sensors

MDPI

*Article*

# Improved Location Estimation in Wireless Sensor Networks Using a Vector-Based Swarm Optimized Connected Dominating Set

**Gulshan Kumar [1,*], Rahul Saha [1], Mritunjay Kumar Rai [2], Reji Thomas [3], Tai-Hoon Kim [4], Se-Jung Lim [5] and Jai Sukh Paul Singh [2]**

[1] Department of Computer Science and Engineering, Lovely Professional University, Jalandhar-Delhi GT Road, Phagwara 144 441, India; rsahaaot@gmail.com
[2] Department of Electronics and Electrical Engineering, Lovely Professional University, Jalandhar-Delhi GT Road, Phagwara 144 441, India; raimritunjay@gmail.com (M.K.R.); khalsa128@gmail.com (J.S.P.S.)
[3] Division of Research and Development, Lovely Professional University, Jalandhar-Delhi GT Road, Phagwara 144 441, India; rthomas.eyyalil@gmail.com
[4] Department of Convergence Security, Sungshin Women's University, Seoul 02844, Korea; taihoonn@daum.net
[5] Department of Computer Engineering, Chonnam National University, 50 Daehak-ro, Yeosu 59626, Jeollanam-do, Korea; limsejung@korea.ac.kr
* Correspondence: gulshan3971@gmail.com; Tel.: +91-814-652-4362

Received: 12 November 2018; Accepted: 12 January 2019; Published: 17 January 2019

**Abstract:** Location estimation in wireless sensor networks (WSNs) has received tremendous attention in recent times. Improved technology and efficient algorithms systematically empower WSNs with precise location identification. However, while algorithms are efficient in improving the location estimation error, the factor of the network lifetime has not been researched thoroughly. In addition, algorithms are not optimized in balancing the load among nodes, which reduces the overall network lifetime. In this paper, we have proposed an algorithm that balances the load of computation for location estimation among the anchor nodes. We have used vector-based swarm optimization on the connected dominating set (CDS), consisting of anchor nodes for that purpose. In this algorithm, major tasks are performed by the base station with a minimum number of messages exchanged by anchor nodes and unknown nodes. The simulation results showed that the proposed algorithm significantly improves the network lifetime and reduces the location estimation error. Furthermore, the proposed optimized CDS is capable of providing a global optimum solution with a minimum number of iterations.

**Keywords:** localization; accuracy; load balance; swarm optimization; network lifetime

## 1. Introduction

Wireless sensor networks (WSNs) are widely used in observing, agriculture, home, and defence environments that are dynamic. Emerging information technology and application requirements have shifted the paradigm of a static network scenario to dynamic environments of WSNs. The mobility of nodes (anchor and unknown) in the network makes it difficult to estimate the location of an unknown node. The accuracy of the location of nodes in such networks is of great importance and depends on location estimation and verification [1,2]. The major drawback of WSNs is the lifetime of the network, as the nodes have limited resources to power up. Compared to unknown nodes, anchor nodes are more privileged, with better power and computational resources. However, unknown nodes drain their resources faster to perform heavy computational tasks. Therefore, the network becomes partially unavailable until the exhausted nodes are replaced. However, the replacement process is not always feasible in the dynamic configuration settings of the applications. Moreover, WSNs use a virtual backbone to prolong the network lifetime [3–5]. Therefore, the localization process has to be efficient in the terms of minimizing resource utilization. Such efficient utilization of resources is achieved by optimization [6–8] and load balancing [9] on the virtual backbones.

In this paper, we introduce an approach by constructing a connected dominating set (CDS) based virtual backbone only with anchor nodes optimized in terms of load balance using a vector-based swarm optimization (VBSO) mechanism [10]. This optimized CDS is then used for location estimation in WSNs. The purpose of including CDS in the proposed algorithm is to use the anchor nodes properly and to increase the overall network lifetime. The remaining paper is organized as follows. Section 2 reviews the related work in this field. The network model for the algorithm is explained in Section 3. The proposed algorithm is shown in Section 4. The detailed process of the applied optimization is explained in Section 5, and the localization details are discussed in Section 6. The results are explained in Section 7. Finally, conclusions are drawn in Section 8.

## 2. Related Work

Researchers have recently considered various algorithms to improve the lifetime of nodes without compromising their location estimation accuracy. Localization techniques are broadly categorized into (i) direct and (ii) indirect approaches. The former is based on manual configuration and GPS-based techniques and the latter on range-based and range-free techniques [1]. Location identification of nodes with range-based approaches uses different methods, such as time difference of arrival (TDoA), angle of arrival (AoA), time of arrival (ToA), and a received signal strength indicator (RSSI). On the other hand, range-free approaches use region inclusion, hop count, and neighbor location identification methods that avoid a direct relationship between point-to-point distances and are, therefore, more economic. Some hybrid approaches, including RSS–AoA and RSS–ToA, have been discussed in References [11–14]. These hybrid approaches provide generality by applying data fusion and are also used for global synchronization with high accuracy but lag behind in computational cost, special hardware requirements, and energy consumption. Another hybrid localization algorithm is shown in Reference [15] that uses message passing and approximation of posterior distribution of targets. It also uses Bayesian distribution [16,17]. These techniques either use range or angle as their measurement parameter and geometrical interpretations (such as triangulation or trilateration) are calculated accordingly. This seems easier to apply in the absence of noise. Moreover, the problem with such a geometric-based calculation is the existence of a larger intersection area of hyperbolas rather than a point. Therefore, to improve the efficiency of the localization techniques in the presence of noise, the two measurements—range and angle—are integrated as hybrid approaches. The generic approaches of range-free methods were recently further modified for their economic value and are discussed hereafter. Energy-efficient load- balanced clustering (EELBC) is one of the range-free approaches for localization [18]. This algorithm uses a minimum heap of cluster heads (CHs) or gateway nodes and the nodes are assumed to be aware of the position through GPS. Additionally, network setup is performed in two phases, bootstrapping and clustering. In the clustering phase, the sink executes the clustering algorithm depending on the minimum heap algorithm. The extra cost of GPS makes this algorithm less attractive.

Another variant of the load-balanced cluster-based algorithm creates clusters based on distances and the distribution density of the nodes [19]. Here, node distribution follows Poisson's process and has fixed locations and, hence, mobility is not considered. This algorithm lacks load balancing factors. In another range-free approach, three NP-hard problems: the min–max degree maximal independent set (MDMIS), the load-balanced virtual backbone (LBVB), and the min–max valid-degree nonbackbone node allocation (MVBA) were considered [20]. An additional approximation algorithm was also proposed in the method which uses the linear relaxing and random rounding techniques. Genetic-based approaches have been discussed in References [21–24]. One variant of the genetic algorithm using a simplex method is shown in Reference [25]. This simplex method is used to optimize the error minimization problem in widely used DV-hop for searching a local optima. The use of the multiobjective genetic algorithm to create a load-balanced virtual backbone is shown in Reference [26]. The mobility of the network increases the iterations of the algorithm to get a local optimum solution, making the overall approach disadvantageous. The identification of balanced nodes in wireless sensor networks for data aggregation has also been shown with a probabilistic network model for a data aggregation tree [27]. Moreover, an algorithm for the identification of a load-balanced maximal dominator node Set (LBMDS) and connector nodes for LBMDS have been developed. An expected allocation probability algorithm (EAP) to solve parent node assignment (PNA) has been proposed. A novel distributed tracking process of the sensor nodes has been introduced [28]. In this approach, a local node using a fuzzy system provides a partial solution, with a centralized algorithm that merges all the partial solutions. The centralized algorithm is based on the calculation of the centroid of the partial solutions. The deviation of the partial solutions increases the complexity of the algorithm, which is an identified concern in this approach.

The solution for swarm configuration-based localization uses the min–max method and swarm optimization [29]. Swarm uses a sequential Monte Carlo localization method to consider mobility and does not require additional hardware. As it can work with seed movement, the mobility factor lowers the cost of implementation without compromising accuracy. We have considered this approach as a candidate, as well for comparison. An ant colony-based optimization approach is described in the literature [30]. A distributed method for parametric regression in a clustered WSN using particle swarm optimization was also introduced recently with multiobjective optimization (MO) [31]. Moreover, the same as the vector-evaluated particle swarm optimization method, it considers the MO problem in two phases; the algorithm obtains a set of candidate network regressors and computes the final model using a weighted averaging rule. However, this parametric regression is not suitable for real time-based scenarios with the mobility effect. A novel optimized localization method using glowworm swarm optimization (GSOL) has been introduced to overcome this drawback [32]. The fitness function of this optimization process follows the deviation or error calculation with the Euclidean distance aspect. After the multi-iterations, all the mobile nodes within the network concentrates around one or multiple nodes providing local optimal location create the global optimal location. We have considered this algorithm also as another candidate for comparison due to its claimed efficiency. Some other techniques for the optimization and load balancing the virtual backbone include: Neural networks [33], the machine learning approach [34], and the differential evolution approach [35].

The aforementioned short survey on the range-free optimization algorithms in WSNs has some drawbacks: (i) The algorithms are not load-balanced and, therefore, partial exhaustion of the network exists; (ii) the redundancy of the clustering nodes makes the process complex; (iii) heavy functioning load on some nodes lowers the network lifetime; and (iv) validation for the overall mobile (unknown and anchor: Both are mobile) environment is not ensured. The present work aims to provide a solution for the abovementioned deficiencies by developing an approach of localization using a VBSO-based CDS construction.

## 3. Network Scenario

The present research work considers that the $n$ unknown nodes are located within a plane of $M \times M$ area with true locations defined by $u_i = (x_i, y_i)$ and estimated locations are defined by $\bar{u_i} = (\bar{x_i}, \bar{y_i}), \forall i = 1, 2, .., n$. We have also considered $m$ anchor nodes with known locations defined by $a_j = (x_j, y_j), \forall j = 1, 2, .., m$.

The transmission area is considered to be a circle. The centre of the circle consists of the corresponding node itself. The number of unknown nodes connected to an anchor node is variable at different time intervals and, therefore, we have calculated that the expected number of connectivity for each anchor node in the network can be given as:

$$\zeta = \frac{n}{M^2}\pi r^2. \tag{1}$$

The network of $n$ unknown nodes and $m$ anchor nodes along with their connectivity is considered as a graph $G(V,E)$. $G^{'}(V^{'},E^{'}) \subseteq G(V,E)$, where $G(V,E)$ is the original sensor network scenario and $G^{'}\ (V^{'},E^{'})$ consists only of the optimized backbone anchor nodes considered as optimized CDS and its links to the unknown nodes $u_1, u_2, ..., u_n \in U$, the set of unknown nodes, such that $\forall u_i, a_j \in V^{'}$ $\{i = 1,2,..,n$ and $j = 1,2,..,m\}$. All the unknown nodes in the set $U$ are the one-hop neighbors of any anchor node present in the CDS such that $D \subseteq A$, the set of anchor nodes $\{a_1, a_2, .., a_m\}$, $V^{'} \subseteq V$ and $V^{'} = D \cup U$, where $D$ is the dominating set.

The proposed network model is mobile in nature for both the anchor nodes and unknown nodes. Relative mobility is required to be calculated to check the mobility direction and the relative distances. The relative mobility between an unknown node $u_i$ with respect to anchor node $a_j$ at a given time $t$ is given by:

$$RM_t^{a,u} = d_{a,u_t} - d_{a,u_{t-1}}, \tag{2}$$

where $d_{a,u_t} = \frac{k}{\sqrt{P_r}}$ and $P_r$ is the power received by the receiving antenna and is calculated by the Friis space model [36]. $k$ is constant here. $d_{a,u_t}$ is used to calculate the distance and closeness between two nodes. $RM_t^{a,u}$ is positive if node $u$ is moving away from $a$ and negative if $u$ is coming closer to $a$. This relative mobility effects the CDS generation and further the related vectors. This relative mobility is used to generate the updated distance matrices for location estimation.

## 4. Vector-Based Swarm Optimized CDS

Our proposed algorithm considers the following assumptions:

- The anchor nodes are preinstalled with their own location, also known to the base station (BS).
- The base station can control all the anchor and unknown nodes. Multiple base stations are allowed, depending on the range of communication.
- Unknown nodes and anchor nodes have random mobility.
- According to the privilege concern, BS is the most privileged in terms of storage and resource. Anchor nodes are more privileged, compared to unknown nodes. Therefore, the maximum computing tasks are controlled by BS; anchor nodes and unknown nodes are kept only with a minimum message exchange process. One BS broadcasts the aggregated information to other BS.

The proposed method starts with broadcasting by anchor nodes which have more privileges than the resource constrained unknown nodes. The anchor nodes broadcast the neighbor discovery packet (NDP) as per the following format.

$$NDP : \{A\_addr, Leash\_C = 1, Seq, Accept\_S = 0\},$$

where $A\_addr$ is the source address of the anchor node, $Leash\_C$ is set to 1 to bound the packet with a limited transmission of one-hop nodes, Seq is the sequence number of the NDP to avoid void routing and stale of packets, and $Accept\_S$ is the status of the acceptance of the NDP by an unknown node set at 0. It may happen that in this process of broadcast, another anchor node receives the NDP, in which case, the receiving anchor node will simply discard the NDP. Once this NDP is received by the unknown nodes who are in the one-hop neighborhood of the sender, the unknown nodes reply with a reply neighbor set (RPNS) by decrementing the leash value by 1 and changing the acceptance status to 1. The RPNS is composed of the following components.

$$RPNS : \{SrcAddr, Leash\_C = 0, Seq, Accept\_S = 1, U\_addr\} | U\_addr\ is\ the\ address\ of\ unknown\ node.$$

Upon receiving the RPNS, the anchor nodes enlist the $U\_addr$ as its one-hop neighbor in a neighbor set list (NSL) and send it to the base station (BS) with cryptographic services so that the list cannot be manipulated by any third party attack or compromised insider node.

$$Anchor \rightarrow BS : NSL = \{A\_addr|U\_addr_1, U\_addr_2, ...\}$$

After receiving all the NSLs from the anchor nodes, BS calculates the maximum degree anchor node and enlists the anchor node in an empty dominating set D. This operation iterates until at least three anchor nodes are in the dominating set and all the unknown nodes are a one-hop neighbor of at least one of the anchor nodes in the set of anchor nodes A. The process outputs a connected dominating set (CDS), but the problem is the anchor nodes are unbalanced in terms of number connections with unknown nodes. To solve this problem, swarm optimization is performed on the CDS. BS creates binary vector $V_j$ for each of the anchor nodes present in the CDS and vector-based swarm optimization is applied for the best solution to achieve an optimized CDS for the backbone. All these $n$-bit binary vectors (as number of unknown nodes is $n$) create the initial population of swarm. Fitness value is calculated on each of these vectors and direct and indirect cooperation is calculated. After mutation on the initial population of vectors, boundary values are checked and selection starts again for the next iteration. If the convergence condition is met successfully, we get the output of the best-fit solution for our purpose. Once we get the optimized dominating set, trilateration is performed to get the location of unknown nodes. The algorithm is summarized in Algorithm 1.

**Algorithm 1** Proposed Localization Algorithm

```
procedure ANCHOR NODES $a_j$ AND UNKNOWN NODES $u_i$
    Initialize $\{A\} = \{a_1, a_2, ..., a_m\}$ and $\{U\} = \{u_1, u_2, ......, u_n\}$ and dominating set $D = \emptyset$
    for Broadcast by anchor node $a_j$ do
        Send $NDP : \{A\_addr, Leash\_C = 1, Seq, Accept\_S = 0\}$
    end for
    for On receiving, node $u_i$ computes do
        Leash_C = Leash_C - 1;
        Accept_S = 1;
        Send $RPNS : \{A\_addr, Leash\_C, Seq, Accept\_S = 1, U\_addr\}$
    end for
    if $Leash\_C == 0$ && $Accept\_S == 1$ then
        Create $NSL$
        $a_j \rightarrow BS : NSL = \{A\_addr|U\_addr_1, U\_addr_2, ......\}$
    else
        exit
    end if
    BS performs:
    while True do
        if $\forall\ u_i \in (\exists\ N\ (a_j)$ && $|D| \geq 3)$ then
            Break while loop;
        else
            Calculate the maximum degree anchor nodes $a_{max}$
            $D = D \cup a_{max}$
        end if
    end while
    $m' = |D|$
    for k=1 to m′ do
        BS creates binary vector $\vee_j, \forall_j = 1, 2, ..., m'$
        Apply $VBSO$ algorithm
        Apply Trilateration
    end for
end procedure
```

## 5. Optimization Process with VBSO

To optimize the obtained CDS, we have applied vector-based swarm optimization (VBSO) on the obtained CDS. The solution for optimizing the CDS exhibits the features of the maximization problem. To solve this problem, we considered the connections of anchor nodes in the CDS with their corresponding one-hop neighboring unknown nodes as $n$-dimensional vector $V_j$. The proposed method provides a global maximization solution, which is formulated as a pair $(V_j, f)$, where $V_j \subseteq R^n$ is a bounded set on $R^n$ and $f : V_j \to R$ is $n$ dimensional real valued function. The domain $R^n$ of function $f$ is considered the search space. We have to find a point $\hat{x} \in V_j$ so that $f(\hat{x})$ is a global maximum on $V_k$. Thus, each element $x$ of $R^n$ is considered a candidate solution in the defined search space with $\hat{x}$ being optimal such that $f(\hat{x}) \leq f(x)$. The optimization process for CDS with VBSO algorithm has been summarized in Algorithm 2.

### *5.1. Initialization of Vectors*

Each unknown node in the network is allocated to one of the anchor nodes in the CDS. Each anchor node $a_j$ and its connections with one-hop neighboring unknown nodes $u_i$ is represented by an n-dimensional binary vector $V_j, \forall j = 1, 2, .., m^{'}$, where $m^{'}$ is the number of anchor nodes in CDS and $m^{'} \subseteq m$.

Let us consider the scenario shown in Figure 1. The scenario consists of five anchor nodes and twelve unknown nodes. The blue circles represent anchor nodes, green circle nodes represent unknown nodes, blue solid lines represent a CDS and green solid lines represent the connection between an anchor node of CDS and unknown node in the network. As per our algorithm, we require at least three anchor nodes in the CDS to calculate a location with the trilateration process. Therefore, anchor nodes 1, 2, and 3 create the required CDS for the purpose for which the vector representation (V1, V2, and V3) is shown in Figure 2. These vectors are considered parent vectors for the VBSO algorithm. The vector representation for each anchor node is formulated as:

$$V_{j_i} = \begin{cases} 1, & if\ anchor\ a_j\ is\ one\ hop\ neighbour\ of\ u_i \\ 0, & else \end{cases} \quad \forall\, i = 1, 2, .., n\ and\ \forall\, j = 1, 2, .., m^{'} \tag{3}$$

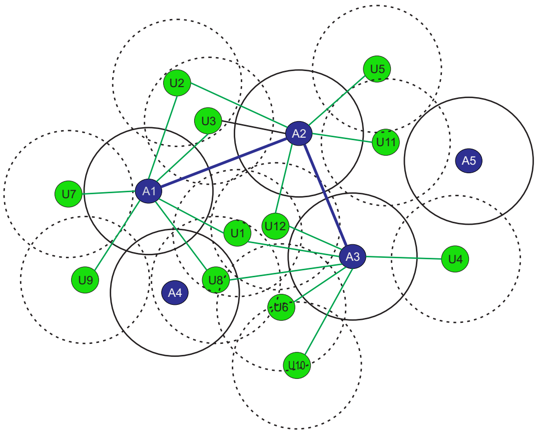


**Figure 1.** Example scenario.

| | $U_1$ | $U_2$ | $U_3$ | $U_4$ | $U_5$ | $U_6$ | $U_7$ | $U_8$ | $U_9$ | $U_{10}$ | $U_{11}$ | $U_{12}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| V₁ | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| V₂ | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| V₃ | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

**Figure 2.** Vector representation for connected dominating set (CDS) shown in Figure 1.

### *5.2. Calculate the Fitness Vector*

The formulation of the fitness function is a critical and important task in the optimization process, as it helps for fast convergence with the optimum solution. To formulate the fitness function, we have used two parameters, which are also considered in our previous work [37].

Dominating set $\theta$ value for the connected dominating set D given as: $|CDS_\theta| = \left(\sum_{j=1}^{m'} |deg_j - d\bar{e}g|^{\theta}\right)^{\frac{1}{\theta}}$, where $deg_j$ is the degree of each dominator anchor node in the set $D$ and $d\bar{e}g$ is the mean of all the degrees of the dominator in the set $D$. The calculation of allocation scheme value: $\mid Al_\theta| = \left(\sum_{j=1}^{m'} |deg_j' - \zeta|^{\theta}\right)^{\frac{1}{\theta}}$ considers that there are $m'$ disjointed sets for $m'$ anchor nodes present in the set $D$, such that $N(a_j) \cap N(a_k) = \emptyset$ and $\forall u_i \in N(a_j), 1 \leq j \leq m'$ so that $(u_i, a_i) \in E'$. $N()$ here represents the neighborhood. $deg_i'$ is the valid degree of an anchor node $a_j$, i.e., the number of unknown nodes already allocated to the anchor node $a_j$.

The objective of the problem defined in this paper is to identify an optimized CDS with minimum $D_\theta$ and $Al_\theta$ values. Therefore, the fitness function can be given as:

$$\begin{cases} f(V_j) = \dfrac{n - |D|}{\mathcal{W}_1|CDS_\theta| + \mathcal{W}_2|Al_\theta|} \\ \mathcal{W}_1 + \mathcal{W}_2 = 1, \textit{such that } 0 < \mathcal{W}_1, \mathcal{W}_2 < 1 \end{cases} \tag{4}$$

The linear combination of the two metrics $CDS_\theta$ and $Al_\theta$ is made to adjust the bias factors $\mathcal{W}_1$ and $\mathcal{W}_2$, according to the network environment. The above equation emphasizes that to maximize the fitness function for achieving optimum value, the denominator needs to be minimized and the numerator needs to be maximized so that the overall fitness function will be maximized accordingly. The experimented value of $\theta$ = 2, as it is analogous for an electro static field.

### *5.3. Reproduction in VBSO*

The reproduction method in the VBSO algorithm primarily depends on creating cooperation vectors. Multiple vectors in the search space are combined with vector operations to achieve the cooperation vector.

The direct cooperation vector and differential cooperation vectors are used to create a cooperation vector. The direct cooperation vector at the kth iteration is given by:

$$V_{dir} = w_1 \cdot V_x + w_2 \cdot V_y + w_3 \cdot V_z + w_4 \cdot V_b + w_5 \cdot V_r, \tag{5}$$

where $V_x$,$V_y$,$V_z$ belongs to the initial population vectors, $V_b$ is the best vector in the neighborhood, and $V_r$ is the random vector and $w_1$ + $w_2$ + $w_3$ + $w_4$ + $w_5$ =1. These values are called cooperative weighting coefficients (CWC). Figure 3 represents a direct cooperation vector with $V_x$ and $V_b$ which actually outputs the global optimum (GO). Following the same line, Figure 4 represents another direct cooperation vector with $V_x$, $V_y$, $V_b$m and $V_r$.

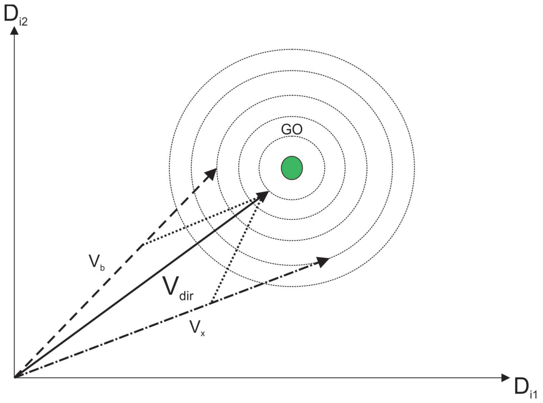


**Figure 3.** Global optima with $V_x$ and $V_b$.

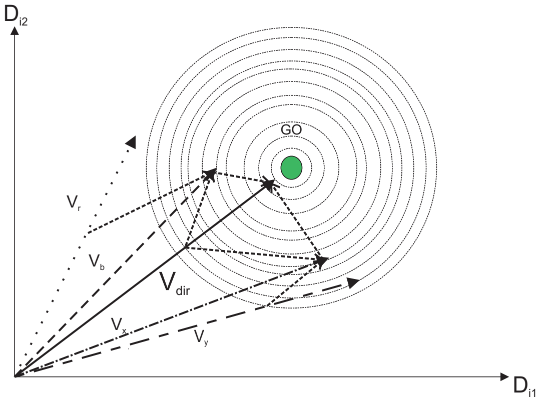


**Figure 4.** Direct cooperation vector.

For the scenario of more than three initial population vectors, the selection of three among them is another crucial part which depends on higher fitness values and with a higher probability of selection. The probability is therefore given as:

$$P(V_j) = \frac{f(V_j)}{\sum_{j=1}^{m'} f(V_j)}. \tag{6}$$

The differential cooperation vector signifies the search around direct cooperation vector. It helps the algorithm to pass local optimums and converge to global one, as shown in Figure 5. Direct cooperation vectors provide a local solution, whereas the differential cooperation vector uses these direct vectors to get the global optimal solution, which is as good or better than all the other feasible solutions. To obtain the unknown global optimum, we have followed $k$ iterations, in which if two simultaneous iterations have output the same values of local optima, we have considered it the global optimum. The differential cooperation vector ($V_{diff}$) of the population is calculated by considering proper portions of differential vectors of a current solution, for example $V_c$, given as:

$$V_{diff} = w_6 \cdot (V_x - V_c) + w_7 \cdot (V_y - V_c) + w_8 \cdot (V_z - V_c) + w_9 \cdot (V_b - V_c) + w_{10} \cdot (V_r - V_c) \tag{7}$$

$$where\ w_6 + w_7 + w_8 + w_9 + w_{10} = 1 \tag{8}$$

Figure 6 shows the generation of the cooperation vector obtained from direct and differential cooperation vectors, which can be given as:

$$V_{co} = V_{diff} + V_{dir}. \tag{9}$$

In all Figures 3–7, the contour lines represent the cost function that depends on the fitness value of the vectors.

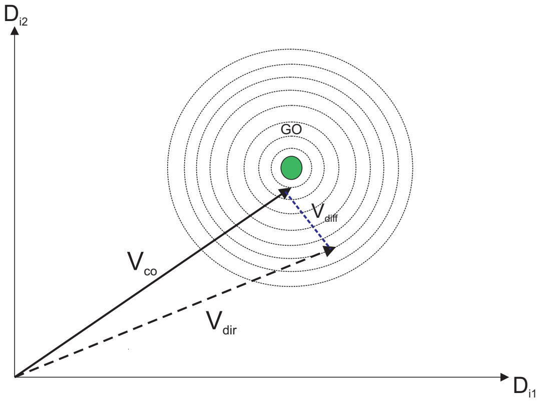


**Figure 5.** Differential cooperation vector creation.

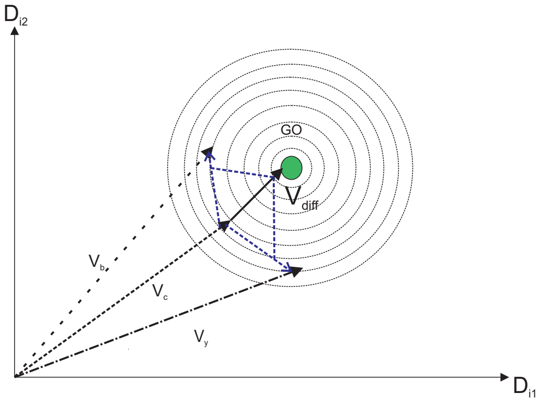


**Figure 6.** Generation of $V_{diff}$ with initial population vectors.

### 5.4. Mutation

To increase the diversity of the population, mutation is applied to the initial population vectors. A number of mutation operators have been suggested in Reference [38–40]. Applying direct and differential cooperation implicitly provide mutation in the process. In addition, a second mutation is also applied to transfer the search space origin to a point far enough from the current origin. A random mutation probability $p_m$ is selected. A random number r is generated and then compared with $p_m$. The second mutation is applied if $rand > p_m$ and $rand$ is in $(0, 1)$. In the mutation process, the random bits in a vector $V_k$ are flipped to get the mutation vector $V_m$ . The candidate offspring vector $V_{off}$ is obtained by the following equation, and also shown in Figure 7:

$$V_{off} = V_m + V_{co}. \tag{10}$$

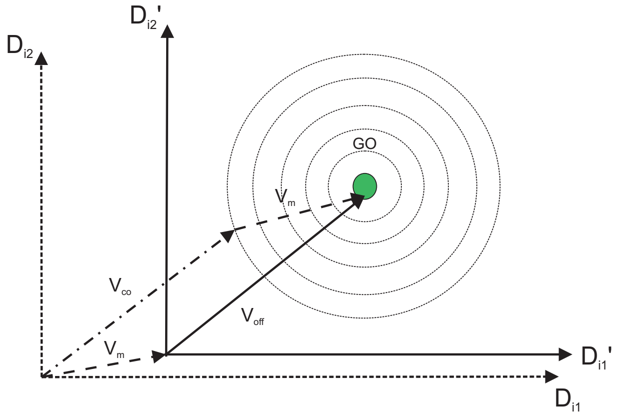


**Figure 7.** Candidate offspring generation with mutation vector.

### 5.5. Boundary Check

After the mutation, new offspring vectors are checked for the boundary condition so that all the offsprings must belong to the search space. An invalid offspring can lead to ab uncoordinated and invalidated search and the convergence will become a delay. Different methods are shown in References [41,42]. We have used the boundary check with a fitness function. If the fitness function of a vector in the current iteration is smaller than the fitness function of the best vector in the previous iteration, then replace the current unfit vector with its parent vector. The formulation of this process is given as:

$$if\ f(V_j)_i < f(V_{best})_{i-1} : f(V_{best})_i = max(f(V_{best})_{i-1},\ replace\ current\ vector\ with\ parent \qquad (11)$$

For example, considering the scenario of Figure 2, the fitness value of the best solution is 5.89 (considering $r = 2m$ and $\theta = 2$ (analogy to electro static field). After reproduction and mutation, we have got the offspring vector as in Figure 8.

| | $U_1$ | $U_2$ | $U_3$ | $U_4$ | $U_5$ | $U_6$ | $U_7$ | $U_8$ | $U_9$ | $U_{10}$ | $U_{11}$ | $U_{12}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $V_{off}$ | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

**Figure 8.** Offspring of the parent vectors shown in Figure 2.

The candidate offspring has a fitness value of 0.278, which is less than the previous best solution of 5.89. Therefore, we need to replace this candidate with one of the parents of the previous iteration by calculating the hamming distance. This vector $V_{off}$ has a hamming distance of 3, 6, and 9 from the initial parent vectors $V_1$, $V_2$, and $V_3$, respectively. Thus, this candidate offspring will be replaced by vector $V_1$ that corresponds to $A_1$. This process of retaining the parent of best solution helps to converge faster to the global optimum.

### 5.6. Selection of Offsprings

The identification of the next iteration vectors requires a selection process for suitable offspring vectors. Various methods of selection have been discussed in References [38,43,44]. In our proposed approach, all the valid candidate offsprings are directly transferred to the next iteration. The overall process of the vector-based optimization approach used in our purpose has been summarized in Algorithm 2. The convergence process of the optimization method is depicted in the form of a flowchart in Figure 9.

**Algorithm 2** VBSO Algorithm on CDS

```
1:  procedure VECTOR $\vee_j, \forall, j = 1,2,...,m'$
2:      while $f(V_j)$ does not change successive $l$ iterations do
3:          Calculate $|CDS_\theta| = (\sum_{j=1}^{m'} |deg_j - \overline{deg}|^\theta)^{\frac{1}{\theta}}$
4:          Calculate expected allocated neighbour values as: $\zeta = \frac{n}{M^2}\pi r^2$
5:          Calculate the allocation value given as: $|Al_\theta| = (\sum_{j=1}^{m'} |deg'_j - \zeta|^\theta)^{\frac{1}{\theta}}$
6:          Calculate $f(V_j) = \frac{n-|D|}{W_1|CDS_\theta|+W_2|Al_\theta|}$
7:          if $f(V_j)$ does not change in simultaneously $l$ iterations then
8:              goto Step 15
9:          else
10:             Calculate direct and indirect cooperation vectors
11:             Apply mutation
12:             Check boundary condition
13:             Selection of offsprings for next iteration
14:             goto 6
15:             return the current optimal solution
16:         end if
17:     end while
18: end procedure
```

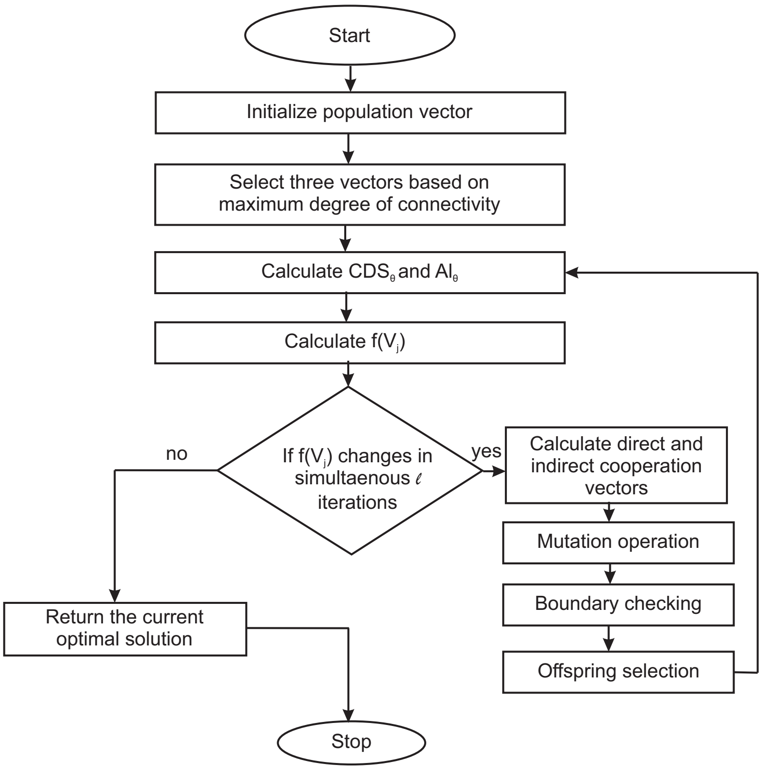

Start
Initialize population vector
Select three vectors based on maximum degree of connectivity
Calculate CDS$_\theta$ and Al$_\theta$
Calculate f(V$_j$)
If f(V$_j$) changes in simultaenous $\ell$ iterations
no
yes
Calculate direct and indirect cooperation vectors
Mutation operation
Boundary checking
Offspring selection
Return the current optimal solution
Stop

**Figure 9.** Flowchart of convergence process.

## 6. Localization of Unknown Nodes

Once the optimized CDS is achieved, we can apply any trilateration method to estimate the location of an unknown node $u_i$. The estimated distance $\bar{d_j}$ from an unknown node $u_i$ to each dominator anchor node in optimized CDS is given by the respective Euclidean distances between them, as shown below.

$$\bar{d_j}^2 = (x_j - \bar{x})^2 + (y_j - \bar{y})^2 | \, \forall \, j = 1,2,..,m', \tag{12}$$

where $(\bar{x}, \bar{y})$ is the estimated location of an unknown node of $u_i$.

The calculated distance with relative mobility becomes:

$$d_j = \bar{d}_j + RM_t^{a,u}. \tag{13}$$

The above equation for $j = 1, 2, .., m^{'}$ can be transformed into the $AX = b$ form, where:

$$A = \begin{vmatrix} (x_1 - x_{m'}) & (y_1 - y_{m'}) \\ . & . \\ . & . \\ (x_{m'-1} - x_{m'}) & (y_{m'-1} - y_{m'}) \end{vmatrix} \tag{14a}$$

$$X = \begin{vmatrix} \bar{x} \\ \bar{y} \end{vmatrix} \tag{14b}$$

$$b = \frac{1}{2} \begin{vmatrix} d_1^2 - d_{m'}^2 + d_{m'u}^2 - d_{1u}^2 & \\ d_2^2 - d_{m'}^2 + d_{m'u}^2 - d_{2u}^2 & \\ . & . \\ . & . \\ d_{m'-1}^2 - d_{m'}^2 + d_{m'u}^2 - d_{(m'-1)u}^2 & \end{vmatrix} \tag{14c}$$

Equation $AX = b$ is solvable by the standards of the least-square [45] solution to provide the value of $x$ given by:

$$\bar{x} = (A^T A)^{-1} A^T b. \tag{15}$$

The problems identified in the existing algorithms as stated in the section of related work have been addressed by our proposed solution and summarized in Table 1.

**Table 1.** Addressing problem with proposed solution.

| **Existing Problems** | **Solution in the Proposed Algorithm** |
|---|---|
| Algorithms are not load-balanced and, therefore, partial exhaustion of the network exists. | Load allocation attribute has been used with $CDS_\theta$ and $Al_\theta$. |
| Redundancy of clustering nodes make the process complex and also causes heavy functioning load on some particular nodes. | Redundancy is removed as the CDS is maintained only by anchor nodes having a maximum degree, so that other nodes can be at rest and can be used later. |
| Validation for overall mobile (sensor and anchor: Both are mobile) environment is not ensured. | In the proposed solution, we have considered all anchor nodes and sensor nodes to be mobile and therefore support full mobility of the network. |

## 7. Result and Analysis

The proposed algorithm has been simulated in NS-2 with the following parameters enlisted in the Table 2.

**Table 2.** Simulation parameters.

| | |
|---|---|
| Simulation Area | 100 × 100 m$^2$ |
| No. of unknown nodes | 100 to 200 |
| No. of anchor nodes | 10 to 40 |
| Mobility | Random |
| Population size | 10 to 40 |
| Mutation probability | 0.2 |

The performance of the algorithm has been analyzed in the terms of localization error and network lifetime and number of iterations used to provide the global optimum solution. The location estimation error is defined as a squared error of the estimation and is given by:

$$e_i = \sum_{j=1}^{m'} (d_j - \bar{d}_j)^2, for\ j = 1, 2, .., m', \tag{16}$$

where $\bar{d}_j$ is the estimated distance and $d_j$ is the true distance between unknown node $u_i$ and anchor node $a_j$. The average error is given as:

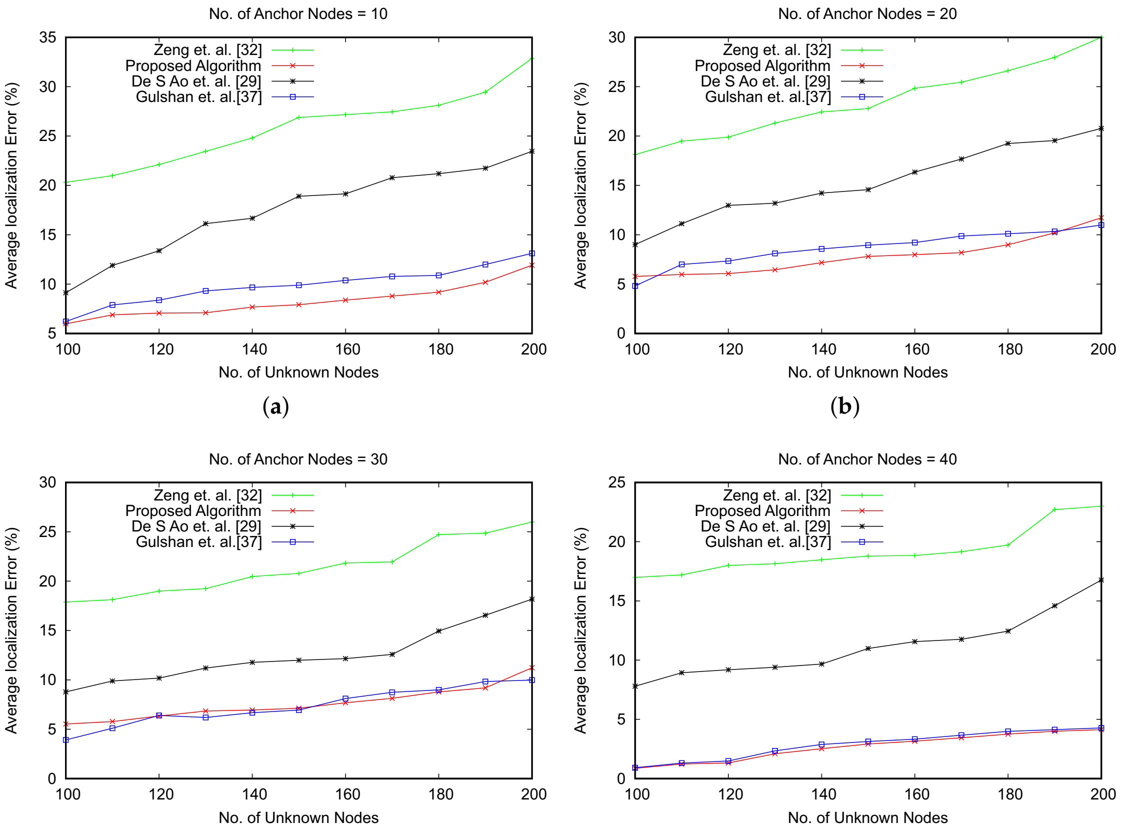
$$e_{avg} = \frac{\sum_{j=1}^{m'} e_i}{m'} \tag{17}$$

The statistical values of localization errors for our different simulation environment are shown in Table 3. Results are shown for the variable number of anchor nodes ranging from 10 to 40, with unknown nodes ranging from 100 to 200. The table also represents the number of anchor nodes used in the VBSO-optimized CDS, by which the given number of unknown nodes has been localized. The result signifies the fact that only 50% to 57% of all the anchor nodes are used. This means that the unused anchor nodes can save their energy for future tasks, which actually elongates the network lifetime.

**Table 3.** Statistical values of experimentation.

| No. of Anchor Nodes | No. of Unknown Nodes | Performance of Proposed Algorithm | | |
|---|---|---|---|---|
| | | Average Localization Error | Optimized CDS Size | Comp. Time (s) |
| 10 | 100 | 0.0291 | 5 | 4.768 |
| 10 | 120 | 0.0338 | 5 | 6.984 |
| 10 | 140 | 0.0379 | 6 | 7.883 |
| 10 | 160 | 0.0534 | 7 | 9.111 |
| 10 | 180 | 0.0549 | 7 | 10.254 |
| 10 | 200 | 0.0588 | 7 | 13.532 |
| 20 | 100 | 0.0277 | 8 | 6.777 |
| 20 | 120 | 0.0279 | 8 | 7.19 |
| 20 | 140 | 0.0388 | 10 | 10.657 |
| 20 | 160 | 0.0397 | 10 | 12.097 |
| 20 | 180 | 0.0444 | 11 | 14.541 |
| 20 | 200 | 0.0487 | 11 | 17.234 |
| 30 | 100 | 0.0276 | 11 | 8.675 |
| 30 | 120 | 0.0295 | 12 | 9.892 |
| 30 | 140 | 0.0387 | 13 | 12.01 |
| 30 | 160 | 0.0455 | 13 | 13.542 |
| 30 | 180 | 0.0487 | 14 | 13.987 |
| 30 | 200 | 0.0499 | 14 | 16.001 |
| 40 | 100 | 0.0243 | 16 | 10.278 |
| 40 | 120 | 0.0377 | 16 | 13.985 |
| 40 | 140 | 0.0487 | 16 | 14.146 |
| 40 | 160 | 0.0489 | 20 | 15.675 |
| 40 | 180 | 0.0502 | 20 | 16.999 |
| 40 | 200 | 0.0548 | 22 | 19.333 |

The results of the average localization error have also been compared with three recent optimization-based approaches, as shown in References [29,32,37]. The comparison results are shown in Figure 10. The results emphasize that our proposed approach reduces the average error for localization of unknown nodes. We have observed that with the increasing number of anchor nodes, the proposed algorithm and the work in Reference [37] have a marginal difference of average localization error. However, the advantage of the proposed algorithm lies with the number of iterations for obtaining the global optimum, as shown in Figure 11. The optimization algorithms consider the number of iterations in searching the global optima as an important performance metric. We have compared the number of iterations executed by varying the number of anchor nodes and unknown nodes. Figure 11 shows the results of comparison, which signify that our proposed algorithm is faster in providing the global optimum value with a lower number of iterations of the optimization process.



(**a**) (**b**)

(**c**) (**d**)

**Figure 10.** Average localization error comparison.

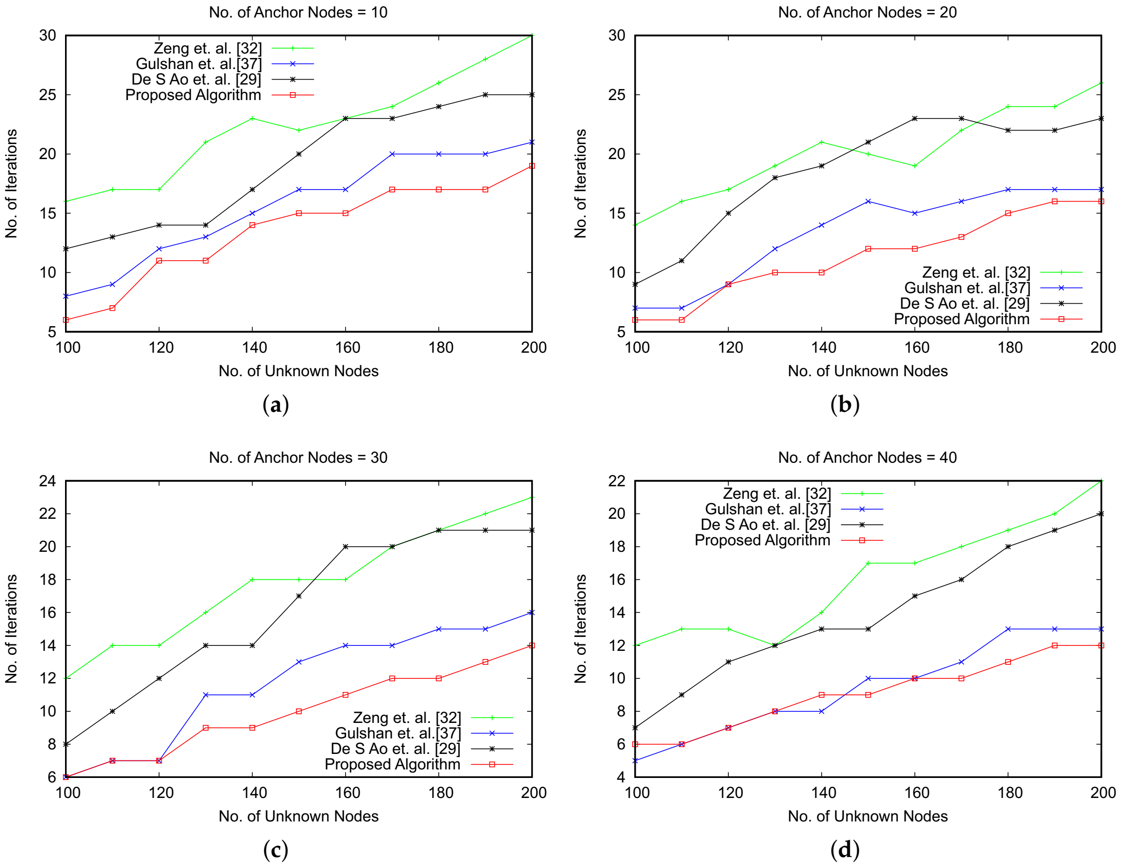
The objective of the proposed algorithm is to elongate the network lifetime. Therefore, we have also evaluated the performance of the proposed algorithm in terms of residual energy. The proposed algorithm uses a linear energy model. The overall residual energy of the proposed optimized CDS depends on the constituents, i.e., the dominator anchor nodes. We have considered a timeline, say *t1* to *t2*, for this energy consumption measurement. Residual energy in time *t* is defined by omitting consumed energy in $\triangle t$ from the initial battery power in $t - \triangle t$. Thus, the energy consumption is determined in $\triangle t$. Therefore, the residual energy equation is given as:

$$\mathcal{E}_{res} = \frac{\sum_{j=1}^{|D|} \mathcal{E}_{init}(t - \triangle t)_j - \mathcal{E}_{consmp}(t - \triangle t)_j}{|D|} \tag{18}$$

$$\mathcal{E}_{res_{percentage}}(\%) = \frac{\mathcal{E}_{res}}{\mathcal{E}_{init}} \times 100 \tag{19}$$

With the minimum usage of message exchange between anchor nodes and unknown nodes and the handing over the optimization and localization tasks on BS, our proposed approach significantly improves the residual energy of the overall network scenario. Figure 12 describes the result for the network lifetime comparison. The data are represented by varying the anchor nodes from 10 to 40. The data are plotted for a number of unknown nodes vs. the percentage of residual energy. In all the circumstances as shown in Figure 12, the percentage of residual energy for our proposed algorithm is higher than the other algorithms in comparison. Moreover, it shows an effective point for our proposed algorithm: With the increasing number of anchor nodes, the energy residual amount also increases.

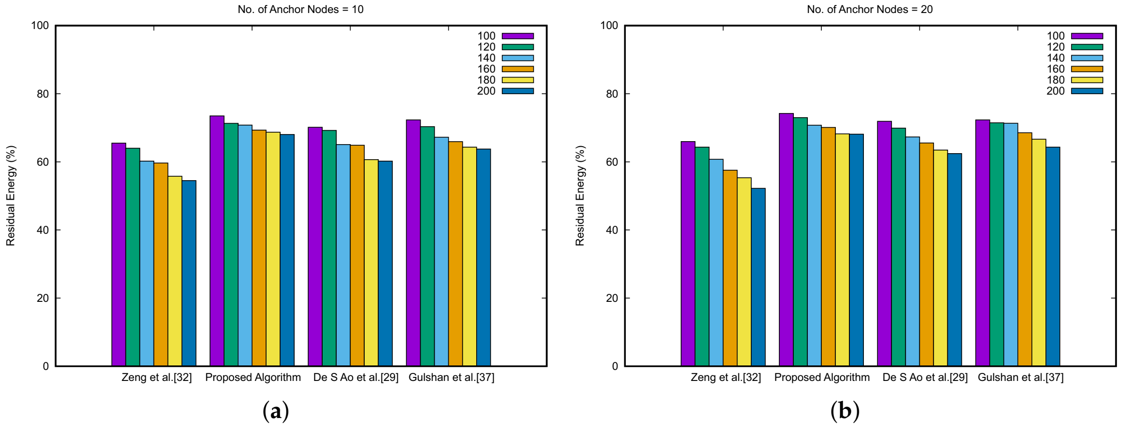
This signifies that the load is balanced and distributed, for which overall residual energy increases. Furthermore, we have analyzed the complexity of algorithms, which are compared in Table 4, where $m$ is the number of anchor nodes used for localization and $n$ is the number of unknown nodes.



(**a**) (**b**)

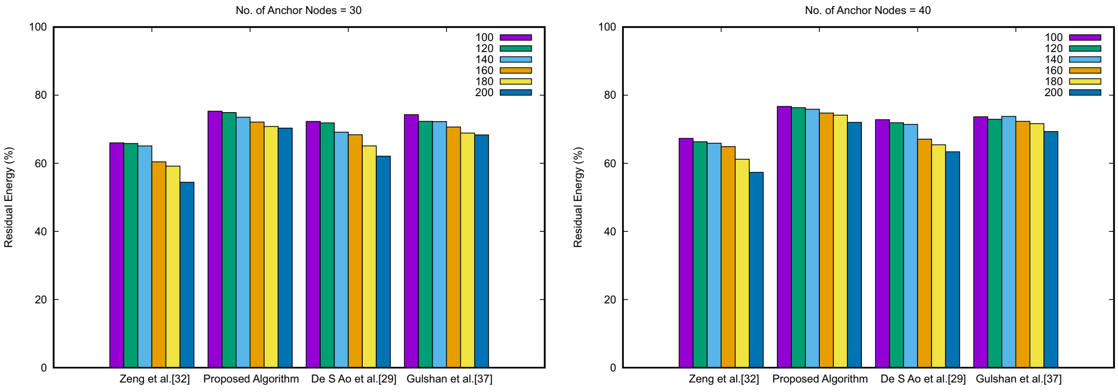
(**c**) (**d**)

**Figure 11.** Comparison of iteration numbers with varying anchor nodes and unknown nodes.



(**a**) (**b**)

**Figure 12.** *Cont.*



(**c**) (**d**)

**Figure 12.** Residual energy comparison.

**Table 4.** Complexity comparison.

| Algorithms | Complexity |
|---|---|
| De S Ao et al. [29] | O($m^2 + n$) |
| Zeng et al. [32] | O($m^n + n$) |
| Gulshan et al. [37] | O($m^n + log_2 n$) |
| Proposed Algorithm | O($nm\ log_n$) |

## 8. Conclusions

In the present study, a novel algorithm is proposed to optimize anchor nodes in CDS. In this case, the algorithm used random mobility for both the anchor as well as unknown nodes and BS was considered to perform the major tasks, such as optimization of CDS, localization with optimized CDS, etc. The vector-based swarm method used for optimization process converges to the global optima rather than the local optima. By keeping anchor nodes and unknown nodes to a minimum working level, traffic overload in the network and message-processing functions are substantially reduced. The localization process is executed with a transformation matrix of nonlinear equations and applying the standard least-square method. The simulated results were compared with recent algorithms. The accuracy and network lifetime are approximately 30% and 67%, respectively, better than the other compared approaches. This algorithm can be used in a dynamic environment, as it converges to the global optimum solution with a lower number of iterations. Additionally, the method is scalable and works with different network sizes, which makes it adaptable to various applications.

**Author Contributions:** G.K. and R.S. conceptualized the idea, proposed the methodology and did the software implementation; M.K.R. supervised the experiments; R.T., T.K., J.L., and J.S.P.S. interpreted the results and performed the validation and formal analysis. All authors have proofread the paper.

**Funding:** This research received no external funding.

**Conflicts of Interest:** All the authors declare no conflict of interest.

## References

1. Kuriakose, J.; Joshi. S.; Raju, R.V.; Kilaru, A. A Review on Localization in Wireless Sensor Networks. In *Advances in Signal Processing and Intelligent Recognition Systems*; Series of Advances in Intelligent Systems and Computing; Springer: Cham, Switzerland, 2014; Volume 264, pp. 599–610.
2. Alrajeh, N.A.; Bashir, M.; Shams, B. Localization techniques in wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2013**. [CrossRef]
3. He, J.S.; Ji, S.; Li, Y.; Pan, Y. *Wireless Ad Hoc and Sensor Networks: Management, Performance, and Applications*; CRC Press: Boca Raton, FL, USA, 2013.

4. Asgarnezhad, R.; Torkestani, J.A. A survey on backbone formation algorithms for Wireless Sensor Networks: (A New Classification). In Proceedings of the Australasian Telecommunication Networks and Applications Conference (ATNAC), Melbourne, Australia, 9–11 November 2011; pp. 1–4.
5. Yetgin, H.; Cheung, K.T.K.; El-Hajjar, M.; Hanzo, L. A Survey of Network Lifetime Maximization Techniques in Wireless Sensor Networks. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 828–854. [CrossRef]
6. Fei, Z.; Li, B.; Yang, S.; Xing, C.; Chen, H.; Hanzo, L. A Survey of Multi-Objective Optimization in Wireless Sensor Networks: Metrics, Algorithms, and Open Problems. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 550–586. [CrossRef]
7. Asorey-Cacheda, R.; Garcia-Sanchez, A.J.; Garcia-Sanchez, F.; Garcia-Haro, J. A survey on non-linear optimization problems in wireless sensor networks. *J. Netw. Comput. Appl.* **2017**, *82*, 1–20. [CrossRef]
8. Curry, R.M.; Smith, J.C. A survey of optimization algorithms for wireless sensor network lifetime maximization. *Comput. Ind. Eng.* **2016**, *101*, 145–166. [CrossRef]
9. Kacimi, R.; Dhaou, R.; Beylot, A.L. Load balancing techniques for lifetime maximizing in wireless sensor networks. *Ad Hoc Netw.* **2013**, *11*, 2172–2186. [CrossRef]
10. Lim, K.S.; Buyamin, S.; Ahmad, A.; Shapiai, M.I.; Naim, F.; Mubin, M.; Kim, D.H. Improving vector evaluated particle swarm optimisation using multiple nondominated leaders. *Sci. World J.* **2014**, *2014*. [CrossRef]
11. Catovic, A.; Sahinoglu, Z. The Cramer-Rao bounds of hybrid TOA/RSS and TDOA/RSS location estimation schemes. *IEEE Commun. Lett.* **2004**, *8*, 626–628. [CrossRef]
12. Tomic, S.; Beko, M.; Dinis, R.; Bernardo, L. On Target Localization Using Combined RSS and AoA Measurements. *Sensors* **2018**, *18*, 1266. [CrossRef] [PubMed]
13. Panichcharoenrat, T.; Lee, W. Two hybrid RSS/TOA localization techniques in cognitive radio system. In Proceedings of the 6th International Conference on Knowledge and Smart Technology (KST), Chonburi, Thailand, 30–31 January 2014; pp. 23–28.
14. Tiwari, S.; Wang, D.; Fattouche, M.; Ghannouchi, F. A Hybrid RSS/TOA Method for 3D Positioning in an Indoor Environment. *ISRN Signal Process.* **2012**, *2012*. [CrossRef]
15. Naseri, H.; Koivunen, V. A Bayesian algorithm for distributed network localization using distance and direction data. *IEEE Trans. Wirel. Commun.* **2017**.
16. Cai, B.; Huang, L.; Xie, M. Bayesian Networks in Fault Diagnosis. *IEEE Trans. Ind. Inform.* **2017**, *13*, 2227–2240. [CrossRef]
17. Cai, B.; Kong, X.; Liu, Y. ; Lin, J.; Yuan, X.; Xu, H.; Ji, R. Application of Bayesian Networks in Reliability Evaluation. *IEEE Trans. Ind. Inform.* **2018**. [CrossRef]
18. Kuila, P.; Jana, P.K. Energy Efficient Load-Balanced Clustering Algorithm for Wireless Sensor Networks. *Procedia Technol.* **2012**, *6*, 771–777. [CrossRef]
19. Liao, Y.; Qi, H.; Li, W. Load-Balanced Clustering Algorithm with Distributed Self- Organization for Wireless Sensor Networks. *IEEE Sens. J.* **2013**, *13*, 1498–1506. [CrossRef]
20. He, J.; Ji, S.; Pan, Y.; Cai, Z. Approximation algorithms for load-balanced virtual backbone construction in wireless sensor networks. *Theor. Comput. Sci.* **2013**, *507*, 2–16. [CrossRef]
21. Raha, A.; Naskar, M.K.; Paul, A. A Genetic Algorithm Inspired Load Balancing Protocol for Congestion Control in Wireless Sensor Networks using Trust Based Routing Framework. *Int. J. Comput. Netw. Inf. Secur.* **2013**, *5*, 9. [CrossRef]
22. Singh, V.K.; Sharma, V. Elitist Genetic Algorithm Based Energy Balanced Routing Strategy to Prolong Lifetime of Wireless Sensor Networks. *Chin. J. Eng.* **2014**, *2014*. [CrossRef]
23. Peng, B.; Li, L. An improved localization algorithm based on genetic algorithm in wireless sensor networks. *Cogn. Neurodyn.* **2015**, *9*, 249–256. [CrossRef]
24. hmed, Y.E.E.; Adjallah, K.H.; Stock, R.; Babikier, S.F. Wireless Sensor Network Lifespan Optimization with Simple, Rotated, Order and Modified Partially Matched Crossover Genetic Algorithms. *IFAC-PapersOnLine* **2016**, *49*, 182–187. [CrossRef]
25. Wang, F.; Wang, C.; Wang, Z.; Zhang, X.Y. A hybrid algorithm of GA + simplex method in the WSN localization. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 731894. [CrossRef]
26. He, J.S.; Ji, S.; Beyah, R.; Xie, Y.; Li, Y. Constructing load-balanced virtual backbones in probabilistic wireless sensor networks via multi-objective genetic algorithm. *Trans. Emerg. Telecommun. Technol.* **2015**, *26*, 147–163. [CrossRef]

27. Wani, S.M.; Nalbalwar, S.L. Identification of Balanced Node for Data Aggregation in Wireless Sensor Network. In Proceedings of the International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 3–5 March 2016.
28. Larios, D.F.; Barbancho, J.; Molina, F.J.; León, C. LIS: Localization based on an intelligent distributed fuzzy system applied to a WSN. *Ad Hoc Netw.* **2012**, *10*, 604–622. [CrossRef]
29. De Sá, A.O.; Nedjah, N.; Mourelle, L.D.M. Distributed efficient localization in swarm robotics using Min-Max and Particle Swarm Optimization. *Expert Syst. Appl.* **2016**, *50*, 55–65. [CrossRef]
30. Sharma, V.; Grover, A. A modified ant colony optimization algorithm (mACO) for energy efficient wireless sensor networks. *Optik Int. J. Light Electron Opt.* **2016**, *127*, 2169–2172. [CrossRef]
31. Shakibian, H.; Moghadam Charkari, N. In-cluster vector evaluated particle swarm optimization for distributed regression in WSNs. *J. Netw. Comput. Appl.* **2014**, *42*, 80–91. [CrossRef]
32. Zeng, T.; Hua, Y.; Zhao X.; Liu, T. Research on Glowworm swarm optimization localization algorithm based on wireless sensor network. In Proceedings of the IEEE International Frequency Control Symposium (IFCS), New Orleans, LA, USA, 9–12 May 2016.
33. Li, S.; Qin, F. A dynamic neural network approach for solving nonlinear inequalities defined on a graph and its application to distributed, routing-free, range-free localization of WSNs. *Neurocomputing* **2013**, *117*, 72–80. [CrossRef]
34. Cottone, P.; Gaglio, S.; Lo Re, G.; Ortolani, M. A machine learning approach for user localization exploiting connectivity data. *Eng. Appl. Artif. Intell.* **2016**, *50*, 125–134. [CrossRef]
35. Cespedes-mota, A.; Castanfnon, G.; Martinez-herrera, A.F.; Cardenas-barron, L.E. Optimization of the Distribution and Localization of Wireless Sensor Networks Based on Differential Evolution Approach. *Math. Probl. Eng.* **2016**, 7918581. [CrossRef]
36. Free space model. Available online: https://www.isi.edu/nsnam/ns/doc/node217.html (1 October 2018).
37. Kumar, G.; Rai, M.K. An energy efficient and optimized load balanced localization method using CDS with one-hop neighbourhood and genetic algorithm in WSNs. *J. Netw. Comput. Appl.* **2017**, *78*, 73–82. [CrossRef]
38. Zhou, Y.; Li, X.; Gao, L. A differential evolution algorithm with intersect mutation operator. *Appl. Soft Comput. J.* **2013**, *13*, 390–401. [CrossRef]
39. Tang, P.H.; Tseng, M.H. Adaptive directed mutation for real-coded genetic algorithms. *Appl. Soft Comput. J.* **2013**, *13*, 600–614. [CrossRef]
40. Gao, H.; Xu, W. Particle swarm algorithm with hybrid mutation strategy. *Appl. Soft Comput. J.* **2011**, *11*, 5129–5142. [CrossRef]
41. Haupt, R.L.; Haupt, S.E. *Practical Genetic Algorithms*, 2nd ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2004; ISBN 0471455652.
42. Price, K.V. Differential Evolution. *Intell. Syst. Ref. Libr.* **2013**. [CrossRef]
43. Karaboga, D.; Akay, B. A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems. *Appl. Soft Comput. J.* **2011**, *11*, 3021–3031. [CrossRef]
44. Mallipeddi, R.; Suganthan, P.N.; Pan, Q.K.; Tasgetiren, M.F. Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl. Soft Comput. J.* **2011**, *11*, 1679–1696. [CrossRef]
45. Madsen, K.; Nielsen, H.B.; Tingleff, O. *Methods For Non-Linear Least Squares Problems*; Informatics and Mathematical Modelling Technical University of Denmark: Lyngby, Denmark, 2004.