improved Chaos Driven Hybrid Differential Evolution and Butterfly Optimization Algorithm for Passive Target Localization Using TDOA Measurements

Maja Rosić [1,*], Miloš Sedak [1], Mirjana Simić [2] and Predrag Pejović [2]

[1] Faculty of Mechanical Engineering, University of Belgrade, 11000 Belgrade, Serbia
[2] School of Electrical Engineering, University of Belgrade, 11000 Belgrade, Serbia
* Correspondence: mrosic@mas.bg.ac.rs

Abstract: This paper addresses the problem of time difference of arrival (TDOA) based passive target localizationand proposes an improved chaos-driven hybrid differential evolution (DE) algorithm and butterfly optimization algorithm (BOA), named ICDEBOA, to solve this complex optimization problem. The proposed algorithm consists of a new mutation strategy with the mechanisms of the BOA algorithm incorporated into the DE algorithm. To boost optimization effectiveness, chaos theory is employed to adjust the control parameter values. The considered localization problem is formulated using the maximum likelihood estimator. To perform the accuracy comparison, the convex constrained weighting least squares algorithm is applied to the considered localization problem as the widely used method in literature. The statistical analysis shows that the proposed modifications to the ICDEBOA algorithm improve its optimization performance, as demonstrated by the improved performance on the CEC2014 benchmark problems. The ICDEBOA algorithm is also shown to be more robust than existing algorithms in noisy environments. Numerical simulation results show that the proposed ICDEBOA algorithm meets the CRLB and achieves better performance than the CWLS, DE, and BOA algorithms.

Keywords: localization; time difference of arrival; butterfly optimization algorithm; hybrid optimization; diferential evolution; Cramer-Rao lower bound

Citation: Rosić, M.; Sedak, M.; Simić, M.; Pejović, P. An Improved Chaos Driven Hybrid Differential Evolution and Butterfly Optimization Algorithm for Passive Target Localization Using TDOA Measurements. *Appl. Sci.* **2023**, *13*, 684. https://doi.org/10.3390/app13020684

Academic Editor: Giancarlo Mauri

Received: 12 November 2022
Revised: 23 December 2022
Accepted: 30 December 2022
Published: 4 January 2023

1. Introduction

Accurate localization based on difference of arrival (TDOA) data collected from a single receiver and a set of transmitters is essential for many wireless sensor network (WSN) applications [1,2]. The majority of localization technologies that have been developed and are widely in usage, such as the global positioning system (GPS) [3], ultrasonic-based systems [4], infrared-based systems [5], and RF-based systems [6], require the attachment of an additional device with its own power source to localize the target. Systems such as these are known as "active localization systems," since the target directly obtains the range measurements, thus being directly involved in localization. Recent research focuses on device-free passive localization, in which the target to be located does not need to be equipped with additional devices or take any active part in the localization process and simply reflects the signals coming from the transmitters [7]. In practice, the GPS, as a prominent active system, shows significant drawbacks, such as excessive power consumption and poor localization performance in cases with weak satellite signals such us indoor areas [3]. Passive target localization has become one of the most popular techniques used to locate the passive targets in a variety of applications, including emergency rescue, intelligent transportation systems, radar, and mobile phone localization, due to its effectiveness as a replacement for GPS and other traditional positioning systems [8,9]. The vast majority of TDOA-based localization techniques in the literature are classified as "active systems". In this paper, we outline the development of a device-free passive target localization system

for real-world environments based on TDOA measurements, along with the appropriate design of a metaheuristic hybrid algorithm to obtain a solution to the considered problem.

Therefore, this work considers the passive target localization model, where the positioning is performed based on TDOA measurements under the noisy conditions. The TDOA measurements are produced by subtracting the required time propagation of a signal emitted from the transmitter over the target reflected to a receiver from the direct signal propagation time captured at the receiver. Due to a nonlinear connection between the measured distances and the target's true location, estimating the target's unknown position is a difficult process. A reasonable assumption for the error distribution of TDOA measurements when modelling the considered passive target localization problem in a line of sight (LOS) environment is to model the measurement error as Gaussian-distributed random variables [2]. A number of estimation techniques, including nonlinear least squares (NLS) [10] and maximum likelihood (ML) [11] estimators, that do not have a closed form solution, are used in literature to tackle localization problems. With the linear least squares (LLS) and weighed linear least squares (WLS) methods, the NLS method's nonlinear objective function can be made linear, thus providing a closed-form solution [12]. The constrained weighted least squares (CWLS) approach is introduced to provide a solution, in closed form, that has higher quality [13]. In this regard, the relationship between the target location and an auxiliary variable is introduced into the CWLS optimization problem as the equality constraint that has the highest term of the second order.

The ML estimator is another widely used estimation method that can improve estimation accuracy. When the statistical information regarding the distribution of the error in measurements is identified, the ML estimator method is straightforward to apply [11]. Due to the pronounced nonentities and complex landscape of the ML objective function, it is difficult to obtain a global optimum solution. In order to solve this type of optimization problem, various efficient optimization algorithms have been suggested in the literature. The semidefinite programming (SDP), which transforms the multimodal optimization problem into a convex one by applying the corresponding relaxations, is commonly used [2]. It has been demonstrated, however, that the SDP approach has some drawbacks that impact the accuracy of the global optimal solution, particularly when the measurement noise level is high [14]. In this context, obtaining a high precision solution that represents global optimum to the nonlinear, non-convex ML objective function has become a formidable obstacle.

This paper presents evolutionary algorithms (EAs) as a way to overcome these limitations while searching for the global optimum solution of a multimodal and complex target estimation problem [15–17]. In the development of the EAs, the exploration and exploitation phases have a crucial impact on the obtained solutions [18]. EAs perform exploration in searching for a promising location of the global optimum, whereas exploitation is performed when looking to improve the obtained global best solution. Therefore, to achieve a promising solution, the optimization algorithm must provide an effective balance between these two phases.

Various metaheuristic algorithms, such as the genetic algorithm (GA) [19], particle swarm optimization (PSO) [20], butterfly optimization algorithm (BOA) [21], differential evolutionary (DE) [22], cuckoo search algorithm (CS) [23], artificial bee colony (ABC) [24], gray wolf optimizer (GWO) [25], whale optimization algorithm (WOA) [26], and firefly algorithm (FA) [27], have been developed and applied to various optimization problems using key characteristics of these biological systems. BOA and DE algorithms have been successfully used in recent years to identify the global optimal solution to a variety of complex optimization problems due to their advantages such as ease of development and fast convergence [28–30].

Differential evolution is a stochastic population-based evolutionary algorithm that employs the recombination approach, which creates new solutions depending upon the weighted difference between two randomly selected population members added to a third population member to direct the search in the solution space. Storn and Price [22] first proposed this method for minimizing nonlinear functions that are not differentiable. Since

then, DE has proven to be highly effective in a wide range of scientific optimization problems thanks to its ability to prevent premature and encourage rapid convergence as well as guarantee population diversity [16,31]. Like other EAs, it employs a population of candidate solutions and searches stochastically by applying mutation, crossover, and selection operators to guide the population toward the promising solutions. The simplicity of conventional DE lies in the fact that the crossover rate, scale factor, and population size are the only three control parameters that need to be adjusted. The mutation technique and the selection of the control parameters have a significant impact on the efficiency with which DE solves a given optimization problem [28]. In certain optimization situations, DE is prone to converge prematurely to local optima and exhibits poor convergence late in the search process, motivating researchers to develop new enhanced variants of the DE algorithm to improve the optimization performance of the DE to overcome these challenges [32,33]. There are a variety of mutation strategies in the literature. This is because the choice of mutation strategy has a strong effect on how well an optimization problem is solved. In this way, DE/rand/1 and DE/rand/2 are better for exploration, whereas the DE/current-to-best/1 and DE/current-to-best/2 are better for local searches because they focus on the best candidate in the current population [28]. To boost optimization effectiveness, a new DE algorithm, JADE [33], has been proposed, with a new mutation strategy, "DE/current-to-p best" as an extension to the traditional "DE/current-to-best/1", along with an optional external archive. Parameter adaptation eliminates the need for experts to be aware of the specifics of optimization problems by automatically updating the control parameters to suitable values. Results from simulations demonstrate that JADE is superior to both traditional DE algorithms and adaptive variants. Built upon the JADE, the researchers developed a parameter adaptive DE algorithm (SHADE) that stores the parameters used in previous successful evolution rounds [34]. jDE (self-adaptive tuning parameters) is another well-known variant of DE algorithm that has shown acceptable performance on benchmark problems, which uses an evolution process to fine tune the optimization algorithm parameters [35].

BOA is a novel optimization technique that is based on the foraging and mating behaviour of butterflies [21] . As they search for food, butterflies emit a fragrance, the intensity of which is proportional to the abundance of food sources. Other butterflies in the swarm can detect the strength of the fragrance released by other butterflies to estimate the likely location of a food supply or mate. Therefore, the BOA algorithm imitates feeding and mating habits of biological butterflies, assuming that a butterfly would produce a fragrance of varying strength, which is associated with the butterfly's objective function value. The fragrance will propagate over a distance, and other butterflies can sense it. This is how the butterflies can share their personal information with other butterflies and form a collective social knowledge network. Due to its simplicity and effectiveness, BOA has been suggested to solve a variety of engineering applications, from WSN sensor localization [29], engineering design problems [36], and neural network training [30], etc. Despite the fact that the BOA has shown better performance in real-world applications compared to other meta-heuristic algorithms, it also faces many challenging issues. For instance, the basic BOA is easy to fall into the local optimum when solving highly complex problems, and the convergence speed will reduce greatly in the late iteration process [37]. Therefore, many improved variants of BOA have been suggested to overcome the two aspects [36,38].

In order to further improve the optimization performance on complex optimization problems, researchers resorted to hybridizing two or more algorithms, thus creating one new algorithm that exploits the complementary strengths of each method. Some examples of the combination of two meta-heuristic algorithms include: the HFPSO [39] algorithm, where the FA and PSO algorithms are hybridized with the aim of increasing accuracy and improving convergence; a DE/WOA [26] hybrid algorithm, which combines local search abilities of WOA with DE global search; HPSOBOA [40], which combines BOA and PSO algorithms; etc. In a recent study, it was proposed to combine the butterfly optimization algorithm (BOA) and differential evolution (DE), named HBODEA [41], in order to more

effectively utilise each method's advantages. Two enhancements—mutation and crossover operations of DE and dynamic adaptive operators—are added to the original BOA in the proposed identification strategy to increase global search capability and decrease the risk of being trapped in a local optimum. However, this hybridization left a lot of different improvements and variations to be performed. As a result, all of these papers demonstrate that the hybridization of EAs speeds up the process of achieving the best solution, prevents the issue of premature convergence, and yields better answers.

Numerous applications of nonlinear dynamics, especially those incorporating chaos, have gained prominence as a consequence of the development of nonlinear dynamics and are now commonly employed to enhance the effectiveness of various optimization techniques [42]. The infinitely unstable periodic motions that characterise the bounded dynamic behaviour of non-linear systems is characterised as chaos [43]. One of the most effective methods for improving the performance of metaheuristic algorithms is to use chaos maps, which generate bounded random numbers based on initial conditions. In the literature, there are various chaos maps, such as the piecewise map, iterative map, logistic map, sine map, Tent map, and so on [44]. Chaos has been used in the self-adaptive chaos differential evolution (SACDE) [45], where the mutation factor is controlled using chaos maps and dynamically changing weighting factors and crossover factors are also introduced. Recently, a new hybrid optimization method based on differential evolution (DE) and chaos theory was made [46]. This method uses chaotic sequences to change DE parameters automatically in order to help the algorithm to escape local minimums and improve global convergence. For the purpose of tackling three engineering optimization problems, Arora and Singh [36] suggested an enhanced butterfly optimization technique with several chaotic maps. Here, three chaotic BOA algorithms were created. The first one used chaos maps to generate initial solutions, the second used chaos to control switch probability p, and the third was a combination of the previous two. The results showed that BOA modified with chaotic maps shows improved performance in terms of avoiding local optima and accelerating convergence. To prove that the suggested chaotic BOA is correct, engineering design issues, benchmark test functions, and both multimodal and unimodal benchmarks are used. As a result, the pseudo-randomness, ergodicity, and irregularity of chaos maps make them an ideal addition to EAs, helping to boost the efficiency of optimization and the quality of the solutions they provide [36,47].

Motivated by the above considerations, we recognise the importance of passive localization, especially in the indoor environment, where the traditional techniques cannot be applied with the required accuracy. By also looking at the above strengths and weaknesses of the BOA and DE algorithms, we can see that the DE algorithm has a good ability to explore, which depends on the mutation scheme chosen, and that the BOA algorithm has a good ability to search locally. Therefore, in this study, we suggest ICDEBOA, a hybridization between BOA and DE, to effectively tackle the passive target localization problem. In the suggested approach, we add a new composite roulette wheel-based mutation operator into the DE algorithm. To take advantage of the BOA algorithm's good local search ability, a hybrid algorithm is created by using BOA expressions for particle movement to produce a mutant vector in the DE algorithm. Furthermore, an adaptive technique incorporating chaos map is used to update the BOA algorithm's sensory fragrance.

In the context of range-based target localization, the CRLB is usually utilised to access the statistical performance of various localization algorithms. As a result, determining CRLB as an optimality criterion for the considered localization problem is a critical issue [48]. In this aspect, the CRLB establishes the physical impossibility of an unbiased estimator's variance being less than the given bound. Consequently, in this research, we define the CRLB for the investigated localization problem and use it to evaluate the root-mean-square error (RMSE) of the presented ICDEBOA technique and compare against that of several other widely used methods.

Given the aforementioned information, the primary goals of this work were as follows:

- To formulate a new problem of localization of a passive target using the obtained TDOA measurements.
- To present a new global optimization method that is a hybridization of BOA and DE algorithms, named ICDEBOA, so that the passive target localization problem can be solved successfully.
- To validate the proposed algorithm's performance on benchmark test functions by comparing simulation results to a number of state-of-the-art methods.
- To show how well the proposed method works by comparing simulation results on an example localization system to well-known approaches and derived CRLB.

The remaining sections of the paper are structured as follows. In Section 2, relevant work on localization and the application of evolutionary algorithms to localization problems is presented. In Section 3, the general localization problem of a passive target using TODA measurements is formulated. In Section 4, the derivation of the appropriate ML estimator for the considered localization problem is presented. In Section 5, the formulation of the CWLS is outlined. Section 6 describes differential evolution algorithm with the employed improvements. The procedure of the BOA algorithm along with the improvements is given in Section 7. The derivation of ICDEBOA is proposed in Section 8. Section 9 describes the CRLB in the context of the considered localization. In Section 10, numerical experiment results are presented. In Section 11, conclusions and future work are stated. The detailed derivation of CRLB for passive TDOA-based localization is presented in Appendix A, while the detiled derivation of CWLS is provided in Appendix B.

2. Background and Related Work

Finding a passive target has gained interest in recent years due to its wide applications, especially in indoor positioning and defensive systems where GPS signals are unavailable [3]. In general, solutions for indoor localization fall into two categories: range-based and range-free. Lateration, such as trilateration and multilateration, is the most widely employed range-based localization approach. Here, the range is defined as the distance between the target and the receivers. It can be derived from multiple signal parameters, including time of arrival (TOA) [49], TODA, received signal strength (RSS) [50], angle of arrival (AOA) [51], or a combination of these parameters [52]. It has been demonstrated that the TDOA parameter delivers more accurate and straightforward localization compared to localization techniques based on other parameters [13]. As an alternative to range-based localization techniques for indoor environments, range-free localization techniques such as the Distance Vector Hop (DV-Hop) [53], centroid or weighted centroid approach [54], and approximation point-in-triangulation test (APIT) [55] are extensively utilised. These technologies are both affordable and easy to apply since they do not necessitate complicated equipment structure. However, the range-based localization methods for indoor positioning are more accurate compared to range-free methods [56].

The formulated passive target localization problem using TDOA measurements is a nontrivial task due to its high nonlinearity and nonconvexity. In order to solve such complex localization problems, in the literature, the most widely used methods include the NLS [10] and ML [11] estimators. The NLS estimator is based on finding the extremum of the objective function, which is formulated as the sum of the squared measurement errors. However, when solving the NLS estimator with conventional gradient based methods, it can be shown to have some convergence issues, and there is a need for a promising intial solution. LLS methods are often used to linearize the NLS estimation problem so that it can be solved in closed form [12]. However, it has been shown that the CRLB can only be reached with LLS methods when the measurement errors are small enough. To improve the accuracy and provide a closed-form solution to the localization problem, a two-step weighted least-squares (2WLS) method is put forward to approximate the NLS function [12]. The two-step WLS method rearranges the nonlinear TDOA equations into a set of linear equations by introducing an auxiliary variable [57]. In the first step, it estimates the source position and auxiliary variable independently, and then in the second step it

refines the estimate by considering their connection. To further improve the accuracy and avoid ill-matrix problems, the CWLS method is proposed, which explicitly incorporates the relationship between the source position and the auxiliary variable as a constraint [58]. In this way, the CWLS method formulates the quadratically constrained quadratic problem (QCQP), which is a non-convex problem. However, it can be converted to a convex problem by exploiting the hidden convexity in the constraint.

Another widely applied estimator is the ML estimator, which has been regarded as an optimal method for target estimation [11]. However, the formulation of the ML target localization problem using TDOA measurements is a non-convex optimization problem. To address this issue, a SDP method is proposed, which relaxes the non-convex ML optimization to a convex SDP problem [59]. However, as shown in the literature, the performance of the SDP method is not satisfactory in the presence of a large measurement error. In the literature, evolutionary algorithms (EAs) are used to deal with the fact that the ML estimation problem is not convex. This is because EAs can find the best global solution to such problems.

Different kinds of EAs, including FA, PSO, and BOA, etc., have been successfully utilized during the last few years in solving different localization problems [29]. Since the performance of a wireless sensor network is significantly impacted by the accuracy of the sensor node localization, Arora and Singh [29] proposed the BOA method, a nature-inspired optimization algorithm, to reduce localization errors when calculating the position of sensor nodes in WSNs. In comparison to the PSO and FA based node localization systems, the proposed method demonstrates more consistent and precise node positioning resuls. Researchers suggested using PSO and TDOA combined, which significantly improves positioning accuracy [60]. Comparisons with established methods like WLS and LLS demonstrate that PSO is superior. In addition, the ML estimator is utilised to create the objective function for the hybrid TDOA/AOA location estimation issue, and the PSO improved with chaos is offered as a solution [15]. Comparative investigation demonstrates that the upgraded PSO algorithm outperforms the traditional optimization algorithms in terms of global search capability. To improve localization accuracy in WSNs while using fewer power and resources, researchers presented a two-stage PSO algorithm for locating wireless sensor nodes in concave space [61]. The authors proposed employing a TDOA passive positioning sensor selection approach based on the tabu search to balance the positioning accuracy of the sensor network and system utilization [62]. Consideration is given to the passive TDOA model, with sensor position errors taken into consideration, and the tabu search method is used to solve the transformed sensor selection problem. Simulation findings demonstrate that the performance of the suggested sensor optimization approach closely resembles that of the exhaustive search. A genetic algorithm with the flexibility to be modified for varied circumstances and ground models is proposed to select the appropriate node localization in order to decrease the CRLB in each sensor of an asynchronous TDOA architecture [63]. The proposed methodology allows for the optimization of a passive A-TDOA architecture's 3D sensor deployment, including ground modeling flexibility and the reduction of spatial discretization to achieve better results, which is critical for precision applications and indoor and outdoor navigation of automatic vehicles such as automatic ground vehicles and unmanned aerial vehicles.

It has been established that combining two or more EAs can improve solution quality and boost optimization performance. Combining the WLS and FA algorithms, the Hybrid Firefly Algorithm (hybrid-FA) is proposed for estimating target coordinates from TDOA data [27]. According to the simulation findings, the hybrid-FA approach provides better performance and localization accuracy than the state-of-the-art localization approaches. Furthermore, the improved DE algorithm has been successfully hybridised with the FA method and deployed to address the passive target localization problem utilising TOA measurements [16]. The results reveal that the proposed method outperforms the standard algorithms in terms of localization performance. GA is hybrdized with the DE algorithm in order to improve the perofrmance in the WSN localization problem [64]. In addition, com-

bining the DE algorithm with the standard direct search Nelder-Mead approach yielded higher quality solutions for an objective function defined using the ML method [65]. Moreover, a number of hybrid EA variants, such as DE and FA (HFDLA) [66], HPSOVNS [67], and PSO and variable neighbourhood search (HPSOVNS), are used successfully to improve positioning accuracy.

The techniques for passive localization have found a wide range of applications in recent years due to their ability to accurately locate a target without relying on active sensors or emitting radio signals. Based on TDOA data, the authors provided an experimental evaluation of multipath mitigation in two distinct interior settings utilizing a beam steering broadband circular polarization antenna (BSBCPA) in an indoor passive localization system [68]. Experiments are proposed using LOS and complicated multipath while delivering a signal with a 20 MHz bandwidth via WLAN. Compared to traditional omnidirectional linear polarization antennas, they discovered that the BSBCPA can enhance positioning accuracy by at least 51%. The experimental findings demonstrate that the proposed BSBCPA mitigates multipath propagation for TDOA-based indoor passive localization in a considerable manner. Passive localization found application during the launch mission of a launch vehicle. Using ground-based telemetry stations increases the reliability of the launch vehicle's flight safety control system [69]. The authors applied passive TDOA/AOA localization utilizing two ground telemetry stations and developed a strategy to increase the accuracy of its localization based on second-order polynomial regression. By comparing the findings of the suggested approach to the measured data from an on-board GPS utilized during the third flight of the Korea Space Launch Vehicle-1, the localization accuracy has been enhanced. For security applications, the authors presented a passive localization method based on TOA measurements on the most recent 802.11ac WiFi standard, taking advantage of the widespread deployment of WiFi systems, which typically use RSS parameters for localization [70]. Using the experimental setting, a prototype was created that demonstrated a considerable improvement in positioning accuracy over previous methods. Furthermore, by leveraging the WiFi standard, the authors propose a passive localization tracking system to predict the footprints of suspects. Using ultra-wideband radio signals is widely used for the implementation of passive localization systems. For the passive localization of fixed targets based on the hybrid RSS/TDoA method, the authors presented an algorithm for the optimal estimation of the amplitude and propagation delay time of an ultra-wideband radio signal [71]. An algorithm has been developed for the optimal estimation of distances from the radiation source to base stations, based on the results of the measurements of the amplitude and the propagation delay time of the UWB radio signal. Localization in crowded urban areas is another intriguing application area. Here, sensors are carried by flying unmanned aerial vehicles to precisely locate the signal source passively, making it possible to establish a wireless sensor network with the ideal geometry configuration [72]. According to the min-max criterion, the authors presented a condensed formulation of the TDOA localization method. The nonlinear equations are relaxed using semidefinite programming to produce an initial solution, which is then used as the starting point of the iterative algorithm to refine the solution in order to determine the source's location with accuracy.

Based on the previous, this paper proposes an enhanced ICDEBOA method as the hybridization of two well-proven EAs, with the aim of improving the accuracy of estimating passive target location based on solving an ML objective function formulated using TDOA measurements, particularly when measurement noise is high.

3. Localization Problem

The proposed passive target localization problem uses TDOA signal parameter to estimate the passive target's unknown position under the conditions of LOS environment. Figure 1 illustrates the considered localization system, which consists of a set of transmitters $\mathbf{x}_i^t = \begin{bmatrix} x_i^t & y_i^t \end{bmatrix}^T \in \mathbb{R}^2, \quad \forall i \in \{1, 2, ..., N\}$, whose positions are previously known and the passive target $\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T \in \mathbb{R}^2$ at the unknown coordinates x and y. The target is

illuminated by the signals coming from the transmitters, which is then reflected from the target and captured at the receiver, which is, without the loss of generality, placed in the origin $\mathbf{x}_r = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$.

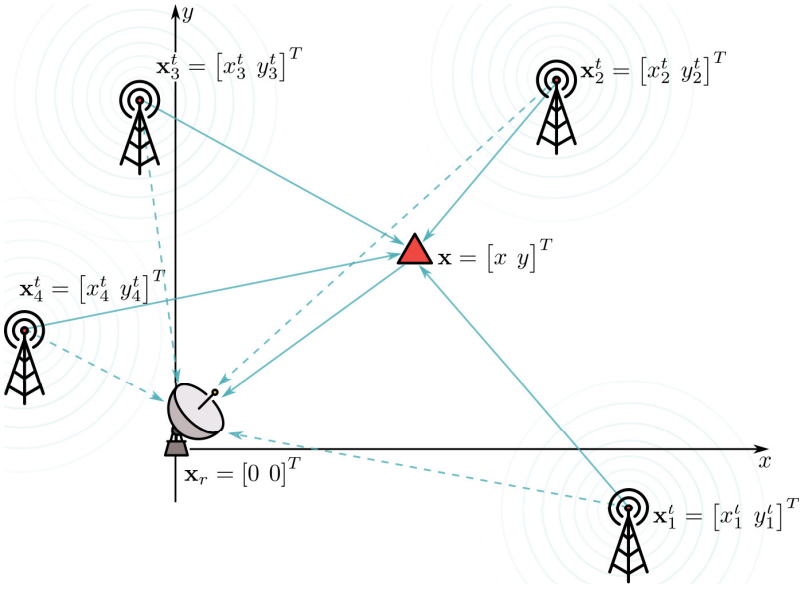

Figure 1. Illustration of the plaement of transmitters, receiver and a target and a distribution of signals in a considered localization problem.

Here, we model TDOA measurement error using the Gaussian distribution, which in a LOS context, is a justified proposition [2]. Assuming the time synchronisation of $\mathbf{x}_i^t, \forall i \in \{1, 2, ..., N\}$ and $\mathbf{x}_r$ is obtained and that the passive target simply reflects signals back and forth between $\mathbf{x}_i^t$ and $\mathbf{x}_r$ [48], the noisy TDOA measurements are obtained as the signal time difference

$$t_{\mathrm{i}} = t_{r,i} - t_{d,i}, \quad i \in \{1, 2, ..., N\} \tag{1}$$

where $t_{r,i}$ denotes the TOA of a signal which is reflected from the target coming from the ith transmitter and captured at the receiver, and $t_{d,i}$ is the TOA of a signal which is directly coming from the ith transmitter to a receiver. Then, by substituting the coordinates of receiver, transmitters and target the TDOA measurement is represented as

$$t_{\mathrm{i}} = \frac{1}{c}\left(\left\|\mathbf{x}_i^t - \mathbf{x}\right\|_2 + \left\|\mathbf{x}_r - \mathbf{x}\right\|_2 - \left\|\mathbf{x}_i^t - \mathbf{x}_r\right\|_2\right) + \bar{n}_i, \forall i \in \{1,2,...,N\}, \tag{2}$$

where $\|\cdot\|_2$ denotes the two-dimensional Euclidean distance, c denotes propagation speed of light, and $\bar{n}_i$ is the Gaussian modelled measurement noise. Eliminating the term c, the range measurement of the signal coming from ith transmitter is obtained

$$\begin{aligned} r_i &= c \cdot t_i \\ &= \left(\left\|\mathbf{x}_i^t - \mathbf{x}\right\|_2 + \left\|\mathbf{x}_r - \mathbf{x}\right\|_2 - \left\|\mathbf{x}_i^t - \mathbf{x}_r\right\|_2\right) + n_i, \forall i \in \{1, 2, ..., N\}, \end{aligned} \tag{3}$$

where $n_i = c\bar{n}_i$ is Gaussian zero-mean distributed variable. Since the distance $\left\|\mathbf{x}_i^t - \mathbf{x}_r\right\|_2$ between each of the transmitters and receiver is previously known, we introduce a new variable to rearrange a range measurement as follows

$$\tilde{r}_i = r_i + \left\|\mathbf{x}_i^t - \mathbf{x}_r\right\|_2 \tag{4}$$

Therefore, the Equation (3) can be rewritten as

$$\tilde{r}_i = \left\|\mathbf{x}_r - \mathbf{x}\right\|_2 + \left\|\mathbf{x}_i^t - \mathbf{x}\right\|_2 + n_i, \forall i \in \{1, 2, ..., N\}. \tag{5}$$

Which is represented in the corresponding vector form

$$\tilde{\mathbf{r}} = \mathbf{d}(\mathbf{x}) + \mathbf{n}, \tag{6}$$

in which

$$\mathbf{d}(\mathbf{x}) = \begin{bmatrix} \left\|\mathbf{x}_1^t - \mathbf{x}\right\|_2 + \|\mathbf{x}_r - x\|_2 \\ \vdots \\ \left\|\mathbf{x}_N^t - \mathbf{x}\right\|_2 + \|\mathbf{x}_r - x\|_2 \end{bmatrix}, \tag{7}$$

and $\mathbf{n} = [n_1, n_2, \ldots, n_N]^T$ is the measurement noise vector, under assumption that each component of a vector $\mathbf{n}$ is independent and an identically distributed zero-mean Gaussian variable. The $\mathbf{C}$ denotes the appropriate matrix of covariance, whose elements can be calculated using the expectation operator $E[\cdot]$ as $\mathbf{C} = E\left[\mathbf{n}\mathbf{n}^T\right]$. Taking into consideration that the measurement noise is the zero-mean Gaussian variable, it can be simplified to $\mathbf{C} = \sigma^2\mathbf{I}_N$ [48].

This paper aims to accurately predict the passive target's $\mathbf{x}$ location, which is unknown, using TDOA parameters under the presence of noise. This is achieved solving the ML estimation problem, which is in general nonlinear and has multiple optima, which is outlined in the following section.

4. Formulation of Maximum Likelihood Estimator

To estimate the position of a passive target the Maximum Likelihood estimator is employed, since the probability density functions of measurement errors are known. A zero-mean Gaussian distribution is assumed for the TDOA measurement errors in the formulation of the localization issue [2]. Therefore, the likelihood function is obtained as

$$L(\tilde{\mathbf{r}}|\mathbf{x}) = \frac{1}{(2\pi)^{N/2}\det(\mathbf{C})^{1/2}} \exp\left(-\frac{1}{2}(\tilde{\mathbf{r}} - \mathbf{d}(\mathbf{x}))^T\mathbf{C}^{-1}(\tilde{\mathbf{r}} - \mathbf{d}(\mathbf{x}))\right), \tag{8}$$

Then, taking the logarithm of the Equation (8), log-likelihood function is yielded

$$\ln L(\tilde{\mathbf{r}}|\mathbf{x}) = k - \frac{1}{2\sigma^2}\left((\tilde{\mathbf{r}} - \mathbf{d}(\mathbf{x}))^T(\tilde{\mathbf{r}} - \mathbf{d}(\mathbf{x}))\right), \tag{9}$$

in which $k = \ln\left(\frac{1}{(2\pi)^{N/2}\det(\mathbf{C})^{1/2}}\right)$ is neglected as a constant independent of $\mathbf{x}$. As the solution to the minimization problem, we obtain the estimated target position $\hat{\mathbf{x}}$

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in R^2}{\operatorname{argmin}}\,(J_{ML}(\mathbf{x})), \tag{10}$$

with objective function of ML estimator

$$J_{ML}(\mathbf{x}) = (\tilde{\mathbf{r}} - \mathbf{d}(\mathbf{x}))^T((\tilde{\mathbf{r}} - \mathbf{d}(\mathbf{x}))). \tag{11}$$

The corresponding objective function $J_{ML}(\mathbf{x})$ of the formulated ML estimation problem is depicted on Figure 2.

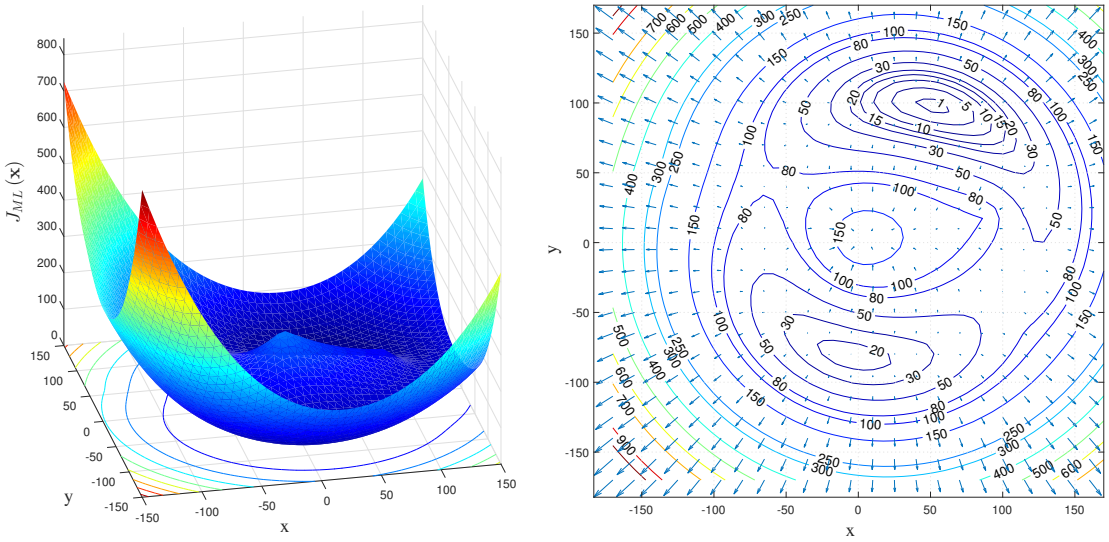

Figure 2. The plot of the considered ML objective function $J_{ML}(\mathbf{x})$.

Based on an examination of the topology of the function $J_{ML}(\mathbf{x})$ depicted in Figure 2, it is determined that $J_{ML}(\mathbf{x})$ is a multi-modal function with pronounced nonlinearities. Wherever the global minimum of a function occurs is where the target $\mathbf{x}$ should be located, due to the difficulty of the $J_{ML}(\mathbf{x})$ function, it is important to use advanced algorithms for finding the best possible solution globally.

5. Constrained Weighted Least-Squares

In this section, the CWLS algorithm is developed for the considered passive target localization problem. Therefore, the nonlinear TDOA equations can be transformed into a set of linear equations, by introducing the variable $R^r = \|x_r - x\|$, neglecting n_i^2 and squaring the Equation (5) giving the following expression

$$-2\left(\mathbf{x}_r - \mathbf{x}_i^t\right)^T \mathbf{x} - 2\tilde{r}_i R^r + \tilde{r}_i^2 + \|\mathbf{x}_r\|^2 - \left\|\mathbf{x}_i^t\right\|^2 = 2(\tilde{r}_i - R^r)n_i, \quad \forall i \in \{1, 2, ..., N\}, \tag{12}$$

If we write $\boldsymbol{\theta}$, as

$$\boldsymbol{\theta} = \begin{bmatrix} \mathbf{x} & R^r \end{bmatrix}^T, \tag{13}$$

then Equation (12) is converted to matrix form

$$\mathbf{b} - \mathbf{A}\boldsymbol{\theta} = \mathbf{m}, \tag{14}$$

where

$$\mathbf{A} = 2\begin{bmatrix} x_r - x_1^t & y_r - y_1^r & \tilde{r}_1 \\ x_r - x_2^t & y_r - y_2^r & \tilde{r}_2 \\ \vdots & \vdots & \vdots \\ x_r - x_N^t & y_r - y_N^r & \tilde{r}_N \end{bmatrix}, \tag{15}$$

$$\mathbf{b} = \begin{bmatrix} \tilde{r}_1^2 + x_r^2 + y_r^2 - \left(x_1^t\right)^2 - \left(y_1^t\right)^2 \\ \tilde{r}_2^2 + x_r^2 + y_r^2 - \left(x_2^t\right)^2 - \left(y_2^t\right)^2 \\ \vdots \\ \tilde{r}_N^2 + x_r^2 + y_r^2 - \left(x_N^t\right)^2 - \left(y_N^t\right)^2 \end{bmatrix}, \tag{16}$$

and

$$\mathbf{m} = 2\begin{bmatrix} (\tilde{r}_1 - R^r)n_1 \\ (\tilde{r}_2 - R^r)n_2 \\ \vdots \\ (\tilde{r}_N - R^r)n_N \end{bmatrix}. \tag{17}$$

The following constrained optimization problem can be used to estimate the coordinates of the target in the presence of measurement errors

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} (\mathbf{b} - \mathbf{A}\boldsymbol{\theta})^T \mathbf{W} (\mathbf{b} - \mathbf{A}\boldsymbol{\theta}) \tag{18}$$

$$s.t. \qquad \boldsymbol{\theta}^T \mathbf{S} \boldsymbol{\theta} = 0, \tag{19}$$

where $\mathbf{W} \in R^{N \times N}$ is a matrix of weights that is symetric, $\mathbf{S} = \text{diag}\{1, 1, -1\}$ is a diagonal and orthonormal matrix. It is possible to approximatively determine the elements of matrix of weights under conditions of sufficiently low measurement noise, as

$$\mathbf{W} = \left[E\left\{\mathbf{m}\mathbf{m}^T\right\}\right]^{-1} = \left(\mathbf{D}^T \mathbf{C} \mathbf{D}\right)^{-1}, \tag{20}$$

where

$$\mathbf{D} = \text{diag}\{2(\tilde{r}_1 - R^r), 2(\tilde{r}_2 - R^r), \ldots, 2(\tilde{r}_N - R^r)\}, \tag{21}$$

The optimization problem (18) is a quadratically constrained quadratic program (QCQP), which can be solved using the Lagrange multiplier method. Therefore, the Lagrangian function of the optimization problem (18) is given by

$$L(\boldsymbol{\theta}, \lambda) = (\mathbf{b} - \mathbf{A}\boldsymbol{\theta})^T W (\mathbf{b} - \mathbf{A}\boldsymbol{\theta}) + \lambda \boldsymbol{\theta}^T S \boldsymbol{\theta}, \tag{22}$$

where λ is the Lagrange multiplier. The minimum of Equation (22) is obtained by differentiating $L(\boldsymbol{\theta}, \lambda)$ with respect to $(\boldsymbol{\theta}, \lambda)$, as follows

$$\frac{\partial L(\boldsymbol{\theta}, \lambda)}{\partial \boldsymbol{\theta}^T} = \left(\mathbf{A}^T \mathbf{W} \mathbf{A} + \lambda \mathbf{S}\right)\boldsymbol{\theta} - \mathbf{A}^T \mathbf{W} \mathbf{b} = \mathbf{0}, \tag{23}$$

$$\frac{\partial L(\boldsymbol{\theta}, \lambda)}{\partial \lambda} = \boldsymbol{\theta}^T S \boldsymbol{\theta} = 0. \tag{24}$$

Solving the Equation (23) for $\boldsymbol{\theta}$, the following CWLS solution, denoted by $\widehat{\boldsymbol{\theta}}_{CWLS}$, is obtained

$$\hat{\boldsymbol{\theta}}_{CWLS} = \left(\mathbf{A}^T \mathbf{W} \mathbf{A} + \lambda \mathbf{S}\right)^{-1} \mathbf{A}^T \mathbf{W} \mathbf{b}. \tag{25}$$

where the value of λ is unknown, and can be determined as a root of the 4th-order polynomial, which is obtained by substituting Equation (25) into Equation (24) (see Appendix B), given as

$$\sum_{i=1}^{3} \frac{\alpha_i \beta_i}{(\zeta_i + \lambda)^2} = 0, \tag{26}$$

where $\{\alpha_i\}$, $\{\beta_i\}$ and $\{\zeta_i\}$, $\forall i = 1, 2, 3$, are given in Appendix B. Then, the procedure based on CWLS method is

Step 1: Initialize elements of matrix of weights as $\mathbf{W} = \mathbf{I}$.
Step 2: Using the standard root finding algorithm determine all real-valued roots of the Equation (26).
Step 3: Substitute obtained λ in previous step into Equation (25), and determine all sub estimates of $\hat{\boldsymbol{\theta}}_{CWLS}$ for which the $J(\boldsymbol{\theta})$is minimal.
Step 4: Using the Equation (20) construct the weighting matrix $\mathbf{W}$.
Step 5: Repeat steps 2–4 until $\hat{\boldsymbol{\theta}}_{CWLS}$ satisfies the stopping criterion.

6. Differential Evolution Algorithm and Improvements

6.1. Conventional DE

The DE method belongs to group of evolutionary optimization algorithms that are capable of obtaining a global optimum of difficult multimodal optimization problems [22]. Initialization, mutation, crossover, and selection are the four cornerstone operators of the DE algorithm's optimization procedure. These operators are discussed briefly in the following subsections.

6.1.1. Initialization

The DE algorithm begins its search for the global optimum with a population of N_P candidate solutions that are randomly positioned in the feasible search space. Each member of the population in the Gth generation is vector of components $\mathbf{x}_i^{(G)} = \left[x_{i,1}^{(G)}, x_{i,2}^{(G)}, \ldots x_{i,j}^{(G)}, \ldots, x_{i,n}^{(G)}\right]^T$, where $x_{i,j}^{(G)}$ denotes the jth component of the ith vector, of nth dimension. For the considered localization problem, the component of each individual in the population is initialized as follows

$$\left\{\mathbf{x}_i^{(G)} : x_{i,j}^{(G)} \in \left[x_{i,j}^L, x_{i,j}^{UI}\right]\right\}, \forall i \in \{1,2,...,NP\}, \forall j \in \{1,2,...,n\} \tag{27}$$

where $rand_j$ denotes a random number generated in the range $[0,1]$. Here, $x_{i,j}^L$ and $x_{i,j}^{UI}$ denote the maximum and minimum bounds of the feasible search space for each vector component.

6.1.2. Mutation

The mutation is a crucial operator in the DE algorithm, which promotes the exploration of the search space to locate the region of the global minimum and improves the quality of the solutions. In this regard, for each target vector $\mathbf{x}_i^{(G)}$, the DE algorithm creates the corresponding mutant vector $\mathbf{v}_i^{(G+1)} = \left[v_{i,1}^{(G+1)}, v_{i,2}^{(G+1)}, \ldots, v_{i,j}^{(G+1)}, \ldots, v_{i,n}^{(G+1)}\right]^T$. The vector $\mathbf{v}_i^{(G)}$ is usually created as a combination of weighted vector differences between randomly chosen vectors. Numerous mutation techniques have been presented in the literature with the intent of enhancing exploration, exploitation, or both. The most prominent mutation schemes include [73]:

DE/rand/1

$$\mathbf{v}_i^{(G+1)} = \mathbf{x}_{r_1}^{(G)} + F\left(\mathbf{x}_{r_2}^{(G)} - \mathbf{x}_{r_3}^{(G)}\right), \tag{28}$$

DE/rand/2

$$\mathbf{v}_i^{(G)} = \mathbf{x}_{r_1}^{(G)} + F\left(\mathbf{x}_{r_2}^{(G)} - \mathbf{x}_{r_3}^{(G)}\right) + F\left(\mathbf{x}_{r_4}^{(G)} - \mathbf{x}_{r_5}^{(G)}\right), \tag{29}$$

DE/best/1

$$\mathbf{v}_i^{(G+1)} = \mathbf{x}_{best}^{(G)} + F\left(\mathbf{x}_{r_1}^{(G)} - \mathbf{x}_{r_2}^{(G)}\right), \tag{30}$$

DE/best/2

$$\mathbf{v}_i^{(G+1)} = \mathbf{x}_{best}^{(G)} + F\left(\mathbf{x}_{r1}^{(G)} - \mathbf{x}_{r2}^{(G)}\right) + F\left(\mathbf{x}_{r3}^{(G)} - \mathbf{x}_{r4}^{(G)}\right), \tag{31}$$

DE/current-to-best/1

$$\mathbf{v}_i^{(G+1)} = \mathbf{x}_i^{(G)} + F\left(\mathbf{x}_{best}^{(G)} - \mathbf{x}_i^{(G)}\right) + F\left(\mathbf{x}_{r_1}^{(G)} - \mathbf{x}_{r_2}^{(G)}\right), \tag{32}$$

where r_1, r_2, r_3, r_4 and r_5 are mutually distinct integers randomly generated from the set $\{1,2,\ldots,NP\}\backslash\{i\}$, the $\mathbf{x}_{best}^{(G)}$ denotes the individual with the best objective function value in the entire population at the current generation G and F is the scale factor that controls the scaling ratio of the difference vector. It has been shown that besides choosing the appropriate mutation scheme the value of scale factor plays a significant role in the optimization outcome [28].

6.1.3. Crossover

The crossover operator is used in the DE algorithm to promote population diversity. In this operator, the algorithm generates a trial vector component $u_{i,j}^{(G+1)}$ by combining the elements of the target vector $x_{i,j}^{(G)}$ and the matching mutant vector $v_{i,j}^{(G)}$, created in the previous step, as follows

$$u_{i,j}^{(G+1)} = \begin{cases} v_{i,j}^{(G+1)} & \text{if} \quad rand_{i,j} \leqslant CR \vee j = j_{rand} \\ x_{i,j}^{(G)} & otherwise \end{cases} \tag{33}$$

where $rand_{i,j}$ is a uniformly generated random number from the range $[0, 1]$, and CR is a crossover rate, a parameter controlling the number of components from the mutant vector is injected into the trial vector, and $j_{rand} \in \{1, 2, \ldots, n\}$ is an integer selected randomly. It has been demonstrated that the number of vector components chosen for crossover varies depending on the stage of the optimization process [74]. Adaptively changing the crossover rate during optimization is, thus, essential.

6.1.4. Selection

Finally, in the selection operator, the DE algorithm determines which vector poses the qualities required to solve the considered optimization problem and is chosen to form a population for the next generation. As the measure of the quality of the solution, the objective function value at the trial vector $f\left(\mathbf{u}_i^{(G+1)}\right)$ and matching target vector $f\left(\mathbf{x}_i^{(G)}\right)$ is employed. Therefore, the individual with the lower objective function value is retained for the next generation according to

$$\mathbf{x}_i^{(G+1)} = \begin{cases} \mathbf{u}_i^{(G+1)} & \text{if} \quad f\left(\mathbf{u}_i^{(G+1)}\right) \leqslant f\left(\mathbf{x}_i^{(G)}\right) \\ \mathbf{x}_i^{(G)} & otherwise \end{cases}, \tag{34}$$

where $\mathbf{x}_i^{(G+1)}$ denotes the target vector selected to the next generation.

Repeated iterations of mutation, selection, and crossover are carried out until the maximum number of generations $G_{\max}$ specified by the termination criteria has been reached.

6.2. Improvement of the Differential Evolution Algorithm

SHADE is a powerful variant of the DE algorithm that emerged as an improvement on the well-known JADE algorithm [34]. Therefore, in this study, we suggest using the SHADE method to provide specific control parameters for each population member. In this regard, for each individual the scale factor $F_i^{(G)}$ and crossover rate $CR_i^{(G)}$ are produced according to the Cauchy and normal distributions, respectively, as follows

$$F_i^{(G)} = randc_i\left(\mu_F^{(G)}, 0.1\right), \tag{35}$$

$$CR_i^{(G)} = randn_i\left(\mu_{CR}^{(G)}, 0.1\right), \tag{36}$$

where $randc_i\left(\mu_F^{(G)}, 0.1\right)$ represents a random number drawn according with the Cauchy distribution whose median is denoted as $\mu_F^{(G)}$ and the scale parameter is fixed to 0.1, and $rand_i\left(\mu_F^{(G)}, 0.1\right)$ is a random number drawn from the normal distribution with mean value denoted as $\mu_{CR}^{(G)}$ and standard deviation 0.1.

Both $\mu_F^{(0)}$ and $\mu_{CR}^{(0)}$ are initialized to 0.5. Since the domains of the scale factor and crossover rate in DE algorithm are usually defined as $F \in [0, 1]$ and $CR \in [0, 1]$ [73], the in the cases when the values generated according to Equations (35) and (36) are out of bounds the following truncation rules are applied: When scale factor is $F_i^{(G)} > 1$, we truncate $F_i^{(G)}$

to 1,otherwise in case $F_i^{(G)} < 0$, $F_i^{(G)}$ is regenerated using Equation (35) and domain is checked again. For the crossover rate check if $CR_i^{(G)} > 1$, and if satisfied truncate to 1, in the case $CR_i^{(G)} < 0$ truncate $CR_i^{(G)}$ to 0.

The SHADE algorithm has introduced a set where the successful previous versions of generated parameters are stored in order to influence the generation of new values in the next generation. Therefore, the $F_i^{(G)}$ and $CR_i^{(G)}$ values which have proven to be successful are stored in the sets S_F and S_{CR}, respectively in each iteration. Then, in the next generation the $\mu_F^{(G)}$ and $\mu_{CR}^{(G)}$parameters are generated according to

$$\mu_F^{(G+1)} = \begin{cases} (1-c)\mu_F^{(G)} + c \cdot mean_{WL}(S_F) & if \quad S_F \neq \varnothing \\ (1-c)\mu_F^{(G)} + c \cdot rand(0,1) & otherwise \end{cases}, \tag{37}$$

$$\mu_{CR}^{(G+1)} = \begin{cases} (1-c)\mu_{CR}^{(G)} + c \cdot mean_{WA}(S_{CR}) & if \quad S_{CR} \neq \varnothing \\ (1-c)\mu_{CR}^{(G)} + c \cdot rand(0,1) & otherwise \end{cases}, \tag{38}$$

where c denotes learning rate, a constant that controls the amount of previous experience taken for new parameter generation, which is set to $c = 0.1$ [34], while $mean_{WL}(\cdot)$ and $mean_{WA}(\cdot)$ are the weighted Lehmer mean and weighted arithmetic mean, respectively, which are determined as

$$mean_{WL}(S_F) = \frac{\sum_{k=1}^{L} w_k \cdot S_{F,k}^2}{\sum_{k=1}^{L} w_k \cdot S_{F,k}}, \tag{39}$$

$$mean_{WA}(S_{CR}) = \frac{\sum_{k=1}^{L} w_k \cdot S_{CR,k}^2}{\sum_{k=1}^{L} w_k \cdot S_{CR,k}}, \tag{40}$$

where the size of the sets S_F and S_{CR} is $L = |S_F| = |S_{CR}|$, and w_k coefficients are determined as

$$w_k = \frac{\left| f\left(\mathbf{u}_i^{(G)}\right) - f\left(\mathbf{x}_i^{(G)}\right) \right|}{\sum_{k=1}^{L} \left| f\left(\mathbf{u}_i^{(G)}\right) - f\left(\mathbf{x}_i^{(G)}\right) \right|}. \tag{41}$$

6.3. Mutation Scheme DE/current-to-pbest/1

The popular JADE method includes a mutation operator, DE/current-to-pbest/1, that narrows the search to the top percentile of population members and quickens convergence [33]. This operator is an extended with improved parameter adaption scheme introduced in the SHADE algorithm. This demonstrates the significance of the mutation strategy to the performance of optimization. The following may be stated about the accompanying mutation vector based on the suggested DE/current-to-pbest/1 mutation operator:

$$v_i^{(G)} = \mathbf{x}_i^{(G)} + F_i^{(G)}\left(\mathbf{x}_{pbest}^{(G)} - \mathbf{x}_i^{(G)}\right) + F_i^{(G)}\left(\mathbf{x}_{r1}^{(G)} - \mathbf{x}_{r2}^{(G)}\right), \tag{42}$$

where $\mathbf{x}_{pbest}^{(G)}$ is randomly selected from the top ($p \times 100\% \times NP$) individuals in the current population with $p \in (0, 1]$ and $F_i^{(G)}$ is generated to Equation (35).

Analysing the Eq., we observe that the search ability of the algorithm is enhanced, particularly in the latter stages of the evolution process, when the mutation vector explores the neighborhood of each $\mathbf{x}_i^{(G)}$ in the direction of the $\mathbf{x}_{pbest}^{(G)}$solution.

7. Butterfly Optimization Algorithm and Improvements

7.1. Original BOA Algorithm

BOA is a novel metaheuristic optimization technique that was inspired by butterfly foraging and butterfly mating [21]. The BOA algorithm's algorithmic representation of the biological system is based on three hypotheses:

1. All butterflies attract each-other by emitting a scent.
2. Their flight is random or aimed to the best butterfly emitting more intense scent.
3. Butterfly stimulus strength is determined by the objective function value at the considered butterfly.

There are three stages to BOA algorithm's optimization process: startup, iteration, and final optimization. In each execution of BOA, the initialization phase is conducted first, followed by iteration, and the algorithm is finished after the optimal solution has been identified in the final optimization phase. The objective function and solution space are specified during the phase of initialization. In this phase, the BOA parameters are initialized, and the method then generates a random initial butterfly population within the upper and lower bounds calculating and storing their fragrance and fitness values. The iteration phase of BOA follows the initialization step and with it is when the optimum solution is found via a series of iterations. During each iteration, the value of the objective function is calculated for each butterfly, and then the butterflies use that value to influence the amount of scent φ_i they produce in their current positions

$$\varphi_i = cI^a, \tag{43}$$

in which a denotes the power exponent chosen as $a \in [0, 1]$ and c is the sensory fragrance in the range $[0, 1]$, and I is the stimulus intensity. The change in fragrance absorption is governed by the parameter a value. Given two limit scenarios, i.e., when $a = 1$, there is no absorption of scent, i.e., the same amount of fragrance is recognized by all butterflies. Consequently, a fragrant butterfly can be discovered anywhere within the region, and a global solution can be reached. In contrast, when $a = 0$, the scent cannot be sensed by other butterflies, thus impacting the algorithm's optimization behaviour. The algorithm's convergence rate can be controlled by the value of the parameter c, which is usually chosen based on the properties of the objective function to be optimized. The values of a and c, have a considerable effect on the algorithm's convergence rate and optimization performance. The value of the stimulus intensity I depends on the objective function value at the considered butterfly, as follows

$$I = f\left(\mathbf{x}_i^{(G)}\right), \tag{44}$$

where $f\left(\mathbf{x}_i^{(G)}\right)$ is the objective function value of ith butterfly at Gth generation. In the iteration phase, the BOA evaluates two steps, i.e., global and local search. The butterfly with the highest obj. function value, indicated by $\mathbf{g}^*$, is moved in the step of global search using the formula

$$\mathbf{x}_i^{(G+1)} = \mathbf{x}_i^{(G)} + \left(r^2 \times \mathbf{g}^* - \mathbf{x}_i^{(G)}\right) \times \varphi_i, \tag{45}$$

where $r \in [0, 1]$ is the randomly selected number. The next step, which deals with the local search, assumes movement of the butterfly in the vicinity of other neighbouring butterflies, as follows

$$\mathbf{x}_i^{(G+1)} = \mathbf{x}_i^{(G)} + \left(r^2 \times \mathbf{x}_j^{(G)} - \mathbf{x}_k^{(G)}\right) \times \varphi_i, \tag{46}$$

where $\mathbf{x}_j^{(G)}$ and $\mathbf{x}_k^{(G)}$ denote the appropriate butterflies taken from the solution space.

In each iteration, switch probability p is used to choose between expressions in Equations (46) and (45) according to the expression

$$\mathbf{x}_i^{(G+1)} = \begin{cases} \mathbf{x}_i^{(G)} + \left(r^2 \times \mathbf{g}^* - \mathbf{x}_i^{(G)}\right) \times \varphi_i, & \text{for } r_p < p \\ \mathbf{x}_i^{(G)} + \left(r^2 \times \mathbf{x}_j^{(G)} - \mathbf{x}_k^{(G)}\right) \times \varphi_i, & \text{otherwise} \end{cases} \tag{47}$$

where $r_p \in [0, 1]$ is generated randomly at each generation. Iterations continue until a certain condition is reached, and an optimal solution is presented.

7.2. Chaos Enhanced BOA Algorithm

Literature shows that the traditional BOA method has high local search ability, but it often prematurely converges to local optima, which degrades optimization efficiency, particularly when dealing with the complicated and multi-modal optimization [37]. Therefore there is a need to modify original BOA and improve optimization performance. This is why some academics have recommended incorporating chaos into BOA in order to boost overall optimization efficiency and avoid the issues in performance [36]. In nonlinear dynamic systems, we can observe a bounded dynamic behavior called chaos, which is described with chaos maps, which generate a limited series of random integers in response to a given starting condition.

By analyzing Equation (43), we see that the butterfly's ability to detect the scents emitted off by other butterflies is one of the most crucial aspects in BOA, since it directs their flight throughout the search phase. Thus, when solving complex optimization problems, parameter c should not be kept constant during the optimization process. Searching with a small c may cause premature convergence, while a large constant value of c early in the search phase increases the risk of missing the region of global optimum, which affects optimization performance. Therefore, we propose an adaptive mechanism to change value of sensory fragrance c during the optimization process which is based on the use of chaos maps, which can be written in the following form

$$c^{(G)} = e^{-\frac{G}{G\max}} \cdot x_c^{(G)} \tag{48}$$

where $x_c^{(G)}$ represents the chaotic motion produced by chosen chaos map in the Gth generation. The Equation (48) describes a parameter which has the chaotic motion but is upper bounded by the fator $e^{-\frac{G}{G\max}}$.

Numerous distinct chaos maps may be found in literature. The five most popular chaotic maps have been utilised in the current study to change the value of parameter c and to identify which one is most appropriate for the task at hand.

Piecewise map: The Piecewise map is described as

$$x_c^{(G+1)} = \begin{cases} \frac{x_c^{(G)}}{P}, & 0 \le x_c^{(G)} < P \\ \frac{x_c^{(G)} - P}{0.5 - P}, & P \le x_c^{(G)} \le 0.5 \\ \frac{1 - P - x_c^{(G)}}{0.5 - P}, & 0.5 \le x_c^{(G)} \le 1 - P \\ \frac{1 - x_c^{(G)}}{P}, & 1 - P \le x_c^{(G)} \le 1 \end{cases} \tag{49}$$

$$G = 1, 2, \ldots, G_{max}. \tag{50}$$

Iterative map: The expression for the Iterative map can be written as

$$x_c^{(G+1)} = \sin\left(\frac{P\pi}{x_c^{(G)}}\right), \quad G = 1, 2, \ldots, G_{max}. \tag{51}$$

where P is the parameters choosen as $P \in (0, 1)$.

Logistic map: The Logistic map is written as

$$x_c^{(G+1)} = P x_c^{(G)} \left(1 - x_c^{(G)}\right), \quad G = 1, 2, \ldots, G_{max}. \tag{52}$$

Sine map: The expression for Sine map can be written as

$$x_c^{(G+1)} = \frac{a}{4} \sin\left(\pi x_c^{(G)}\right), \quad G = 1, 2, \ldots, G_{max}. \tag{53}$$

where the chaotic motion is achieved for $a \in (0, 4)$.

Tent map: The Tent map is written acording to the equation

$$x_c^{(G+1)} = \begin{cases} \frac{x_c^{(G)}}{0.7}, & x_c^{(G)} < 0.7 \\ \frac{10}{3}\left(1 - x_c^{(G)}\right), & x_c^{(G)} \geq 0.7 \end{cases} \quad G = 1, 2, \ldots, G_{max}. \tag{54}$$

In this respect, Figure 3 illustrates the changes of the employed chaos maps given in Equations (49)–(54), with the increase in generations, and the plot chaos based sensory fragrance $c^{(G)}$ from Equation (48) with the employment of each chaos map expression for setting the value of $x_c^{(G)}$.

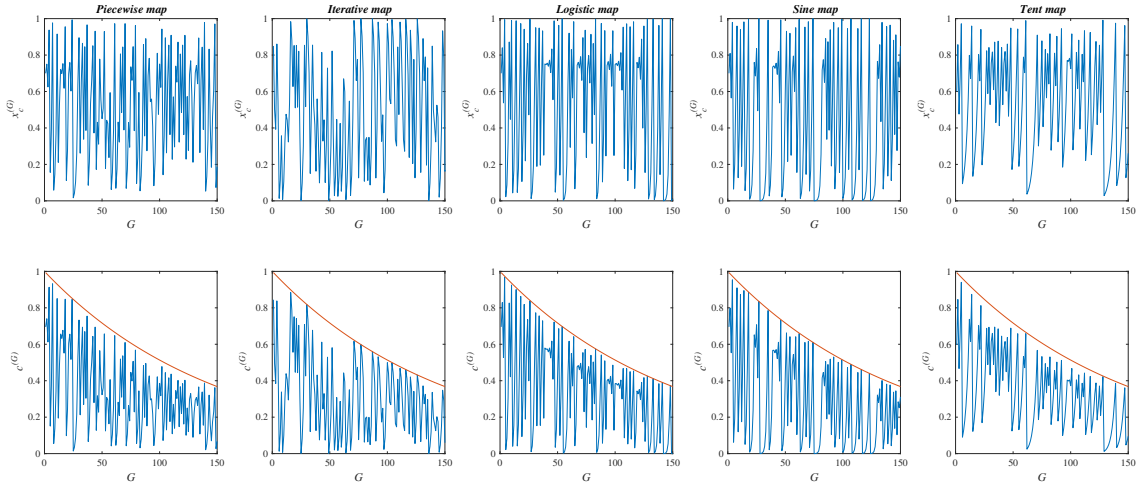

Figure 3. Plot of the employed chaos maps and corresponding chaos based sensory fragrance $c^{(G)}$ as a function of generation number.

From the Figure 3, we find that the considered chaos maps generate a number sequence that is both ergodic and irregular, but each considered map has unique properties that are visible on the graph. Moreover, we observe that the exponential factor $e^{-G/G_{\max}}$ serves as the top boundary for the chaos-based sensory scent $c^{(G)}$. As a result, at the beginning of the optimization process, the considered chaotic sequences have higher values, and as more generations are added, the value of the resulting sequence of numbers decreases. Higher values at the start of the optimization process are beneficial for encouraging solution space exploration. The lower values at the end of the optimization process, however, are important for the improvement of local search in the later stage of the optimization process.

In order to determine which of the considered chaos maps is best suited to increase the performance of BOA algorithm, in this paper the statistical analysis is applied, which is explained in detail in Section 10.1.1. Based on the results of the Friedman's test, presented in Table 1 we observe that the introduction of chaos maps into the BOA algorithm leads to an increase in optimization performance compared to conventional BOA. Furthermore, it is concluded that the BOA algorithm enhanced with sine map has achieved the best

performance (best rank). Therefore the chaos-based sensory fragrance $c^{(G)}$ can be obtained using the following equations:

$$x_c^{(G+1)} = \frac{a}{4}\sin\left(\pi x_c^{(G)}\right) \tag{55}$$

$$c^{(G)} = e^{-\frac{G}{G\max}} \cdot x_c^{(G)}, \quad G = 1, 2, \ldots, G_{max} \tag{56}$$

where a is the parameter which determines the chaotic motion.

Table 1. Summary of the statistical findings of the Friedman test comparing the performance of the five chaos-improved BOA algorithms and the conventional BOA algorithm.

Algorithm	10*D*	30*D*	50*D*	100*D*	Mean Ranking	Rank
BOA-c-Sine	2.27	2.5	2.65	2.5	2.47	1
BOA-c-Logistic	2.76	2.72	2.62	2.9	2.75	2
BOA-c-Picewise	3.13	3.4	3.05	2.8	3.09	3
BOA-c-Iterative,	3.53	3.52	3.42	3.93	3.6	4
BOA-c-Tent	4.03	3.41	3.75	3.5	6.67	5
Original BOA	5.26	5.45	5.51	5.37	5.4	6
Friedman p-value	6×10^{-10}	8.32×10^{-11}	2.96×10^{-12}	4.42×10^{-12}		

8. Improved Chaos-Driven Hybrid Differential Evolution and Butterfly Optimization Algorithm

In Section 4, it has been demonstrated that the considered passive target localization problem is a non-convex complex problem, which cannot be solved using conventional algorithms. According to a review of the literature, the DE algorithm often performs well in a global search but poorly in a local search [32]. However, applying these tactics throughout the whole evolution process might result in premature convergence. The authors have suggested utilising DE/best/1, DE/best/2, and more recently DE/current-to-p-best/1 mutation strategies to enhance local search [33]. Conversely, BOA algorithm shows excellent local search ability, however, it often prematurely converges to local optima of complex problems [37]. Therefore, the main idea of this paper is to combine good local search abilities of BOA with excellent global search of DE algorithm. In the proposed ICDEBOA algorithm, this is achieved by introducing the BOA particle movement equations into the mutation strategies of DE algorithm. Furthermore, the improvements of DE and BOA algorithms, outlined in the previous subsections, are introduced into the ICDEBOA algorithm.Therefore, in order to improve the exploration abilities of the proposed ICDEBOA algorithm, in this paper we propose the uses of the following two mutation strategies for exploration:

- DE/rand/1 mutation strategy with adaptive scale factor $F_i^{(G)}$, which is calculated according to the Equation (35). Therefore the expression for calculating the mutation vector is

$$\mathbf{v}_i^{(G+1)} = \mathbf{x}_{r_1}^{(G)} + F_i^{(G)}\left(\mathbf{x}_{r_2}^{(G)} - \mathbf{x}_{r_3}^{(G)}\right). \tag{57}$$

- Novel DE/rand-global-BOA/2 mutation strategy, which is produced by introducing the parts of BOA global search step into the DE/rand/2 mutation strategy. Therefore, to achieve this the term $F\left(\mathbf{x}_{r_4}^{(G)} - \mathbf{x}_{r_5}^{(G)}\right)$ in Equation (29) is replaced with the term $\left(r^2 \times \mathbf{g}^* - \mathbf{x}_i^{(G)}\right) \times \varphi_i$ from Equation (45). Furthermore, the adaptive scale factor $F_i^{(G)}$ given in Equation (35) is introduced instead the conventional scale factor F. Additionally, the sine chaos map-based sensory fragrance $c^{(G)}$ is employed in determining the term φ_i. Therefore, the DE/rand-global-BOA/2 mutation strategy has the form

$$\mathbf{v}_i^{(G)} = \mathbf{x}_{r_1}^{(G)} + F_i^{(G)}\left(\mathbf{x}_{r_2}^{(G)} - \mathbf{x}_{r_3}^{(G)}\right) + \left(r^2 \times \mathbf{g}^* - \mathbf{x}_i^{(G)}\right) \times \varphi_i. \tag{58}$$

Next, the modification is performed in order to perform local search and improve the solution accuracy. The use of the following two mutation operators is proposed

- The usage of DE/best/1 mutation strategy with adaptive scale factor $F_i^{(G)}$ from Equation (35) is proposed, which can be expressed as

$$\mathbf{v}_i^{(G+1)} = \mathbf{x}_{best}^{(G)} + F_i^{(G)} \left(\mathbf{x}_{r_1}^{(G)} - \mathbf{x}_{r_2}^{(G)} \right). \tag{59}$$

- A new DE/best-local-BOA/2, produced by hybridizing the mutation operator DE/best/2 form DE algorithm with the local search step of BOA algorithm given in Equation (46). In this regard, the term $F\left(\mathbf{x}_{r3}^{(G)} - \mathbf{x}_{r4}^{(G)} \right)$ from the Equation (31) is replaced with the term $\left(r^2 \times \mathbf{x}_j^{(G)} - \mathbf{x}_k^{(G)} \right) \times \varphi_i$ from Equation (46). Furthermore, the adaptive scale factor $F_i^{(G)}$ from Equation (35) and sine chaos map-based sensory fragrance $c^{(G)}$ from Equation (48) have been introduced, providing the following expression

$$\mathbf{v}_i^{(G+1)} = \mathbf{x}_{best}^{(G)} + F_i^{(G)} \left(\mathbf{x}_{r1}^{(G)} - \mathbf{x}_{r2}^{(G)} \right) + \left(r^2 \times \mathbf{x}_j^{(G)} - \mathbf{x}_k^{(G)} \right) \times \varphi_i. \tag{60}$$

A lot of research has been proposed, which deals with the introduction of new mutation operators into the DE algorithm. In JADE, authors proposed using the DE/current-to-p-best/1 mutation scheme to improve optimization performance [33]. Moreover, authors proposed using multiple mutation strategies [75]. In this way, the suggested technique presents a novel triangular mutation rule based on the convex combination vector of the triplet created by the three randomly selected vectors and the difference vector between the best and poorest individuals from the three randomly picked vectors. Combining the mutation rule with the standard mutation approach via a non-linear decreasing probability rule, authors proposed using a reflection-based mutation operation inspired by the Nelder–Mead technique and two additional fundamental mutation strategies to preserve population diversity [76]. In order to choose mutation strategies, their probabilities of success for each individual are employed using a roulette wheel selection method. Recently, the authors proposed hybridization between DE and FA algorithms with multiple mutation strategies, where the FA is introduced into the mutation scheme of DE and then randomly applied during the evolution process [16].

In this paper, we proposed using the roulette wheel selection, in order to perform the choice of the mutation operator, which is appropriate to the current optimization stage. Initially, each of the mutation operators are assigned random success rates s_r in the range $[0, 1]$ such that $\sum s_{r,k} = 1, \forall k = 1, 2, 3, 4$. During the evolution process, the kth mutation operator, which is used to produce mutant vector $\mathbf{v}_i^{(G)}$ of the ith individual, is assigned value 1 if the produced mutant vector lead to the improvement in the optimization process, e.g., the success rate is calculated as follows

$$s_{r,k}^{(G)} = \begin{cases} 1, & \text{if} f\left(\mathbf{v}_i^{(G)} \right) < f\left(\mathbf{x}_i^{(G)} \right) \\ 0, & \text{otherwise} \end{cases}. \tag{61}$$

Then, the total probability of choosing the appropriate mutation scheme is calculated as follows

$$p_k^{(G)} = \sum_{i=1}^{N_P} \frac{s_{r,k}^{(G)}}{N_{P,k}}, \tag{62}$$

where $N_{P,k}$ denotes the total number of individuals that are assigned the kth mutation operator in the current generation. Therefore, if the value of p_k of kth mutation operator is higher, the probability of choosing that mutation scheme increases.

In this way, the pseudocode of the proposed ICDEBOA algorithm is presented in Algorithm 1, for the considered passive target localization problem.

Algorithm 1 ICDEBOA algorithm pseudo-code

Initialize parameters N_P, $MaxIter$
Generation of initial population of N_P individual
Generate sine map $x_c^{(G)}$ for $G = 1, 2, \ldots, MaxIter$
Initialize values of p_k for $k = 1, 2, 3, 4$
while $G < MaxIter$ **do**
  Determine the probability of choosing the appropriate mutation scheme $p_k^{(G)}$
  Update sensory fragrance $c^{(G)}$ using the Equation (55)
  for $i = 1, 2, \ldots, N_P$ **do**
    Generate three random integers $r_1, r_2, r_3 \in \{1, \ldots, N_P\}$ such that $r_1 \neq r_2 \neq r_3 \neq i$
    Use roulette wheel to select k
    switch k **do**
      case k = 1
        Apply mutation operator in Equation (57)
      case k = 2
        Apply hybrid mutation operator DE/rand-global-BOA/2 in Equation (58)
      case k = 1
        Apply mutation operator in Equation (59)
      case k = 1
        Apply hybrid mutation operator DE/best-local-BOA/2 in Equation (60)
    end switch
    Calculate the success rate $s_{r,k}^{(G)}$ using Equation (61)
    Perform crossover
    Generate random number $j_{rand} \in \{1, 2, \ldots, n\}$
    for $j = 1, 2, \ldots, n$ **do**

$$u_{i,j}^{(G+1)} = \begin{cases} v_{i,j}^{(G+1)} & \text{if} \quad rand_{i,j} \leqslant CR \vee j = j_{rand} \\ x_{i,j}^{(G)} & otherwise \end{cases}$$

    end for
    Perform selection

$$\mathbf{x}_i^{(G+1)} = \begin{cases} \mathbf{u}_i^{(G+1)} & \text{if} \quad f\left(\mathbf{u}_i^{(G+1)}\right) \leqslant f\left(\mathbf{x}_i^{(G)}\right) \\ \mathbf{x}_i^{(G)} & otherwise \end{cases}$$

  end for
end while

9. Cramer-Rao Lower Bound

In this section, the overview of the estimation error measure and the derivation of the appropriate CRLB is provided, which is used to assess the performance of considered estimation methods. The CRLB, which is given by the inverse of the Fisher information matrix (FIM), denoted as $\mathrm{F}(\mathbf{x})$, sets the lower limit for the variance (or covariance matrix) of any unbiased estimator [48]. The FIM $\mathrm{F}(\mathbf{x})$ represents a measure of the knowledge contained in the observed data about the unknown parameter vector, and it may be computed as follows

$$\begin{aligned} \mathrm{F}(\mathbf{x}) &= \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} = \mathrm{E}\left[\left(\frac{\partial \ln(f(\tilde{\mathbf{r}}|\mathbf{x}))}{\partial \mathbf{x}} \right) \left(\frac{\partial \ln(f(\tilde{\mathbf{r}}|\mathbf{x}))}{\partial \mathbf{x}} \right)^T \right] \\ &= -\mathrm{E}\left[\frac{\partial^2 \ln(f(\tilde{\mathbf{r}}|\mathbf{x}))}{\partial \mathbf{x} \partial \mathbf{x}^T} \right]. \end{aligned} \tag{63}$$

with the corresponding elements of the matrix determined according to the expressions

$$F_{11} = \frac{1}{\sigma^2} \sum_{i=1}^{N} \left(\frac{x - x_r}{\|\mathbf{x} - \mathbf{x}_r\|} + \frac{x - x_i^t}{\|\mathbf{x} - \mathbf{x}_i^t\|_2} \right)^2, \tag{64}$$

$$F_{12} = F_{21} = \frac{1}{\sigma^2}\sum_{i=1}^{N}\left(\frac{x - x_r}{\|\mathbf{x} - \mathbf{x}_r\|} + \frac{x - x_i^t}{\|\mathbf{x} - \mathbf{x}_i^t\|_2}\right) \times \left(\frac{y - y_r}{\|\mathbf{x} - \mathbf{x}_r\|} + \frac{y - y_i^t}{\|\mathbf{x} - \mathbf{x}_i^t\|_2}\right), \tag{65}$$

$$F_{22} = \frac{1}{\sigma^2}\sum_{i=1}^{N}\left(\frac{y - y_r}{\|\mathbf{x} - \mathbf{x}_r\|} + \frac{y - y_i^t}{\|\mathbf{x} - \mathbf{x}_i^t\|_2}\right)^2. \tag{66}$$

The process of deriving the Equations (64)–(66) is described in detail in Appendix A. Therefore, the CRLB inequality bound of the unbiased estimator $\hat{\mathbf{x}}$ of parameter $\mathbf{x}$ is represented with the relationship between covariance $\mathrm{cov}(\hat{\mathbf{x}}) = E[\hat{\mathbf{x}}]$and the inverse of the FIM, as follows

$$E\left[(\hat{\mathbf{x}}-\mathbf{x})\,(\hat{\mathbf{x}}-\mathbf{x})^T\right] \geq \mathrm{Tr}\left\{\mathbf{F}(\mathbf{x})^{-1}\right\} = \mathrm{CRLB}(\mathbf{x}), \tag{67}$$

where $\hat{\mathbf{x}}$ denotes the estimated value of $\mathbf{x}$.

10. Experimental Analysis

Experiments were carried out to evaluate the accuracy of the proposed ICDEBOA method on the localization problem and to compare the optimization performance of the proposed algorithm with the performance of state-of-the-art algorithms in the literature on a set of CEC2014 benchmark problems. The results of these experiments are presented in this section. The results collected may be broken down into two categories: (1) an assessment of various EAs' performances on the CEC2014 benchmark, and (2) an examination of localization's effectiveness.

10.1. Statistical Comparisons on CEC2014 Problems

According to [77], the CEC2014 benchmark problems include 30 optimization problems with one objective that are defined on real numbers with the search space dimensions of $D = 10, 30, 50$, and 100. Based on objective functions characteristics, the CEC2014 benchmark problems can be put into four groups:

- Unimodal functions (1–3),
- Simple multimodal functions (4–16),
- Hybrid objective functions(17–22),
- Composition functions (23–30).

In order to compare how well the algorithms performs, the solution error measure $f(\hat{\mathbf{x}}) - f(\mathbf{x}^*)$ was created. This is defined as the difference between the solution $\hat{\mathbf{x}}$, which has the lowest objective function value and was obtained in a single simulation run and solution $\mathbf{x}^*$, which is previously known for the considered objective function. The termination criterion for the subsequent numerical experiments was set at $10,000D$, 100 individual in each algorithm's population is employed, and each algorithm was run 51 times for each of the test functions. Two non-parametric statistical hypothesis tests, such as the Wilcoxon signed-rank test and the Friedman test, have been used to analyze and evaluate the quality of the solutions from a statistical perspective.

The Wilcoxon signed-rank is used to compare the optimization performance of each of the algorithms under consideration. This test provides information whether the first algorithm performs statistically better than the second algorithm, and was performed with $\alpha = 0.05$ significance. Therefore, "there is no difference between the mean results of the two samples" [78] represents the null hypothesis. The hypothesis is rejected when $p < \alpha$, and then "there is a difference in the mean results of the two samples" represents inverse hypothesis.

Here, R^+ denotes accomulated ranks where first algorithm had a superior performance to second one taken to test , and R^- denotes the accomulated ranks for the case when the second tested algorithm achieved better performance.

The Friedman test was employed to determine differences in performance between each algorithm taken for comaprison form the statistical point. This is achieved by obtaining the ranks of each algorithm over each CEC2014 problem on all considered D. Here, the best performing algorithm is the one with the lowest rank. This test is taken with a significance level $\alpha = 0.05$, and the null hypothesis "there is no difference among the performance of all algorithms" is rejected for $p \leq \alpha$. Conversely, the alternative is "there is a difference among the performance of all algorithms" [78].

In the obtained results, the signs $+$, $\approx$, $-$ (better, equal, and worse) denote: $(+)$ that the first algorithm had significantly better optimization performance than the second, $(-)$ that the first algorithm performed significantly worse than the second, and $(\approx)$ that there is no significant difference in optimization performance between the two algorithms.

10.1.1. Analyzing the Effectiveness of the Chaos Maps on BOA Performance

To evaluate the effectiveness of the chaos theory-based sensory fragrance, as proposed in Section 7.2 in Equation (55), on the optimization performance of the BOA algorithm we have performed the following statistical analysis. The performance of the BOA algorithm is evaluated with five chaos maps, as shown in Figure 3, and compared to the conventional BOA algorithm using the Friedman test. The following variations of BOA algorithm were considered, which have the same operators as conventional BOA, except that the sensory fragrance is set according to Equation (48) with the following chaos map:

1. BOA-c-Picewise, using the Picewise map given in Equation (49),
2. BOA-c-Iterative, using the Iterative map given in Equation (51),
3. BOA-c-Logistic, using the Logistic map given in Equation (52),
4. BOA-c-Sine, using the Sine map given in Equation (53),
5. BOA-c-Tent, using the Tent map given in Equation (54).

Table 1 shows a summary of the Friedman test's statistical results of the performance comparison between the five chaos-improved BOA algorithms and the conventional BOA.

According to the results in Table 1, we conclude that the chaos-enhanced BOA algorithms have significantly better performance than the conventional BOA. Moreover, the results demonstrate that the best performance among the tested algorithms is achieved with the BOA-c-Sine algorithm, which demonstrates the effectiveness of the proposed modification. Therefore, we conclude that the improved chaos-based sensory fragrance given in Equation (55) will be taken in hybridization with the DE algorithm with the sine map given in Equation (53).

10.1.2. Analyzing the Effectiveness of the Proposed Hybridization

In this section, we present the results of performing the statistical comparison between the performance of the proposed ICDEBOA algorithm and several state-of-the-art algorithms in literature, including original BOA [21], SHADE [34], HPSOBOA [40], and jDE [35], evaluated over 30 objective functions defined in the CEC2014 benchmark.

The obtained mean (Mean) and standard (STD) deviation values of the best results of each algorithm obtained on each test objective function over each dimension are shown in Table 2. In addition, an appropriate sign is assigned to each tested algorithm, except the proposed ICDEBOA, to indicate whether the current algorithm achieved better $+$, worse $-$ or a similar $\approx$ solution to the proposed algorithm.

Table 2. Mean and standard deviation of the best objective function value obtained by the considered algorithms on a set of the CEC2014 test functions.

		ICDEBOA	HPSOBOA	jDE	BOA	SHADE
				Mean (STD) Sign		
f_1	10	7.80×10^{-15} (8.67×10^{-15})	2.51×10^{8} (1.22×10^{08})−	2.24×10^{6} (2.59×10^{6})−	2.76×10^{6} (2.10×10^{6})−	2.03×10^{6} (1.15×10^{6})−
	30	1.28×10^{4} (1.04×10^{4})	1.28×10^{4} (1.03×10^{4})−	5.80×10^{7} (3.57×10^{7})−	1.62×10^{9} (5.33×10^{8})−	2.32×10^{8} (6.41×10^{7})−
	50	1.28×10^{4} (1.04×10^{4})	1.24×10^{10} (3.15×10^{9})−	5.80×10^{7} (3.57×10^{7})−	1.62×10^{9} (5.33×10^{8})−	2.32×10^{8} (6.41×10^{7})−
	100	4.74×10^{5} (1.91×10^{5})	4.74×10^{5} (1.91×10^{5})−	3.13×10^{8} (1.25×10^{8})−	4.47×10^{9} (9.49×10^{8})−	8.87×10^{8} (1.61×10^{8})−
f_2	10	0.00×10^{0} (0.00×10^{0})	1.22×10^{10} (2.28×10^{9})−	2.69×10^{3} (3.81×10^{3})−	2.78×10^{8} (2.42×10^{8})−	3.04×10^{5} (1.60×10^{5})−
	30	2.46×10^{-13} (1.01×10^{-13})	9.79×10^{10} (1.49×10^{10})−	5.20×10^{9} (3.36×10^{9})−	9.79×10^{10} (1.49×10^{10})−	2.74×10^{8} (9.07×10^{7})−
	50	2.46×10^{-13} (1.01×10^{-13})	1.90×10^{11} (2.41×10^{9})−	5.20×10^{9} (3.36×10^{9})−	9.79×10^{10} (1.49×10^{10})−	2.74×10^{8} (9.07×10^{7})−
	100	2.31×10^{-12} (4.23×10^{-12})	2.31×10^{-12} (4.27×10^{-12})−	3.18×10^{10} (7.48×10^{9})−	2.36×10^{11} (1.60×10^{10})−	5.52×10^{9} (1.06×10^{9})−
f_3	10	0.00×10^{0} (0.00×10^{0})	1.62×10^{4} (2.13×10^{3})−	2.24×10^{3} (2.82×10^{3})−	8.10×10^{2} (2.90×10^{2})−	1.13×10^{3} (6.34×10^{2})−
	30	6.31×10^{-13} (3.62×10^{-13})	1.27×10^{5} (1.24×10^{4})−	4.73×10^{4} (1.15×10^{4})−	1.27×10^{5} (1.24×10^{4})−	3.00×10^{5} (1.46×10^{5})−
	50	6.31×10^{-13} (3.62×10^{-13})	1.88×10^{5} (4.85×10^{3})−	4.73×10^{4} (1.15×10^{4})−	1.27×10^{5} (1.24×10^{4})−	3.00×10^{5} (1.46×10^{5})−
	100	3.73×10^{-12} (5.71×10^{-12})	3.79×10^{-12} (5.76×10^{-12})−	1.45×10^{5} (1.64×10^{4})−	2.64×10^{5} (1.62×10^{4})−	7.03×10^{5} (2.23×10^{5})−
f_4	10	2.18×10^{1} (1.63×10^{1})	2.18×10^{1} (1.63×10^{1})−	3.31×10^{1} (7.85×10^{0})−	6.65×10^{2} (3.59×10^{2})−	3.43×10^{1} (3.25×10^{0})−
	30	3.51×10^{0} (1.49×10^{1})	2.02×10^{0} (1.07×10^{1})≈	5.43×10^{2} (2.22×10^{2})−	2.87×10^{4} (4.08×10^{3})−	2.67×10^{2} (4.55×10^{1})−
	50	3.51×10^{0} (1.49×10^{1})	6.47×10^{4} (2.93×10^{3})−	5.43×10^{2} (2.22×10^{2})−	2.87×10^{4} (4.08×10^{3})−	2.67×10^{2} (4.55×10^{1})−
	100	1.55×10^{2} (4.59×10^{1})	1.55×10^{2} (4.59×10^{1})−	2.39×10^{3} (9.13×10^{2})−	6.59×10^{4} (7.89×10^{3})−	1.06×10^{3} (1.07×10^{2})−
f_5	10	4.37×10^{0} (7.92×10^{0})	4.37×10^{0} (7.92×10^{0})−	1.98×10^{1} (2.83×10^{0})−	1.99×10^{1} (1.17×10^{0})−	2.09×10^{1} (1.41×10^{-1})−
	30	2.03×10^{1} (4.18×10^{-2})	2.03×10^{1} (4.20×10^{-2})−	2.11×10^{1} (4.71×10^{-2})−	2.11×10^{1} (2.76×10^{-2})−	2.14×10^{1} (4.64×10^{-2})−
	50	2.03×10^{1} (4.18×10^{-2})	2.13×10^{1} (3.36×10^{-2})−	2.11×10^{1} (4.71×10^{-2})−	2.11×10^{1} (2.76×10^{-2})−	2.14×10^{1} (4.64×10^{-2})−
	100	2.07×10^{1} (4.29×10^{-2})	2.07×10^{1} (4.29×10^{-2})−	2.13×10^{1} (2.50×10^{-2})−	2.13×10^{1} (2.37×10^{-2})−	2.15×10^{1} (3.22×10^{-2})−
f_6	10	3.55×10^{-1} (6.06×10^{-1})	3.55×10^{-1} (6.06×10^{-1})≈	1.44×10^{0} (8.74×10^{-1})−	4.55×10^{0} (4.65×10^{-1})−	6.27×10^{0} (1.81×10^{0})−
	30	2.35×10^{1} (4.22×10^{0})	2.32×10^{1} (4.19×10^{0})≈	2.65×10^{1} (3.96×10^{0})−	5.89×10^{1} (2.00×10^{0})−	6.80×10^{1} (5.33×10^{0})−
	50	2.35×10^{1} (4.22×10^{0})	2.32×10^{1} (4.19×10^{0})≈	2.65×10^{1} (3.96×10^{0})−	5.89×10^{1} (2.00×10^{0})−	6.80×10^{1} (5.33×10^{0})−
	100	7.02×10^{1} (1.19×10^{1})	7.02×10^{1} (1.19×10^{1})≈	8.03×10^{1} (5.06×10^{0})−	1.40×10^{2} (3.28×10^{0})−	1.53×10^{2} (7.13×10^{0})−
f_7	10	7.46×10^{-2} (3.99×10^{-2})	7.46×10^{-2} (3.99×10^{-2})≈	4.78×10^{-1} (5.82×10^{-1})−	9.20×10^{1} (3.01×10^{1})−	9.22×10^{-1} (8.92×10^{-2})−
	30	1.52×10^{-2} (2.58×10^{-2})	1.82×10^{-2} (3.05×10^{-2})−	4.60×10^{1} (3.21×10^{1})−	1.27×10^{3} (8.60×10^{1})−	3.72×10^{0} (7.66×10^{-1})−
	50	1.52×10^{-2} (2.58×10^{-2})	1.82×10^{-2} (3.05×10^{-2})−	4.60×10^{1} (3.21×10^{1})−	1.27×10^{3} (8.60×10^{1})−	3.72×10^{0} (7.66×10^{-1})−
	100	1.01×10^{-1} (3.33×10^{-1})	1.01×10^{-1} (3.33×10^{-1})≈	2.73×10^{2} (6.99×10^{1})−	2.78×10^{3} (9.78×10^{1})−	5.38×10^{1} (1.27×10^{1})−

Table 2. *Cont.*

		ICDEBOA	HPSOBOA	jDE	BOA	SHADE
				Mean (STD) Sign		
f_8	10	0.00×10^{0} (0.00×10^{0})	9.48×10^{1} (5.62×10^{0})−	5.53×10^{0} (3.07×10^{0})−	3.98×10^{1} (6.06×10^{0})−	3.84×10^{1} (8.60×10^{0})−
	30	2.58×10^{1} (2.35×10^{1})	2.50×10^{1} (2.23×10^{1})−	1.73×10^{2} (3.13×10^{1})−	5.51×10^{2} (2.24×10^{1})−	4.38×10^{2} (2.91×10^{1})−
	50	2.58×10^{1} (2.35×10^{1})	2.50×10^{1} (2.23×10^{1})−	1.73×10^{2} (3.13×10^{1})−	5.51×10^{2} (2.24×10^{1})−	4.38×10^{2} (2.91×10^{1})−
	100	2.12×10^{2} (5.37×10^{1})	2.12×10^{2} (5.37×10^{1})−	5.56×10^{2} (4.88×10^{1})−	1.27×10^{3} (2.83×10^{1})−	1.02×10^{3} (4.43×10^{1})−
f_9	10	4.86×10^{-1} (8.27×10^{-1})	4.86×10^{-1} (8.27×10^{-1})−	1.04×10^{1} (5.22×10^{0})−	3.87×10^{1} (6.38×10^{0})−	5.79×10^{1} (1.05×10^{1})−
	30	8.94×10^{1} (2.29×10^{1})	8.92×10^{1} (2.34×10^{1})≈	1.82×10^{2} (3.17×10^{1})−	6.10×10^{2} (2.70×10^{1})−	5.02×10^{2} (3.41×10^{1})−
	50	8.94×10^{1} (2.29×10^{1})	8.92×10^{1} (2.34×10^{1})≈	1.82×10^{2} (3.17×10^{1})−	6.10×10^{2} (2.70×10^{1})−	5.02×10^{2} (3.41×10^{1})−
	100	3.47×10^{2} (6.33×10^{1})	3.47×10^{2} (6.33×10^{1})−	5.63×10^{2} (6.26×10^{1})−	1.38×10^{3} (4.42×10^{1})−	1.13×10^{3} (5.30×10^{1})−
f_{10}	10	4.42×10^{-1} (9.31×10^{-1})	4.42×10^{-1} (9.31×10^{-1})−	2.16×10^{2} (1.29×10^{2})−	9.91×10^{2} (1.20×10^{2})−	1.58×10^{3} (3.08×10^{2})−
	30	1.58×10^{0} (1.30×10^{0})	1.49×10^{0} (1.15×10^{0})+	5.28×10^{3} (7.33×10^{2})−	1.29×10^{4} (3.37×10^{2})−	1.39×10^{4} (8.79×10^{2})−
	50	1.58×10^{0} (1.30×10^{0})	1.49×10^{0} (1.15×10^{0})+	5.28×10^{3} (7.33×10^{2})−	1.29×10^{4} (3.37×10^{2})−	1.39×10^{4} (8.79×10^{2})−
	100	1.60×10^{0} (1.02×10^{0})	1.60×10^{0} (1.02×10^{0})−	1.56×10^{4} (9.18×10^{2})−	3.02×10^{4} (6.48×10^{2})−	3.18×10^{4} (1.11×10^{3})−
f_{11}	10	1.30×10^{1} (1.13×10^{1})	1.30×10^{1} (1.13×10^{1})+	3.26×10^{2} (1.67×10^{2})−	1.07×10^{3} (1.36×10^{2})−	2.14×10^{3} (2.62×10^{2})−
	30	3.91×10^{3} (4.15×10^{2})	3.88×10^{3} (3.89×10^{2})≈	5.65×10^{3} (7.51×10^{2})−	1.32×10^{4} (4.71×10^{2})−	1.62×10^{4} (7.22×10^{2})−
	50	3.91×10^{3} (4.15×10^{2})	3.88×10^{3} (3.89×10^{2})≈	5.65×10^{3} (7.51×10^{2})−	1.32×10^{4} (4.71×10^{2})−	1.62×10^{4} (7.22×10^{2})−
	100	1.23×10^{4} (1.05×10^{3})	1.23×10^{4} (1.05×10^{3})+	1.38×10^{4} (2.76×10^{3})−	3.07×10^{4} (6.24×10^{2})−	3.47×10^{4} (9.68×10^{2})−
f_{12}	10	1.12×10^{-1} (2.90×10^{-2})	1.12×10^{-1} (2.90×10^{-2})−	3.31×10^{-1} (3.61×10^{-1})−	9.85×10^{-1} (1.31×10^{-1})−	3.39×10^{0} (1.02×10^{0})−
	30	3.05×10^{-1} (4.66×10^{-2})	3.08×10^{-1} (4.45×10^{-2})≈	1.82×10^{0} (1.57×10^{0})−	3.41×10^{0} (3.14×10^{-1})−	6.91×10^{0} (8.13×10^{-1})−
	50	3.05×10^{-1} (4.66×10^{-2})	3.08×10^{-1} (4.45×10^{-2})≈	1.82×10^{0} (1.57×10^{0})−	3.41×10^{0} (3.14×10^{-1})−	6.91×10^{0} (8.13×10^{-1})−
	100	5.84×10^{-1} (6.63×10^{-2})	5.84×10^{-1} (6.63×10^{-2})−	3.35×10^{0} (1.48×10^{0})−	4.10×10^{0} (2.60×10^{-1})−	6.63×10^{0} (6.11×10^{-1})−
f_{13}	10	6.02×10^{-2} (1.36×10^{-2})	6.02×10^{-2} (1.36×10^{-2})+	1.13×10^{-1} (4.66×10^{-2})−	2.74×10^{0} (5.86×10^{-1})−	5.55×10^{-1} (1.42×10^{-1})−
	30	3.91×10^{-1} (6.18×10^{-2})	3.90×10^{-1} (6.43×10^{-2})≈	6.10×10^{-1} (2.55×10^{-1})−	7.85×10^{0} (3.08×10^{-1})−	1.09×10^{0} (1.85×10^{-1})−
	50	3.91×10^{-1} (6.18×10^{-2})	3.90×10^{-1} (6.43×10^{-2})≈	6.10×10^{-1} (2.55×10^{-1})−	7.85×10^{0} (3.08×10^{-1})−	1.09×10^{0} (1.85×10^{-1})−
	100	5.28×10^{-1} (5.81×10^{-2})	5.28×10^{-1} (5.81×10^{-2})+	1.42×10^{0} (9.73×10^{-1})−	9.10×10^{0} (1.84×10^{-1})−	1.15×10^{0} (1.63×10^{-1})−
f_{14}	10	3.77×10^{-2} (1.37×10^{-2})	3.77×10^{-2} (1.37×10^{-2})≈	1.44×10^{-1} (1.62×10^{-1})−	2.06×10^{1} (5.32×10^{0})−	6.00×10^{-1} (1.53×10^{-1})−
	30	2.98×10^{-1} (7.30×10^{-2})	2.99×10^{-1} (7.19×10^{-2})+	1.03×10^{1} (9.86×10^{0})−	3.20×10^{2} (2.64×10^{1})−	1.38×10^{0} (5.45×10^{-1})−
	50	2.98×10^{-1} (7.30×10^{-2})	2.99×10^{-1} (7.19×10^{-2})+	1.03×10^{1} (9.86×10^{0})−	3.20×10^{2} (2.64×10^{1})−	1.38×10^{0} (5.45×10^{-1})−
	100	3.29×10^{-1} (3.16×10^{-2})	3.29×10^{-1} (3.16×10^{-2})≈	8.44×10^{1} (2.16×10^{1})−	8.29×10^{2} (3.69×10^{1})−	4.55×10^{0} (2.57×10^{0})−

Table 2. *Cont.*

		ICDEBOA	HPSOBOA	jDE	BOA	SHADE
				Mean (STD) Sign		
f_{15}	10	4.16×10^{-1} (1.04×10^{-1})	4.86×10^{4} (2.37×10^{4})−	1.00×10^{0} (4.28×10^{-1})−	2.01×10^{2} (2.15×10^{2})−	5.65×10^{0} (9.89×10^{-1})−
	30	1.97×10^{1} (5.86×10^{0})	1.48×10^{7} (3.72×10^{6})−	4.91×10^{2} (8.53×10^{2})−	8.49×10^{5} (3.97×10^{5})−	5.92×10^{1} (8.10×10^{0})−
	50	1.97×10^{1} (5.86×10^{0})	1.48×10^{7} (3.72×10^{6})−	4.91×10^{2} (8.53×10^{2})−	8.49×10^{5} (3.97×10^{5})−	5.92×10^{1} (8.10×10^{0})−
	100	1.28×10^{2} (3.94×10^{1})	4.46×10^{7} (5.50×10^{6})−	1.38×10^{4} (9.95×10^{3})−	7.84×10^{6} (2.61×10^{6})−	2.12×10^{3} (1.55×10^{3})−
f_{16}	10	3.87×10^{-1} (2.67×10^{-1})	3.70×10^{0} (3.63×10^{-2})−	1.90×10^{0} (5.79×10^{-1})−	2.80×10^{0} (2.30×10^{-1})−	4.15×10^{0} (2.00×10^{-1})−
	30	1.72×10^{1} (4.31×10^{-1})	2.32×10^{1} (5.31×10^{-2})−	1.95×10^{1} (9.41×10^{-1})−	2.22×10^{1} (1.81×10^{-1})−	2.39×10^{1} (2.54×10^{-1})−
	50	1.72×10^{1} (4.31×10^{-1})	2.32×10^{1} (5.31×10^{-2})−	1.95×10^{1} (9.41×10^{-1})−	2.22×10^{1} (1.81×10^{-1})−	2.39×10^{1} (2.54×10^{-1})−
	100	4.02×10^{1} (5.16×10^{-1})	4.76×10^{1} (1.65×10^{-1})−	4.35×10^{1} (1.77×10^{0})−	4.64×10^{1} (2.31×10^{-1})−	4.85×10^{1} (3.34×10^{-1})−
f_{17}	10	9.66×10^{1} (7.45×10^{1})	6.57×10^{5} (1.29×10^{5})−	4.25×10^{3} (3.01×10^{3})−	1.28×10^{3} (3.00×10^{2})−	4.78×10^{4} (5.45×10^{4})−
	30	2.12×10^{3} (6.35×10^{2})	8.49×10^{8} (2.51×10^{7})−	2.93×10^{6} (2.81×10^{6})−	7.92×10^{7} (5.10×10^{7})−	4.56×10^{7} (1.86×10^{7})−
	50	2.12×10^{3} (6.35×10^{2})	8.49×10^{8} (2.51×10^{7})−	2.93×10^{6} (2.81×10^{6})−	7.92×10^{7} (5.10×10^{7})−	4.56×10^{7} (1.86×10^{7})−
	100	6.85×10^{4} (3.86×10^{4})	2.86×10^{9} (4.30×10^{8})−	2.38×10^{7} (1.17×10^{7})−	6.68×10^{8} (2.88×10^{8})−	1.89×10^{8} (6.03×10^{7})−
f_{18}	10	4.65×10^{0} (4.33×10^{0})	1.96×10^{6} (2.80×10^{6})−	4.40×10^{3} (5.30×10^{3})−	4.14×10^{2} (1.99×10^{2})−	1.92×10^{3} (2.70×10^{3})−
	30	6.75×10^{2} (7.08×10^{2})	2.93E+10 (4.38×10^{9})−	3.86×10^{7} (9.42×10^{7})−	4.80×10^{9} (2.14×10^{9})−	3.44×10^{6} (2.13×10^{6})−
	50	6.75×10^{2} (7.08×10^{2})	2.93E+10 (4.38×10^{9})−	3.86×10^{7} (9.42×10^{7})−	4.80×10^{9} (2.14×10^{9})−	3.44×10^{6} (2.13×10^{6})−
	100	1.72×10^{3} (1.61×10^{3})	4.90E+10 (2.59×10^{9})−	5.14×10^{8} (4.23×10^{8})−	2.19E+10 (4.77×10^{9})−	1.66×10^{7} (6.25×10^{6})−
f_{19}	10	3.66×10^{-3} (7.06×10^{-3})	9.37×10^{1} (3.80×10^{1})−	2.06×10^{0} (7.88×10^{-1})−	8.00×10^{0} (3.21×10^{0})−	3.98×10^{0} (1.11×10^{0})−
	30	1.49×10^{1} (2.80×10^{0})	5.27×10^{3} (1.16×10^{3})−	6.87×10^{1} (2.55×10^{1})−	1.43×10^{3} (4.46×10^{2})−	6.74×10^{1} (9.25×10^{0})−
	50	1.49×10^{1} (2.80×10^{0})	5.27×10^{3} (1.16×10^{3})−	6.87×10^{1} (2.55×10^{1})−	1.43×10^{3} (4.46×10^{2})−	6.74×10^{1} (9.25×10^{0})−
	100	8.71×10^{1} (2.95×10^{1})	1.55×10^{4} (7.68×10^{2})−	2.92×10^{2} (6.55×10^{1})−	6.46×10^{3} (1.10×10^{3})−	1.83×10^{2} (1.76×10^{1})−
f_{20}	10	8.00×10^{-1} (8.22×10^{-1})	8.08×10^{4} (2.04×10^{4})−	9.70×10^{2} (2.18×10^{3})−	5.04×10^{2} (2.79×10^{2})−	3.52×10^{1} (1.46×10^{1})−
	30	1.68×10^{2} (4.09×10^{1})	4.21×10^{5} (6.46×10^{4})−	1.35×10^{4} (6.09×10^{3})−	6.84×10^{4} (2.24×10^{4})−	1.25×10^{6} (1.63×10^{6})−
	50	1.68×10^{2} (4.09×10^{1})	4.21×10^{5} (6.46×10^{4})−	1.35×10^{4} (6.09×10^{3})−	6.84×10^{4} (2.24×10^{4})−	1.25×10^{6} (1.63×10^{6})−
	100	3.67×10^{2} (7.71×10^{1})	1.36×10^{7} (6.91×10^{6})−	4.86×10^{4} (1.74×10^{4})−	2.90×10^{5} (7.67×10^{4})−	2.75×10^{6} (2.31×10^{6})−
f_{21}	10	1.34×10^{1} (2.20×10^{1})	2.25×10^{6} (1.29×10^{6})−	4.26×10^{3} (3.91×10^{3})−	3.56×10^{3} (1.43×10^{3})−	2.10×10^{3} (6.44×10^{3})−
	30	1.45×10^{3} (4.95×10^{2})	3.12×10^{8} (4.04×10^{6})−	1.82×10^{6} (1.76×10^{6})−	4.34×10^{6} (2.96×10^{6})−	2.81×10^{7} (1.68×10^{7})−
	50	1.45×10^{3} (4.95×10^{2})	3.12×10^{8} (4.04×10^{6})−	1.82×10^{6} (1.76×10^{6})−	4.34×10^{6} (2.96×10^{6})−	2.81×10^{7} (1.68×10^{7})−
	100	1.58×10^{4} (9.32×10^{3})	8.35×10^{8} (1.43×10^{8})−	9.69×10^{6} (4.68×10^{6})−	1.57×10^{8} (6.71×10^{7})−	1.23×10^{8} (4.39×10^{7})−

Table 2. *Cont.*

		ICDEBOA	HPSOBOA	jDE	BOA	SHADE
				Mean (STD) Sign		
f_{22}	10	2.17×10^{-1} (1.85×10^{-1})	6.45×10^{2} $(1.99 \times 10^{2})-$	6.11×10^{1} $(5.32 \times 10^{1})-$	5.66×10^{1} $(1.51 \times 10^{1})-$	1.43×10^{2} $(1.07 \times 10^{2})-$
	30	3.97×10^{2} (1.50×10^{2})	2.87×10^{6} $(1.22 \times 10^{6})-$	7.20×10^{2} $(3.24 \times 10^{2})-$	6.27×10^{3} $(4.31 \times 10^{3})-$	2.76×10^{3} $(3.29 \times 10^{2})-$
	50	3.97×10^{2} (1.50×10^{2})	2.87×10^{6} $(1.22 \times 10^{6})-$	7.20×10^{2} $(3.24 \times 10^{2})-$	6.27×10^{3} $(4.31 \times 10^{3})-$	2.76×10^{3} $(3.29 \times 10^{2})-$
	100	1.47×10^{3} (3.58×10^{2})	5.47×10^{5} $(7.45 \times 10^{4})-$	1.80×10^{3} $(6.15 \times 10^{2})-$	2.57×10^{4} $(2.61 \times 10^{4})-$	6.24×10^{3} $(5.61 \times 10^{2})-$
f_{23}	10	-4.03×10^{2} (0.00×10^{0})	-5.33×10^{2} $(2.91 \times 10^{-4})+$	-4.02×10^{2} $(2.83 \times 10^{0})-$	-5.33×10^{2} $(4.51 \times 10^{0})+$	-4.03×10^{2} $(5.60 \times 10^{-3})-$
	30	-5.07×10^{3} (4.98×10^{-12})	-5.21×10^{3} $(1.39 \times 10^{-3})+$	-5.00×10^{3} $(3.20 \times 10^{1})-$	-5.21×10^{3} $(2.76 \times 10^{-12})+$	-5.06×10^{3} $(1.48 \times 10^{0})-$
	50	-5.07×10^{3} (4.98×10^{-12})	-5.21×10^{3} $(1.39 \times 10^{-3})+$	-5.00×10^{3} $(3.20 \times 10^{1})-$	-5.21×10^{3} $(2.76 \times 10^{-12})+$	-5.06×10^{3} $(1.48 \times 10^{0})-$
	100	-7.19×10^{3} (2.71×10^{-10})	-7.34×10^{3} $(1.73 \times 10^{-3})+$	-6.95×10^{3} $(6.80 \times 10^{1})-$	-7.34×10^{3} $(9.19 \times 10^{-12})+$	-7.10×10^{3} $(1.43 \times 10^{1})-$
f_{24}	10	-1.95×10^{2} (4.25×10^{0})	-1.01×10^{2} $(8.63 \times 10^{-1})-$	-1.81×10^{2} $(1.32 \times 10^{1})-$	-1.69×10^{2} $(6.91 \times 10^{0})-$	-1.40×10^{2} $(8.36 \times 10^{0})-$
	30	-8.71×10^{2} (6.20×10^{0})	-9.60×10^{2} $(2.03 \times 10^{-3})+$	-9.60×10^{2} $(2.51 \times 10^{-4})+$	-9.60×10^{2} $(8.04 \times 10^{-13})+$	-8.65×10^{2} $(3.59 \times 10^{0})-$
	50	-8.71×10^{2} (6.20×10^{0})	-9.60×10^{2} $(2.03 \times 10^{-3})+$	-9.60×10^{2} $(2.51 \times 10^{-4})+$	-9.60×10^{2} $(8.04 \times 10^{-13})+$	-8.65×10^{2} $(3.59 \times 10^{0})-$
	100	-1.71×10^{3} (9.79×10^{0})	-1.95×10^{3} $(3.90 \times 10^{-3})-$	-1.95×10^{3} $(5.76 \times 10^{-4})-$	-1.95×10^{3} $(6.89 \times 10^{-13})-$	-1.69×10^{3} $(6.59 \times 10^{0})-$
f_{25}	10	-1.86×10^{2} (3.33×10^{1})	-1.22×10^{2} $(2.78 \times 10^{-2})-$	-1.32×10^{2} $(2.51 \times 10^{1})-$	-1.42×10^{2} $(1.50 \times 10^{1})-$	-1.23×10^{2} $(8.76 \times 10^{0})-$
	30	-8.46×10^{2} (8.61×10^{0})	-8.62×10^{2} $(3.14 \times 10^{-5})+$	-8.37×10^{2} $(5.53 \times 10^{0})-$	-8.62×10^{2} $(1.15 \times 10^{-12})+$	-7.99×10^{2} $(9.93 \times 10^{0})-$
	50	-8.46×10^{2} (8.61×10^{0})	-8.62×10^{2} $(3.14 \times 10^{-5})+$	-8.37×10^{2} $(5.53 \times 10^{0})-$	-8.62×10^{2} $(1.15 \times 10^{-12})+$	-7.99×10^{2} $(9.93 \times 10^{0})-$
	100	-1.31×10^{3} (1.21×10^{1})	-1.40×10^{3} $(7.49 \times 10^{-5})-$	-1.40×10^{3} $(2.50 \times 10^{1})-$	-1.40×10^{3} $(9.19 \times 10^{-13})-$	-1.25×10^{3} $(1.55 \times 10^{1})-$
f_{26}	10	-7.92×10^{2} (1.58×10^{-2})	-7.87×10^{2} $(6.90 \times 10^{-1})-$	-7.92×10^{2} $(3.21 \times 10^{-2})-$	-7.92×10^{2} $(7.73 \times 10^{-2})-$	-7.92×10^{2} $(1.66 \times 10^{-1})-$
	30	-2.85×10^{3} (1.10×10^{-1})	-2.78×10^{3} $(2.42 \times 10^{1})-$	-2.77×10^{3} $(4.85 \times 10^{1})-$	-2.77×10^{3} $(3.19 \times 10^{1})-$	-2.77×10^{3} $(6.42 \times 10^{1})-$
	50	-2.85×10^{3} (1.10×10^{-1})	-2.78×10^{3} $(2.42 \times 10^{1})-$	-2.77×10^{3} $(4.85 \times 10^{1})-$	-2.77×10^{3} $(3.19 \times 10^{1})-$	-2.77×10^{3} $(6.42 \times 10^{1})-$
	100	-2.62×10^{3} (1.55×10^{-2})	-2.62×10^{3} $(2.39 \times 10^{-7})-$	-2.62×10^{3} $(2.02 \times 10^{-12})-$	-2.62×10^{3} $(2.30 \times 10^{-12})-$	-2.58×10^{3} $(1.50 \times 10^{1})-$
f_{27}	10	-3.25×10^{3} (1.82×10^{2})	-3.18×10^{3} $(2.23 \times 10^{-2})\approx$	-3.11×10^{3} $(1.53 \times 10^{2})-$	-3.37×10^{3} $(2.52 \times 10^{0})+$	-3.14×10^{3} $(1.96 \times 10^{2})-$
	30	-1.01×10^{4} (8.65×10^{1})	-1.10×10^{4} $(2.59 \times 10^{-2})+$	-1.02×10^{4} $(9.68 \times 10^{1})+$	-1.05×10^{4} $(1.11 \times 10^{2})+$	-9.64×10^{3} $(1.43 \times 10^{2})-$
	50	-1.01×10^{4} (8.65×10^{1})	-1.10×10^{4} $(2.59 \times 10^{-2})+$	-1.02×10^{4} $(9.68 \times 10^{1})+$	-1.05×10^{4} $(1.11 \times 10^{2})+$	-9.64×10^{3} $(1.43 \times 10^{2})-$
	100	-1.47×10^{4} (1.69×10^{2})	-1.70×10^{4} $(7.29 \times 10^{-2})\approx$	-1.49×10^{4} $(1.56 \times 10^{2})-$	-1.50×10^{4} $(5.29 \times 10^{2})+$	-1.41×10^{4} $(2.32 \times 10^{2})-$

Table 2. *Cont.*

		ICDEBOA	HPSOBOA	jDE	BOA	SHADE
				Mean (STD) Sign		
f_{28}	10	-6.04×10^3 (5.49×10^1)	-6.26×10^3 (3.34×10^{-2})+	-6.04×10^3 (7.17×10^1)≈	-6.03×10^3 (6.97×10^1)−	-6.03×10^3 (8.59×10^1)−
	30	-2.28×10^4 (2.39×10^2)	-2.43×10^4 (9.75×10^{-2})+	-2.26×10^4 (4.68×10^2)−	-2.14×10^4 (5.24×10^2)−	-2.20×10^4 (1.00×10^3)−
	50	-2.28×10^4 (2.39×10^2)	-2.43×10^4 (9.75×10^{-2})+	-2.26×10^4 (4.68×10^2)−	-2.14×10^4 (5.24×10^2)−	-2.20×10^4 (1.00×10^3)−
	100	-5.38×10^4 (9.93×10^2)	-5.82×10^4 (1.43×10^{-1})+	-5.21×10^4 (9.93×10^2)≈	-4.84×10^4 (1.62×10^3)−	-5.00×10^4 (2.77×10^3)−
f_{29}	10	-1.02×10^9 (2.43×10^1)	-1.02×10^9 (8.96×10^3)−	-1.02×10^9 (4.15×10^5)−	-1.02×10^9 (5.55×10^2)−	-1.02×10^9 (2.84×10^4)−
	30	-7.37×10^9 (1.52×10^2)	-7.37×10^9 (1.15×10^5)−	-7.37×10^9 (2.92×10^6)−	-7.37×10^9 (3.85×10^{-6})+	-7.37×10^9 (5.95×10^5)−
	50	-7.37×10^9 (1.52×10^2)	-7.37×10^9 (1.15×10^5)−	-7.37×10^9 (2.92×10^6)−	-7.37×10^9 (3.85×10^{-6})+	-7.37×10^9 (5.95×10^5)−
	100	−2.71E+10 (2.79×10^2)	−2.71E+10 (1.27×10^5)−	−2.70E+10 (2.40×10^7)−	−2.71E+10 (0.00×10^0)−	−2.71E+10 (1.92×10^6)−
f_{30}	10	-4.40×10^8 (7.29×10^1)	-4.40×10^8 (3.37×10^4)−	-4.40×10^8 (4.53×10^2)−	-4.40×10^8 (2.77×10^2)−	-4.40×10^8 (4.99×10^2)−
	30	-6.15×10^8 (9.26×10^2)	-6.15×10^8 (3.91×10^3)+	-6.15×10^8 (4.38×10^4)−	-6.15×10^8 (6.02×10^{-7})+	-6.14×10^8 (7.08×10^4)−
	50	-6.15×10^8 (9.26×10^2)	-6.15×10^8 (3.91×10^3)+	-6.15×10^8 (4.38×10^4)−	-6.15×10^8 (6.02×10^{-7})+	-6.14×10^8 (7.08×10^4)−
	100	-3.02×10^9 (1.73×10^3)	-3.02×10^9 (2.04×10^4)−	-3.02×10^9 (9.39×10^5)−	-3.02×10^9 (9.63×10^{-7})−	-3.02×10^9 (1.23×10^6)−

Based on the findings in Table 2, we observe that the proposed ICDEBOA algorithm outperforms other well-known algorithms on the large number of considered benchmark objective functions, and only on several isolated cases does the proposed algorithm has worse or similar performance to other considered algorithms. It should be noted that the ICDEBOA algorithm outperformed the SHADE algorithm on all considered problems. Results demonstrated that the HPSOBOA algorithm is in terms of performance right behind the ICDEBOA algorithm in the majority of cases, and successfully outperformed the ICDEBOA algorithm on f_{23} and f_{28} over all dimensions, and on f_6 HPSODEBOA showed similar performance to ICDEBOA. On $D = 10$ and $D = 100$, the HPSOBOA achieved better solutions on functions f_{11}, f_{13}, f_{23} and f_{28}. Over dimensions $D = 30$ and $D = 50$, the HPSOBOA showed better performance over composition functions $f_{23}, f_{24}, f_{25}, f_{27}$. The proposed and HPSOBOA algorithm achieved similar performance for $D = 30$ and 50 over simple multi-modal functions f_4, f_6, f_9, f_{11} and f_{13}, as well as for $D = 10$ and 100 over functions f_6, f_7, f_{14} and f_{27}. The jDE algorithm performed better than ICDEBOA on $D = 30$ and 50 on composite functions f_{24} and f_{27}, while the algorithms demonstrated similar performance on function $f_{28}, D = 10$. The original BOA algorithm outperformed the proposed method on f_{23} and f_{27} functions over all dimensions, and f_{24} and f_{25} on $D = 30$ and 50. Since the proposed method ICDEBOA had significantly better performance than other algorithms in the majority of cases, especially compared to BOA and SHADE, it further proves the validity of the proposed hybridization.

Using the Wilcoxon signed-rank test, the performance of the proposed ICDEBOA algorithm is compared to the performance of other algorithms that have been considered. Table 3 displays the statistical and comparative findings of the Wilcoxon signed-rank test applied to the CEC2014 benchmark problems.

Table 3. The results of performing Wilcoxon test on CEC2014 objective functions for all considered methods.

D	Algorithms	R^+	R^-	*p* Value	+	≈	−	Dec.
10	ICDEBOA vs. BOA	432	33	5.97×10^{-6}	28	2	0	+
	ICDEBOA vs. SHADE	465	0	1.86×10^{-9}	30	0	0	+
	ICDEBOA vs. HPSOBOA	412	53	7.91×10^{-5}	21	5	4	+
	ICDEBOA vs. jDE	465	0	1.86×10^{-9}	29	0	1	+
30	ICDEBOA vs. BOA	398	67	3.45×10^{-4}	24	6	0	+
	ICDEBOA vs. SHADE	465	0	1.86×10^{-9}	30	0	0	+
	ICDEBOA vs. HPSOBOA	292	173	2.29×10^{-1}	17	6	7	≈
	ICDEBOA vs. jDE	443	22	9.98×10^{-7}	28	2	0	+
50	ICDEBOA vs. BOA	398	67	3.45×10^{-4}	24	6	0	+
	ICDEBOA vs. SHADE	465	0	1.86×10^{-9}	30	0	0	+
	ICDEBOA vs. HPSOBOA	334	131	3.64×10^{-2}	17	5	8	≈
	ICDEBOA vs. jDE	443	22	9.98×10^{-7}	28	2	0	+
100	ICDEBOA vs. BOA	398	67	3.45×10^{-4}	28	2	0	+
	ICDEBOA vs. SHADE	465	0	1.86×10^{-9}	30	0	0	+
	ICDEBOA vs. HPSOBOA	320.5	144.5	8.70×10^{-2}	22	4	4	+
	ICDEBOA vs. jDE	434	31	4.42×10^{-6}	29	0	1	+

Based on the obtained results in Table 3, we observe that the ICDEBOA algorithm provides significantly better performance comapred to other considered algorithms on all dimensions. Compared to conventional BOA and SHADE algorithms, we observe improvement of performance, which further confirms the validity of the proposed hybridization. On $D = 30$ and $D = 50$ ICDEBOA achieved similar performance to HPSOBOA algorithm; however, the proposed algorithm has the higher value R^+ compared to R^-.

Moreover, the Friedman test is used to assess if there is a statistically significant difference between performance of considered algorithms. In this way, Table 4 shows the average rankings of the Friedman test for each method studied on the different functions and dimensions of the CEC2014 benchmark problems. Here, the first-place rank is displayed in bold, and underline id used to denoted second one.

Table 4. Results of the performance comparison using Friedman test with significance $\alpha = 0.05$ on CEC2014 test problems.

Algorithm	10D	30D	50D	100D	Mean Ranking	Rank
ICDEBOA	1.23	1.77	1.67	1.75	1.60	1
HPSOBOA	3.50	2.60	2.97	2.65	2.93	2
jDE	3.07	3.10	3.00	3.10	3.07	3
BOA	3.37	3.60	3.50	3.57	3.51	4
SHADE	3.83	3.93	3.87	3.93	3.89	5
Friedman p-value	2.90×10^{-10}	4.50×10^{-7}	1.01×10^{-6}	5.34×10^{-7}		

The results in Table 4, show that the obtained Friedman p values are less than significance α in each considered case. The results demonstrate that the proposed ICDEBOA algorithm has the best performance among all the algorithms taken for considerations. The HPSOBOA algorithm shows the second-best performance. According to statistical analysis, hybridising DE and BOA algorithms improves optimization performance while overcoming their drawbacks. The expanded mutation operators, which incorporate butterfly algorithms equations for position updating, and the inclusion of a chaotic map to the sensory fragrance greatly improve the ICDEBOA method, proving the usefulness of the introduced changes.

10.2. Evaluation of Accuracy of Localization

This section presents the results of the numerical simulations of localization performance comparison between the ICDEBOA alg., proposed in this work, and the widely applied CWLS, DE, and BOA methods on the considered TDOA based passive target localization problem. The goal of this section is to verify the proposed improvements introduced in ICDEBOA algorithm and its efficiency on the considered localization problem. In order to measure the localization efficiency, the algorithms are evaluated through RMSE performance and compared to the derived CRLB, as it represents the lower bound on the variance of any unbiased estimator. To measure the localization accuracy, the RMSE is employed, which can be calculated as

$$RMSE = \sqrt{\frac{1}{N_m} \sum_{n=1}^{N} \left\| \hat{\mathbf{x}}^{(n)} - \mathbf{x} \right\|_2^2}, \tag{68}$$

where $\mathbf{x}$ is the location of target which is known, $\hat{\mathbf{x}}^{(n)}$ is the estimate of the target position and $N_m = 1000$ denotes how much Monte Carlo simulation iterations are made for fixed σ^2.

It has been shown that some algorithms are sensitive to relative target–transmitter–receiver positions in the localization process, which leads to the singularity and bad solutions. To analyze how this relative position affects the localization performance of the considered algorithms, we consider three characteristic cases in the simulation. Thus, in the first four simulations, the transmitters are fixed to locations $\mathbf{x_1^t} = \begin{bmatrix} 100 & 100 \end{bmatrix}^{\mathbf{T}}$ m, $\mathbf{x_2^t} = \begin{bmatrix} 100 & -100 \end{bmatrix}^{\mathbf{T}}$ m, $\mathbf{x_3^t} = \begin{bmatrix} -100 & 100 \end{bmatrix}^{\mathbf{T}}$m, and $\mathbf{x_4^t} = \begin{bmatrix} -100 & -100 \end{bmatrix}^{\mathbf{T}}$ m, and as stated in Section 3, we place the receiver in the center of the introduced coordinate system. The arrangement of transmitters forms a convex polygon, and we firstly consider how the placement of the target near the centroid of the figure formed by transmitter locations, at $\mathbf{x} = \begin{bmatrix} 20 & 30 \end{bmatrix}^{\mathbf{T}}$, affect the localization. Then, in the second simulation the position of the target is moved outside the convex figure formed by transmitters at $\mathbf{x} = \begin{bmatrix} 120 & 130 \end{bmatrix}^{T}$. Finally, we consider the case when we assign the target a random location in the area $150 \times 150\text{m}^2$ in each run of the simulation.

The results of the first simulation scenario are depicted in the Figure 4 as the functional relationship between the measurement noise $p = 10\log(\sigma^2)$ and RMSE performance and compared to the CRLB.

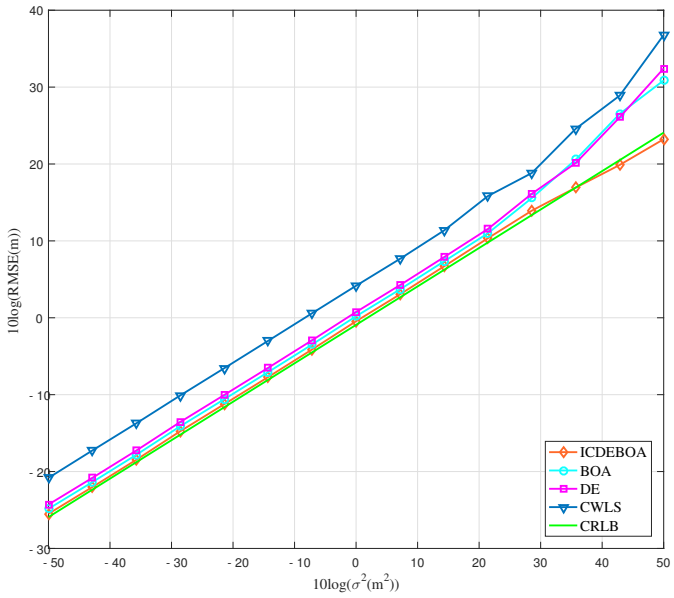

Figure 4. RMSE performance of different algorithms vs. p for first simulation case.

From the Figure 4 it can be concluded that the proposed ICDEBOA algorithm attains the CRLB and outperforms other considered methods on a whole considered range of measurement noise p. The performance of DE and BOA algorithms is slightly above the CRLB, while the CWLS algorithm showed the worse performance, especially when $p > 30$ dB.

Figure 5 presents the results of the second scenario, where we place the target in the position outside of polygon formed by transmitter locations. Here, the RMSE performances of considered algorithms are plotted as a function of the measurement noise, and compared to the CRLB.

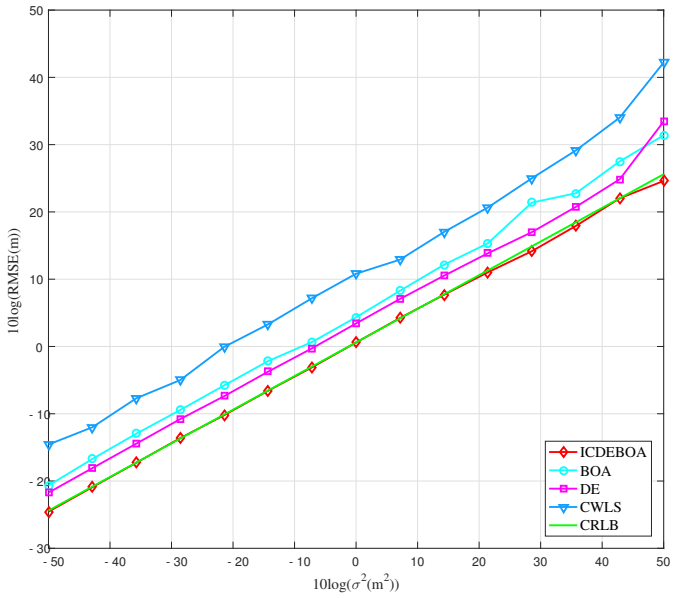

Figure 5. Plot of the obtained RMSE curves in relation to the measurement noise for second simulation case.

Based on the results in Figure 5 we observe that even in this case the proposed ICDEBOA algorithm is able to attain the CRLB accuracy and show better performance with regard to other algorithms taken for simulation. In this case, the DE and BOA algorithm show a slight degradation in performance, compared to the previous case, which is observed in the bigger discrepancy from the CRLB. Again, the CWLS method has the worst performance among all algorithms.

In Figure 6 the results of the third simulation case, which considers a random placement of target in the area 150×150 m^2 in each simulation iteration, are depicted as a function RMSE of p.

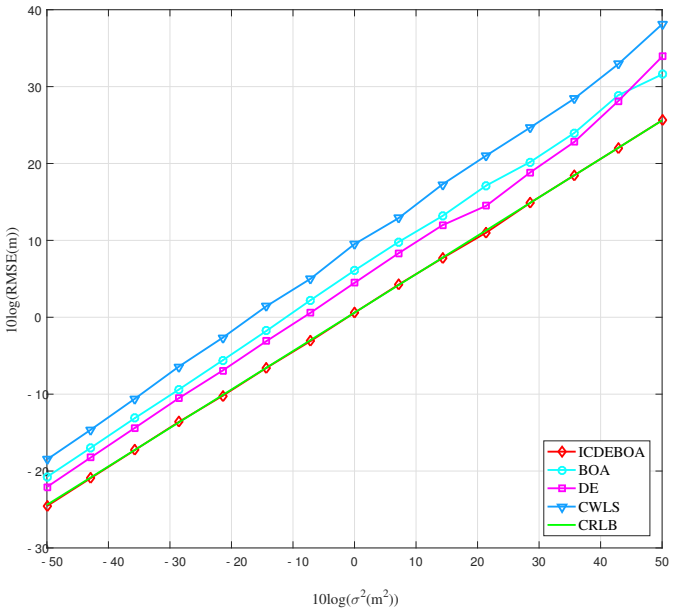

Figure 6. The results obtained for the third simulation case depicted in the form of RMSE performance of different algorithms vs. p.

Based on the results of the third simulation scenario depicted in Figure 6, we observe that the ICDEBOA algorithm is able to reach the CRLB accuracy, while outperforming CWLS, DE and BOA algorithms. In this case, it is notable that there is a higher discrepancy between the CRLB and the DE, BOA, and CWLS algorithms' RMSE performance curves. As in previous cases, the CWLS showed the lowest accuracy, with the curve high above the CRLB.

If we compare the results in Figure 4 through Figure 6 and analyze the obtained RMSE values under different conditions, we find that the considered algorithms had the performance closest to the CRLB in the first simulation case when we placed the target in the area surrounded by transmitters. Alternatively, we observe the biggest descrepency from the CRLB in the case when the target is not surrounded by transmitters, as shown in Figure 5. In addition, it is observed that the proposed ICDEBOA algorithm holds the CRLB accuracy for the entire considered range of p in all of the scenarios that have been considered for simulation.

Further investigation into the localization performance of different approaches is conducted by considering the localization error of considered algorithm trough the cumulative distribution functions (CDFs) when the measurement noise is $\sigma^2 = 10$ m^2. The localization error (LE) is introduced for CDF analysis as the Euclidean norm of the actual and estimated positions discrepancy, and it is computed as

$$LE = \left\| \hat{\mathbf{x}}^{(n)} - \mathbf{x} \right\|_2, \forall n \in \{1, \ldots, N_m\} \tag{69}$$

The CDFs of the localization error for the considered algorithms for the passive target positioned as in the first simulation scenario are presented in Figure 7 as a function of LE.

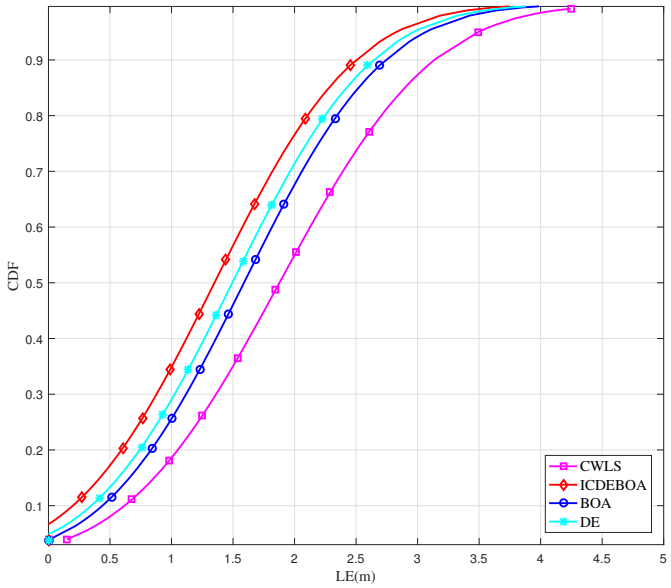

Figure 7. The obtained CDFs of *LE* for considered passive TDOA-based localization problem.

Based on the results shown in Figure 7, we can observe that the proposed ICDEBOA algorithm provides localization accuracy of less than 2.5 m for the 90% CDF percentile, while the BOA, DE, and CWLS algorithms achieve *LE*s of less than 2.74 m, 2.66 m, and 3.12 m, respectively, for the same case. So, we can say that the ICDEBOA algorithm improves LE by 9.6% compared to BOA, by 6.4% compared to DE, and by 24% compared to the CWLS method, proving that it has improved performance.

Lastly, we look at how the accuracy of the localization changes as the number of transmitters involved in localization increases. Consider N^j transmitters that are uniformly dispersed on a circle of a radius $R = 80\sqrt{2}$, such that the coordinate of each transmitter can be obtained as

$$\mathbf{x}_i^t = \begin{bmatrix} R\cos 2\pi / N^j{}_i \\ R\sin 2\pi / N^j{}_i \end{bmatrix}, \quad i \in \left\{1, 2, \ldots, N^j\right\} \tag{70}$$

where we fix the target inside the circle formed by transmitters near the center at the coordinates $x = \begin{bmatrix} 20 & 30 \end{bmatrix}^T$m.

The functional changes of RMSE curves with the change of the number of transmitters employed in the localization for the considered algorithms are displayed in Figure 8, along with the appropriate CRLB, for the variance of measurement noise $\sigma^2 = 1$ m^2.

Figure 8 shows that as the number of transmitters grows from 4 to 10, the RMSE performance of all algorithms under consideration improves noticeably. We observe that the proposed method is able to reach CRLB accuracy in each of the cases. Furthermore, we observe that CWLS algorithm deviates from the CRLB accuracy. If we consider the cases of lowest number of employed transmitters (4) and the case of highest number of transmitters (18) taken for simulation, we observe the improvement in accuracy of location estimation of 52.4% in the proposed method.

Based on simulation results, we came to the conclusion that the proposed ICDEBOA algorithm's performance was less affected by large-scale changes in measurement noise and network topologies than other algorithms we looked at.

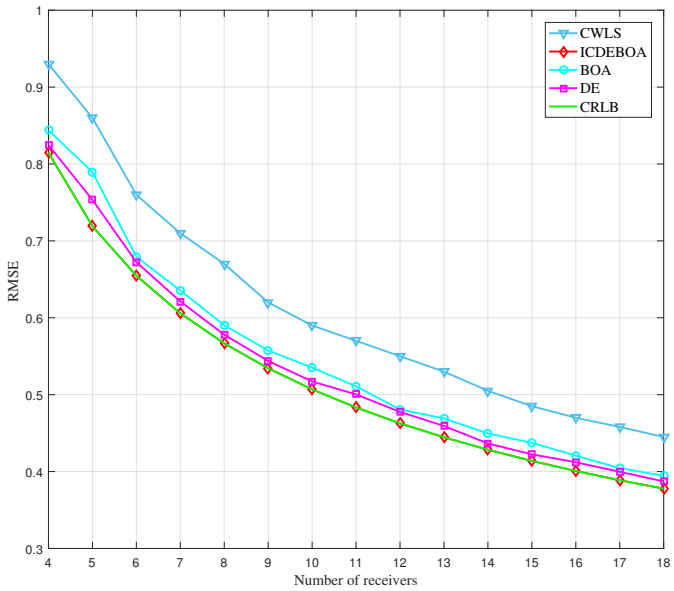

Figure 8. The change in RMSE performance of the considered algorithms with the increase in the participating transmitters for measurement noise variance $\sigma^2 = 1$ m^2.

11. Conclusions

This paper investigates the TDOA-based localization of a passive target in the presence of noise, which is in general a complex and multimodal problem. To solve this problem, the evolutionary algorithms DE and BOA are hybridized, resulting in the ICDEBOA algorithm. This paper aims to investigate the proposed ICDEBOA algorithm's performance and capabilities on the considered passive localization problem, with a focus on the high measurement noise case, where conventional algorithms perform poorly. In the ICDEBOA algorithm, a composite roulette wheel is introduced to the DE algorithm and employed in a new mutation operator to choose the appropriate mutation scheme for current optimization state. Using BOA expressions for particle movement to generate a mutant vector in the DE algorithm yields a hybrid algorithm that takes advantage of the BOA algorithm's strong local search capability. In addition, an adaptive method incorporating chaos theory is used to update the BOA algorithm's sensory fragrance, and adaptive parameter adjudication strategies from the SHADE algorithm are implemented in the DE algorithm. In order to perform the comparison of the localization performance with several well-known algorithms, the CRLB is derived for the considered problem of localization. In addition, to evaluate the optimization performance of the ICDEBOA algorithm and compare to state-of-the-art methods, a set of CEC2014 test functions are employed and statistical comparison is performed.

In the results of statistical comparison performed on CEC2014 test problems we observe that the algorithm proposed in this paper shows improved optimization performance. Regarding the localization accuracy, the proposed ICDEBOA algorithm successfully attains the CRLB and outperforms the CWLS, BOA, and SHADE algorithms in that aspect. In the case of a variation of the network topology, we observe that the ICDEBOA algorithm always achieves CRLB and is not affected with the network topology change.

The focus of upcoming research will be on energy-efficient network structure and performance evaluation in a non-line-of-sight environment. Analyzing the sensitivity to parameter changes can help the optimization process be further improved.

Author Contributions: Conceptualization, M.R. and M.S. (Miloš Sedak); Methodology and Validation, M.R.; Formal Analysis, Software, Investigation, Data Curation, M.R. and M.S. (Miloš Sedak); Resources, M.S. (Mirjana Simić) and P.P.; Writing—Original Draft Preparation, M.R. and M.S. (Miloš Sedak); Writing—Review and Editing, M.S. (Mirjana Simić) and P.P.; Visualization, M.R.; Supervision, M.S. (Mirjana Simić) and P.P. All authors have read and agreed to the published version of the manuscript.

Funding: The research of M. Rosić was supported by the Serbian Ministry of Education and Science under Grant TR35029. The research of M. Sedak was supported by the Serbian Ministry of Education and Science under Grant No. TR35006. The work of M. Simić was supported in part by Serbian Ministry of Education and Science under Grant TR32028.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The first partial derivative of Equation (8) is

$$\frac{\partial \ln L}{\partial \mathbf{x}} = -\frac{1}{2\sigma^2}\frac{\partial}{\partial \mathbf{x}}\Big((\tilde{\mathbf{r}}-\mathbf{d}(\mathbf{x}))^T(\tilde{\mathbf{r}}-\mathbf{d}(\mathbf{x}))\Big) = \frac{1}{\sigma^2}\frac{\partial \mathbf{r_g}(\mathbf{x})^T}{\partial \mathbf{x}}\mathbf{H}^T(\tilde{\mathbf{r}}-\mathbf{d}(\mathbf{x})) \tag{A1}$$

where

$$\mathbf{r}_g(\mathbf{x}) = \begin{bmatrix}\|\mathbf{x}_r-\mathbf{x}\| & \|\mathbf{x}_1^t-\mathbf{x}\| & \dots & \|\mathbf{x}_N^t-\mathbf{x}\|\end{bmatrix}^T, \tag{A2}$$

and

$$\mathbf{H} = \begin{bmatrix}\mathbf{1}_N\mathbf{I}_N\end{bmatrix} = \begin{bmatrix}1 & 1 & \cdots & 0 & 0\\ 1 & 0 & \cdots & 0 & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 1 & 0 & \cdots & 0 & 1\end{bmatrix}. \tag{A3}$$

Based on the expression in Equation (A1), we conclude that the componental partial derivative is

$$\frac{\partial \ln L}{\partial x} = \frac{1}{\sigma^2}\left(\frac{\partial \mathbf{r}_g(\mathbf{x})}{\partial x}\right)^T\mathbf{H}^T(\tilde{\mathbf{r}}-\mathbf{d}(\mathbf{x})). \tag{A4}$$

Then, the first element of matrix $\mathrm{F}(\mathbf{x})$ is

$$\begin{aligned}F_{11} &= E\left[\frac{\partial \ln L}{\partial x}\left(\frac{\partial \ln L}{\partial x}\right)^T\right]\\ &= E\left[\frac{1}{\sigma^4}\left(\frac{\partial \mathbf{r}_g(\mathbf{x})}{\partial x}\right)^T\mathbf{H}(\tilde{\mathbf{r}}-\mathbf{d}(\mathbf{x}))(\tilde{\mathbf{r}}-\mathbf{d}(\mathbf{x}))^T\mathbf{H}\left(\frac{\partial \mathbf{r}_g(\mathbf{x})}{\partial x}\right)\right]\\ &= \frac{1}{\sigma^2}\left(\frac{\partial \mathbf{r}_g(\mathbf{x})}{\partial x}\right)^T\mathbf{H}^T\mathbf{H}\left(\frac{\partial \mathbf{r}_g(\mathbf{x})}{\partial x}\right).\end{aligned} \tag{A5}$$

Analogus to this the remaining elements of $\mathrm{F}(\mathbf{x})$ are

$$F_{12} = F_{21} = E\left[\left(\frac{\partial \ln L}{\partial x}\right)\left(\frac{\partial \ln L}{\partial y}\right)^T\right] = \frac{1}{\sigma^2}\left(\frac{\partial \mathbf{r}_g(\mathbf{x})}{\partial x}\right)^T\mathbf{H}^T\mathbf{H}\left(\frac{\partial \mathbf{r}_g(\mathbf{x})}{\partial y}\right), \tag{A6}$$

$$F_{22} = E\left[\left(\frac{\partial \ln L}{\partial y}\right)\left(\frac{\partial \ln L}{\partial y}\right)^T\right] = \frac{1}{\sigma^2}\left(\frac{\partial \mathbf{r}_g(\mathbf{x})}{\partial y}\right)^T\mathbf{H}^T\mathbf{H}\left(\frac{\partial \mathbf{r}_g(\mathbf{x})}{\partial y}\right). \tag{A7}$$

If we take $\frac{\partial \mathbf{r}_g(\mathbf{x})}{\partial x}$, the following is obtained

$$\frac{\partial \mathbf{r}_g(\mathbf{x})}{\partial x} = \begin{bmatrix} \frac{x-x_r}{\|\mathbf{x}-\mathbf{x}_r\|_2} & \frac{y-y_r}{\|\mathbf{x}-\mathbf{x}_r\|_2} \\ \frac{x-x_1^t}{\|\mathbf{x}-\mathbf{x}_1^t\|_2} & \frac{y-y_1^t}{\|\mathbf{x}-\mathbf{x}_1^t\|_2} \\ \vdots & \vdots \\ \frac{x-x_N^t}{\|\mathbf{x}-\mathbf{x}_N^t\|_2} & \frac{y-y_N^t}{\|\mathbf{x}-\mathbf{x}_N^t\|_2} \end{bmatrix}, \tag{A8}$$

Then, the expressions in Equations (A5)–(A7) are simplified to

$$F_{11} = \frac{1}{\sigma^2}\sum_{i=1}^{N}\left(\frac{x-x_r}{\|\mathbf{x}-\mathbf{x}_r\|_2} + \frac{x-x_i^t}{\|\mathbf{x}-\mathbf{x}_i^t\|_2}\right)^2, \tag{A9}$$

$$F_{12} = F_{21} = \frac{1}{\sigma^2}\sum_{i=1}^{N}\left(\frac{x-x_r}{\|\mathbf{x}-\mathbf{x}_r\|_2} + \frac{x-x_i^t}{\|\mathbf{x}-\mathbf{x}_i^t\|_2}\right)\left(\frac{y-y_r}{\|\mathbf{x}-\mathbf{x}_r\|_2} + \frac{y-y_i^t}{\|\mathbf{x}-\mathbf{x}_i^t\|_2}\right), \tag{A10}$$

$$F_{22} = \frac{1}{\sigma^2}\sum_{i=1}^{N}\left(\frac{y-y_r}{\|\mathbf{x}-\mathbf{x}_r\|_2} + \frac{y-y_i^t}{\|\mathbf{x}-\mathbf{x}_i^t\|_2}\right)^2. \tag{A11}$$

Appendix B

Substituting Equation (23) into the Equation (24), the Lagrange multiplier must satisfy

$$\mathbf{b}^T\mathbf{W}\mathbf{A}\left(\mathbf{A}^T\mathbf{W}\mathbf{A}+\lambda\mathbf{S}\right)^{-1}\mathbf{S}\left(\mathbf{A}^T\mathbf{W}\mathbf{A}+\lambda\mathbf{S}\right)^{-1}\mathbf{A}^T\mathbf{W}\mathbf{b} = 0. \tag{A12}$$

Using eigenvalue decomposition, the matrix $\mathbf{A^T W A S}$ can be diagonalized as follows

$$\mathbf{A^T W A S} = \mathbf{\Sigma D \Sigma^{-1}}, \tag{A13}$$

where $\mathbf{D} = \mathrm{diag}\{\zeta_1, \zeta_2, \zeta_3\}$ and ζ_i, $i = 1, 2, 3$, are the eigenvalues of the matrix $\mathbf{A^T W A S}$, and $\mathbf{\Sigma}$ is the matrix composed of the eigenvectors. Substituting Equation (A13) into $\left(\mathbf{A}^T\mathbf{W}\mathbf{A}+\lambda\mathbf{S}\right)^{-1}$, yields the following equation

$$\left(\mathbf{A}^T\mathbf{W}\mathbf{A}+\lambda\mathbf{S}\right)^{-1} = \mathbf{S\Sigma}(\mathbf{D}+\lambda\mathbf{I})^{-1}\mathbf{\Sigma}^{-1}. \tag{A14}$$

Substituting this expression into Equation (A12), which after some manipulation, can be expressed in the form of λ as

$$\boldsymbol{\alpha}^T(\mathbf{D}+\lambda\mathbf{I})^{-2}\boldsymbol{\beta} = 0, \tag{A15}$$

where $\boldsymbol{\alpha} = \mathbf{\Sigma}^T\mathbf{S}\mathbf{A}^T\mathbf{W}\mathbf{b} = [\alpha_1\ \alpha_2\ \alpha_3]^T$ and $\boldsymbol{\beta} = [\beta_1\ \beta_2\ \beta_3]^T$. As a result of Equation (A15), we have Equation (26), which is a polynomial equation of the fourth degree.

References

1. Sachs, J. *Handbook of Ultra-Wideband Short-Range Sensing: Theory, Sensors, Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2013. [CrossRef]
2. Chalise, B.K.; Zhang, Y.D.; Amin, M.G.; Himed, B. Target localization in a multi-static passive radar system through convex optimization. *Signal Process.* **2014**, *102*, 207–215. [CrossRef]
3. Shen, L.; Zhang, Q.; Pang, J.; Xu, H.; Li, P.; Xue, D. ANTspin: Efficient Absolute Localization Method of RFID Tags via Spinning Antenna. *Sensors* **2019**, *19*, 2194. [CrossRef] [PubMed]
4. Shen, M.; Wang, Y.; Jiang, Y.; Ji, H.; Wang, B.; Huang, Z. A new positioning method based on multiple ultrasonic sensors for autonomous mobile robot. *Sensors* **2019**, *20*, 17. [CrossRef] [PubMed]

5. Siegele, D.; Staso, U.D.; Piovano, M.; Marcher, C.; Matt, D.T. State of the art of non-vision-based localization technologies for AR in facility management. In Proceedings of the International Conference on Augmented Reality, Virtual Reality and Computer Graphics, Lecce, Italy, 7–10 September 2020; pp. 255–272.
6. Ulusar, U.D.; Celik, G.; Al-Turjman, F. Cognitive RF-based localization for mission-critical applications in smart cities: An overview. *Comput. Electr. Eng.* **2020**, *87*, 106780. [CrossRef]
7. Liu, X.; Wang, Y.; Zhou, M.; Nie, W.; Yang, X. Indoor Passive Localization With Channel State Information Using a Single Access Point. *IEEE Sens. J.* **2021**, *21*, 17085–17095. [CrossRef]
8. Choi, K.H.; Ra, W.S.; Park, S.Y.; Park, J.B. Robust least squares approach to passive target localization using ultrasonic receiver array. *IEEE Trans. Ind. Electron.* **2014**, *61*, 1993–2002. [CrossRef]
9. Noroozi, A.; Sebt, M.A. Target localization from bistatic range measurements in multi-transmitter multi-receiver passive radar. *IEEE Signal Process. Lett.* **2015**, *22*, 2445–2449. [CrossRef]
10. Zekavat, R.; Buehrer, R.M. *Handbook of Position Location: Theory, Practice and Advances*; John Wiley & Sons: Hoboken, NJ, USA, 2011; Volume 27. [CrossRef]
11. Destino, G.; Abreu, G. On the maximum likelihood approach for source and network localization. *IEEE Trans. Signal Process.* **2011**, *59*, 4954–4970. [CrossRef]
12. Noroozi, A.; Sebt, M.A. Weighted least squares target location estimation in multi-transmitter multi-receiver passive radar using bistatic range measurements. *IET Radar Sonar Navig.* **2016**, *10*, 1088–1097. [CrossRef]
13. Jin, B.; Xu, X.; Zhang, T. Robust time-difference-of-arrival (TDOA) localization using weighted least squares with cone tangent plane constraint. *Sensors* **2018**, *18*, 778. [CrossRef]
14. Wang, G.; Li, Y.; Ansari, N. A semidefinite relaxation method for source localization using TDOA and FDOA measurements. *IEEE Trans. Veh. Technol.* **2013**, *62*, 853–862. [CrossRef]
15. Yue, Y.; Cao, L.; Hu, J.; Cai, S.; Hang, B.; Wu, H. A Novel Hybrid Location Algorithm Based on Chaotic Particle Swarm Optimization for Mobile Position Estimation. *IEEE Access* **2019**, *7*, 58541–58552. [CrossRef]
16. Rosić, M.B.; Simić, M.I.; Pejović, P.V. An improved adaptive hybrid firefly differential evolution algorithm for passive target localization. *Soft Comput.* **2021**, *25*, 5559–5585. [CrossRef]
17. Lalama, Z.; Boulfekhar, S.; Semechedine, F. Localization Optimization in WSNs Using Meta-Heuristics Optimization Algorithms: A Survey. *Wirel. Pers. Commun.* **2022**, *122*, 1197–1220. [CrossRef]
18. Mohamed, A.W.; Almazyad, A.S. Differential evolution with novel mutation and adaptive crossover strategies for solving large scale global optimization problems. *Appl. Comput. Intell. Soft Comput.* **2017**, *2017*, 7974218. [CrossRef]
19. Zhang, Y.; Wu, Y.I. Multiple sources localization by the WSN using the direction-of-arrivals classified by the genetic algorithm. *IEEE Access* **2019**, *7*, 173626–173635. [CrossRef]
20. Zhang, Y.; Wang, S.; Ji, G. A comprehensive survey on particle swarm optimization algorithm and its applications. *Math. Probl. Eng.* **2015**, *2015*, 931256. [CrossRef]
21. Arora, S.; Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. *Soft Comput.* **2019**, *23*, 715–734. [CrossRef]
22. Storn, R.; Price, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
23. Cheng, J.; Xia, L. An effective Cuckoo search algorithm for node localization in wireless sensor network. *Sensors* **2016**, *16*, 1390. [CrossRef]
24. Arora, S.; Singh, S. An Effective Hybrid Butterfly Optimization Algorithm with Artificial Bee Colony for Numerical Optimization. *Int. J. Interact. Multimed. Artif. Intell.* **2017**, *4*, 14–21. [CrossRef]
25. Devika, E.; Saravanan, A. Enhanced gray wolf optimization for estimation of time difference of arrival in WSNs. *Int. J. Pervasive Comput. Commun.* **2022**. [CrossRef]
26. Xiong, G.; Zhang, J.; Yuan, X.; Shi, D.; He, Y.; Yao, G. Parameter extraction of solar photovoltaic models by means of a hybrid differential evolution with whale optimization algorithm. *Sol. Energy* **2018**, *176*, 742–761. [CrossRef]
27. Wu, P.; Su, S.; Zuo, Z.; Guo, X.; Sun, B.; Wen, X. Time Difference of Arrival (TDoA) Localization Combining Weighted Least Squares and Firefly Algorithm. *Sensors* **2019**, *19*, 2554. [CrossRef]
28. Opara, K.R.; Arabas, J. Differential Evolution: A survey of theoretical analyses. *Swarm Evol. Comput.* **2019**, *44*, 546–558. [CrossRef]
29. Arora, S.; Singh, S. Node localization in wireless sensor networks using butterfly optimization algorithm. *Arab. J. Sci. Eng.* **2017**, *42*, 3325–3335. [CrossRef]
30. Tan, L.S.; Zainuddin, Z.; Ong, P. Wavelet neural networks based solutions for elliptic partial differential equations with improved butterfly optimization algorithm training. *Appl. Soft Comput.* **2020**, *95*, 106518. [CrossRef]
31. Najarro, L.A.C.; Song, I.; Kim, K. Differential evolution with opposition and redirection for source localization using RSS measurements in wireless sensor networks. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1736–1747. [CrossRef]
32. Deng, W.; Shang, S.; Cai, X.; Zhao, H.; Song, Y.; Xu, J. An improved differential evolution algorithm and its application in optimization problem. *Soft Comput.* **2021**, *25*, 5277–5298. [CrossRef]
33. Zhang, J.; Sanderson, A.C. JADE: Adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.* **2009**, *13*, 945–958. [CrossRef]

34. Tanabe, R.; Fukunaga, A. Success-history based parameter adaptation for differential evolution. In Proceedings of the Evolutionary Computation (CEC), Cancun, Mexico, 20–23 June 2013; pp. 71–78.
35. Brest, J.; Greiner, S.; Boskovic, B.; Mernik, M.; Zumer, V. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.* **2006**, *10*, 646–657. [CrossRef]
36. Arora, S.; Singh, S. An improved butterfly optimization algorithm with chaos . *J. Intell. Fuzzy Syst.* **2017**, *32*, 1079–1088. [CrossRef]
37. Fan, Y.; Shao, J.; Sun, G.; Shao, X. A self-adaption butterfly optimization algorithm for numerical optimization problems. *IEEE Access* **2020**, *8*, 88026–88041. [CrossRef]
38. Li, G.; Shuang, F.; Zhao, P.; Le, C. An improved butterfly optimization algorithm for engineering design problems using the cross-entropy method. *Symmetry* **2019**, *11*, 1049. [CrossRef]
39. Aydilek, I.B. A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems. *Appl. Soft Comput.* **2018**, *66*, 232–249. [CrossRef]
40. Zhang, M.; Long, D.; Qin, T.; Yang, J. A Chaotic Hybrid Butterfly Optimization Algorithm with Particle Swarm Optimization for High-Dimensional Optimization Problems. *Symmetry* **2020**, *12*, 1800. [CrossRef]
41. Zhou, H.; Zhang, G.; Wang, X.; Ni, P.; Zhang, J. A hybrid identification method on butterfly optimization and differential evolution algorithm. *Smart Struct. Syst.* **2020**, *26*, 345–360.
42. Yang, D.; Li, G.; Cheng, G. On the efficiency of chaos optimization algorithms for global optimization. *Chaos Solitons Fractals* **2007**, *34*, 1366–1375. [CrossRef]
43. Thompson, J.M.T.; Stewart, H.B. *Nonlinear Dynamics and Chaos*; John Wiley & Sons: Hoboken, NJ, USA, 2002.
44. Tavazoei, M.S.; Haeri, M. Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. *Appl. Math. Comput.* **2007**, *187*, 1076–1085. [CrossRef]
45. Zhenyu, G.; Bo, C.; Min, Y.; Binggang, C. Self-adaptive chaos differential evolution. In Proceedings of the International Conference on Natural Computation, Xi'an, China, 24–28 September 2006; pp. 972–975.
46. Mandal, S.; Mandal, K.K. Optimal energy management of microgrids under environmental constraints using chaos enhanced differential evolution. *Renew. Energy Focus* **2020**, *34*, 129–141. [CrossRef]
47. Tian, D.; Zhao, X.; Shi, Z. Chaotic particle swarm optimization with sigmoid-based acceleration coefficients for numerical function optimization. *Swarm Evol. Comput.* **2019**, *51*, 100573. [CrossRef]
48. Li, W.; Wei, P.; Xiao, X. A robust TDOA-based location method and its performance analysis. *Sci. China Ser. F Inf. Sci.* **2009**, *52*, 876–882. [CrossRef]
49. Shen, J.; Molisch, A.F.; Salmi, J. Accurate passive location estimation using TOA measurements. *IEEE Trans. Wirel. Commun.* **2012**, *11*, 2182–2192. [CrossRef]
50. Hu, Y.; Leus, G. Robust differential received signal strength-based localization. *IEEE Trans. Signal Process.* **2017**, *65*, 3261–3276. [CrossRef]
51. Xu, S.; Doğançay, K. Optimal sensor placement for 3-D angle-of-arrival target localization. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 1196–1211. [CrossRef]
52. Tomic, S.; Beko, M.; Dinis, R.; Bernardo, L. On target localization using combined RSS and AoA measurements. *Sensors* **2018**, *18*, 1266. [CrossRef]
53. Xiao, H.; Zhang, H.; Wang, Z.; Gulliver, T.A. An RSSI based DV-hop algorithm for wireless sensor networks. In Proceedings of the 2017 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), Shanghai, China, 21–25 September 2017; pp. 1–6.
54. Shen, J.; Wang, A.; Wang, C.; Hung, P.C.; Lai, C.F. An efficient centroid-based routing protocol for energy management in WSN-assisted IoT. *IEEE Access* **2017**, *5*, 18469–18479. [CrossRef]
55. Liu, J.; Wang, Z.; Yao, M.; Qiu, Z. VN-APIT: Virtual nodes-based range-free APIT localization scheme for WSN. *Wirel. Netw.* **2016**, *22*, 867–878. [CrossRef]
56. Halder, S.; Ghosal, A. A survey on mobile anchor assisted localization techniques in wireless sensor networks. *Wirel. Netw.* **2016**, *22*, 2317–2336. [CrossRef]
57. Chen, J.; Zhao, Y.; Zhao, C.; Zhao, Y. Improved two-step weighted least squares algorithm for TDOA-based source localization. In Proceedings of the 2018 19th International Radar Symposium (IRS), Bonn, Germany, 20–22 June 2018; pp. 1–6.
58. Lin, L.; So, H.C.; Chan, F.K.; Chan, Y.T.; Ho, K. A new constrained weighted least squares algorithm for TDOA-based localization. *Signal Process.* **2013**, *93*, 2872–2878. [CrossRef]
59. Biswas, P.; Liang, T.C.; Toh, K.C.; Ye, Y.; Wang, T.C. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Trans. Autom. Sci. Eng.* **2006**, *3*, 360–371. [CrossRef]
60. Cakir, O.; Kaya, I.; Yazgan, A.; Cakir, O.; Tugcu, E. Emitter location finding using particle swarm optimization. *Radioengineering* **2014**, *23*, 252–258.
61. Meng, Y.; Zhi, Q.; Zhang, Q.; Yao, N. A Two-Stage Particle Swarm Optimization Algorithm for Wireless Sensor Nodes Localization in Concave Regions. *Information* **2020**, *11*, 488. [CrossRef]
62. Li, Q.; Chen, B.; Yang, M. Time Difference of Arrival Passive Localization Sensor Selection Method Based on Tabu Search. *Sensors* **2020**, *20*, 6547. [CrossRef]
63. Díez-González, J.; Álvarez, R.; González-Bárcena, D.; Sánchez-González, L.; Castejón-Limas, M.; Perez, H. Genetic algorithm approach to the 3D node localization in TDOA systems. *Sensors* **2019**, *19*, 3880. [CrossRef]

64. Srideviponmalar, P.; Jawahar Senthil Kumar, V.; Harikrishnan, R. Hybrid Genetic Algorithm–Differential Evolution Approach for Localization in WSN. In *Intelligent Engineering Informatics*; Springer: Cham, Switzerland, 2018; pp. 263–271.
65. Rosić, M.B.; Simić, M.I.; Pejović, P.V. Passive target localization problem based on improved hybrid adaptive differential evolution and Nelder-Mead algorithm. *J. Sens.* **2020**, *2020*, 3482463 . [CrossRef]
66. Harikrishnan, R.; Kumar, V.J.S.; Ponmalar, P.S. A Comparative Analysis of Intelligent Algorithms for Localization in Wireless Sensor Networks. *Wirel. Pers. Commun.* **2016**, *87*, 1057–1069. [CrossRef]
67. Gumaida, B.F.; Luo, J. A hybrid particle swarm optimization with a variable neighborhood search for the localization enhancement in wireless sensor networks. *Appl. Intell.* **2019**, *49*, 3539–3557. [CrossRef]
68. Su, C.; Liu, Y.; Liu, L.; Yang, M.; Zhao, H.; Yin, X. Experimental Evaluation of Multipath Mitigation in TDOA-Based Indoor Passive Localization System Using A Beam Steering Broadband Circular Polarization Antenna. *Electronics* **2018**, *7*, 362. [CrossRef]
69. Kwon, S.; Choi, Y.; Moon, S.; You, C.; Liu, H.; Kim, J.H.; Kim, D.J.; Park, H.; Kim, J.Y.; Hwang, I. Performance enhancement of hybrid TDOA/AOA using multipath delay estimation. *Wirel. Pers. Commun.* **2020**, *115*, 2551–2568. [CrossRef]
70. Tsai, M.H.; Luo, J.N.; Yang, M.H.; Lo, N.W. Location Tracking and Forensic Analysis of Criminal Suspects' Footprints. In Proceedings of the 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT), Kahului, HI, USA, 14–17 March 2019; pp. 210–214.
71. Ivanov, S.; Kuptsov, V.; Badenko, V.; Fedotov, A. RSS/TDoA-Based Source Localization in Microwave UWB Sensors Networks Using Two Anchor Nodes. *Sensors* **2022**, *22*, 3018. [CrossRef] [PubMed]
72. Wan, P.; Ni, Y.; Hao, B.; Li, Z.; Zhao, Y. Passive localization of signal source based on wireless sensor network in the air. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 1550147718767371. [CrossRef]
73. Wu, G.; Shen, X.; Li, H.; Chen, H.; Lin, A.; Suganthan, P.N. Ensemble of differential evolution variants. *Inf. Sci.* **2018**, *423*, 172–186. [CrossRef]
74. Zhou, Y.Z.; Yi, W.C.; Gao, L.; Li, X.Y. Adaptive differential evolution with sorting crossover rate for continuous optimization problems. *IEEE Trans. Cybern.* **2017**, *47*, 2742–2753. [CrossRef]
75. Mohamed, A.W. An improved differential evolution algorithm with triangular mutation for global numerical optimization. *Comput. Ind. Eng.* **2015**, *85*, 359–375. [CrossRef]
76. Qian, W.; Chai, J.; Xu, Z.; Zhang, Z. Differential evolution algorithm with multiple mutation strategies based on roulette wheel selection. *Appl. Intell.* **2018**, *48*, 3612–3629. [CrossRef]
77. Liang, J.J.; Qu, B.Y.; Suganthan, P.N. Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. *Comput. Intell. Lab. Zhengzhou Univ. Zhengzhou China Tech. Rep. Nanyang Technol. Univ. Singap.* **2013**, *635*, 490.
78. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [CrossRef]