

## Research Article

# Optimization of the Distribution and Localization of Wireless Sensor Networks Based on Differential Evolution Approach

**Armando Céspedes-Mota,<sup>1</sup> Gerardo Castañón,<sup>1</sup>  
Alberto F. Martínez-Herrera,<sup>1</sup> and Leopoldo Eduardo Cárdenas-Barrón<sup>2</sup>**

<sup>1</sup>*Department of Electrical and Computer Engineering, Tecnológico de Monterrey, Eugenio Garza Sada No. 2501 Sur, Monterrey, NL, Mexico*

<sup>2</sup>*School of Engineering and Sciences, Tecnológico de Monterrey, Eugenio Garza Sada No. 2501 Sur, Monterrey, NL, Mexico*

Correspondence should be addressed to Leopoldo Eduardo Cárdenas-Barrón; [lecarden@itesm.mx](mailto:lecarden@itesm.mx)

Received 9 November 2015; Accepted 7 February 2016

Academic Editor: Hou-Sheng Su

Copyright © 2016 Armando Céspedes-Mota et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Location information for wireless sensor nodes is needed in most of the routing protocols for distributed sensor networks to determine the distance between two particular nodes in order to estimate the energy consumption. Differential evolution obtains a suboptimal solution based on three features included in the objective function: area, energy, and redundancy. The use of obstacles is considered to check how these barriers affect the behavior of the whole solution. The obstacles are considered like new restrictions aside of the typical restrictions of area boundaries and the overlap minimization. At each generation, the best element is tested to check whether the node distribution is able to create a minimum spanning tree and then to arrange the nodes using the smallest distance from the initial position to the suboptimal end position based on the Hungarian algorithm. This work presents results for different scenarios delimited by walls and testing whether it is possible to obtain a suboptimal solution with inner obstacles. Also, a case with an area delimited by a star shape is presented showing that the algorithm is able to fill the whole area, even if such area is delimited for the peaks of the star.

## 1. Introduction

It is well-known that location is a fundamental problem in wireless sensor networks (WSN) and it is essential based on either the distance or the connectivity among the nodes in the wireless sensor network. It is necessary to truthfully localize the sensors to determine the value of different parameters such as temperature and geographic coordinates of a given location, detect the occurrence of events, classify a detected object, and track an object. Sensors usually have low energy consumption, network self-organization, collaborative signal processing, and querying abilities. In consequence, it is crucial that the required data be disseminated and localized to the proper end users [1, 2].

A WSN is comprised of a large number of sensor nodes, which are densely positioned into a given area. Basically, they are a set of almost-static nodes and the nodes are not connected to each other. Their distribution inside a given

area with minimum energy connectivity is an NP-complete problem [3–6].

For applications like automatic location, tracking, and guidance, the well-known global positioning system (GPS) is a space-based navigation system that provides location any time and is widely used. One way to acquire location information of sensor nodes is to equip each sensor with a GPS, but, in practice, in sensor networks with a large number of sensor nodes, only a limited portion of sensors are equipped with a GPS called *anchors* in the literature [7, 8]. An example of location-based protocol is presented by [9], it maintains a minimum energy communication network for wireless networks using low power GPS, and also it helps to calculate a minimum-power topology for mobile and stationary nodes by the use of a sink or concentrator node. Moreover, topology optimization finds the best network topology under certain constraints assuming that nodes must maintain communication to at least one neighbor [10].

Localization algorithms usually assume that sensors are distributed in regular areas without cavities or obstacles, but outdoor deployment of wireless sensor networks is very different. To solve this problem a reliable anchor-based localization technique is proposed by [7] in which the localization error due to the irregular deployment areas is reduced. Locating sensors in irregular areas is also analyzed by [8]; the problem is formulated as a constrained least-penalty problem. The effects of *anchors* density, range error, and communication range on localization performances are analyzed. Nonconvex deployment areas such as C-shaped and S-shaped topologies are used. Moreover, in [11] the authors focused on the area coverage problem in a nonconvex region, as the region of interest limited by the existence of obstacles, administrative boundaries, or geographical conditions. The authors apply the discrete particle swarm optimization (PSO) algorithm with the use of a grid system to discretize the region of interest.

WSN deployment consists in determining the positions for sensor nodes in order to obtain the desired coverage, localization, connectivity, and energy efficiency. Available PSO solutions for the deployment problem are analyzed centrally on a base station to determine positions of the sensor nodes, the mobile nodes, or base stations as described in [12–14]. A node localization approach using PSO is described in [15–17].

On the other hand, flocking is the event or singularity in which a large number of agents (sensors) with limited environmental information and simple rules are organized into a coordinated motion and are widely used in many control areas including mobile sensor networks. Flocking of multiagents with virtual leader is presented in [18], in which the sensors in a network can be moved by using the improved algorithm. However, coordinated control of multiagent systems with leaders points to making all agents reach a coordinated motion with leaders. The case with multiple leaders is named containment control, in which every control input of agents is subject to saturation owing to its upper and lower limits in multiagent systems. This case is cleverly described in [19], focused on switching networks. The containment control with periodically intermittent communication and input saturation is deeply analyzed by [20] in which the authors design both state feedback and output feedback containment control protocols with intermittent input saturation via low gain feedback. This technique achieves a practical semiglobal containment of such agents on fixed networks.

There are some existing results in the literature that analyze the wireless sensors distribution in the three-dimensional space such as in [21–23]. These scenarios analyze the distribution of the sensor networks under certain conditions that are not considered in this research work. The three-dimensional sensor distribution is out of the scope of this research paper but is considered as a future work. For instance, to extend this research paper to the three-dimensional case, it is necessary to consider new objective functions, new scenarios with regular and irregular areas, and new upper and lower bounds; also, instead of computing the coverage area, it will be necessary to compute the involved volume and this requires more computational time to obtain results with low granularity.

In this research work, to optimize the distribution and location of sensor nodes, the Differential Evolution Algorithm (DEA) is applied. This technique has been successfully applied on different problems due to its simplicity of use [24–26]. Its taxonomy is in the category of bioinspired techniques. For each cycle denoted as generation, DEA uses a set of potential random solutions in a given search space, called individuals. In this research paper, an individual contains the information of the coordinates of the sensor nodes ( $x$ ,  $y$ ) and their communication radii values. For each individual in the population, a descendant is generated from three parents. One parent, called the main parent, is disturbed by the vector of differences of the other two parents. In the case of the descendant having a better performance according to the objective function, it substitutes the individual, else the original individual is retained and is passed on to the next generation of the algorithm and the descendant is eliminated. This process is repeated till it attains the stop condition. The readers are referred to [24] for a complete theoretical analysis of the DE algorithm.

Early work to solve the node distribution optimizing area and energy based on Differential Evolution Algorithm is presented in authors of [27]. But they only consider a typical square area without obstacles and the work lacks information about the energy consumption and the mobility of the sensor nodes; also, the localization of sensors is not analyzed. Extended cases for irregular areas are shown by [28], in which the interference rate is included and how much nodes are turned on at same time. In [29], the problem is divided in uniformity (node spreading) and connectivity (node clustering) regarding the area; the energy is analyzed by dividing the problem in operational consumed energy and data-exchange consumed energy.

Differential evolutions with heuristic algorithms for nonconvex optimization on sensor network localization are analyzed in [30]; here, the connections and disconnections constraints are considered by the authors. Also, the connectivity-based localization problem is analyzed as an optimization problem with both convex and nonconvex constraints. In [31], the minimization equation of WSN localization error problem and an approach of differential evolution for minimization of localization error in WSN are studied considering that sensor node localization is the ability of an individual node to determine the location information. There are also surveys related to localization in wireless sensor network described in [32, 33].

Compared to the previous listed works, here, the contributions are focused on building and establishing communication links even in the presence of obstacles or very restricted boundaries. DEA is used to optimize the distribution and as a consequence the location of the sensor nodes to minimize the overlap by using an extra function as shown by [28]. The main difference with [28] is that the overlap considered as the interference is also managed in terms of restrictions to minimize the overlap very quickly. It also applies for [29]. Besides, an extra parameter called random- $M$  is used in the mutation operator to avoid the stagnation of DEA as suggested in [25].

In this research work, the results are presented for two area boundaries: inner star and a typical square room. But,

instead of having an empty square room, the room includes subdivisions called walls, which will act as obstacles regarding the signal propagation, showing the optimization of the fitness function formed by the area coverage rate, the energy consumption, and an additional parameter that is related to the overlap among the nodes: the redundancy. For the used configurations, the initial node positions are randomly assigned. For the inner star, a minimum spanning tree (MST) is used at the end of each generation to check the connectivity of the nodes. For the scenarios with obstacles, such verification is performed for each individual of the population. At the end of each generation, the Hungarian algorithm [34] is used to find the shortest distance from their initial to the final location. In all scenarios here presented, the algorithm determines the initial and the final location of the sensor nodes. Each sensor node is located with respect to any of each other following the branches of the minimum spanning tree. The distance between nodes is also calculated by adding the respective distances of the branches among them.

The rest of this paper is organized as follows. Section 2 describes the methodology. Section 3 presents the DEA and minimum spanning tree algorithms for wireless sensor networks distribution and location. Section 4 summarizes the pseudocode applied on DEA. Section 5 presents the numerical results and a discussion. Finally, Section 6 presents some conclusions.

## 2. Description of the Method

**2.1. Sensor Coverage Model.** There is a particular interest in the sensor distribution and location where communication with neighboring nodes is more energy efficient. Also, a particular interest in the sensor distribution and location on irregular areas is considered. The main purpose of the sensor coverage model is to obtain a balance among the maximum effective coverage area, the minimum communication sensor energy, and reducing overlap.

The node set on the target area  $A_t$  is given by

$$\mathcal{N} = \{N_1, N_2, \dots, N_z\}, \quad (1)$$

and here  $z$  is the cardinality of  $\mathcal{N}$ .

The coverage range of a node  $N_i$  is defined as a circle centered at its coordinates  $(x_i, y_i)$  with sensing radius  $r_i$ .  $(x_i, y_i)$  coordinates define the position of sensor nodes and the subindex  $i$  is the sensor node index. A grid point  $(x, y)$  is covered by a sensor node  $i$  if and only if its distance to the center  $(x_i, y_i)$  of the circle is not larger than the sensing radius  $r_i$ .

A random variable  $I_i$  is defined in order to describe the event that the sensor node  $N$  covers a given point  $(x, y)$ . Then, the probability of event  $I_i$ , given by  $P\{I_i\}$ , is equal to the coverage probability  $P_{\text{cov}}(x, y, N_i)$ . This probability is a binary valued function which is expressed as

$$P_{\text{cov}}(x, y, N_i) = \begin{cases} 1 & \text{if } (x - x_i)^2 + (y - y_i)^2 \leq r_i^2 \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where a point  $(x, y)$  is covered by a sensor node  $N_i$  if and only if its distance to the center  $(x_i, y_i)$  is less than or equal

to the sensing radius  $r_i$ . It is obvious that a point must be evaluated against all the nodes in the network; thus

$$S(x, y) = \sum_{i=1}^z P_{\text{cov}}(x, y, N_i), \quad (3)$$

and here  $S(x, y)$  represents the number of times that the coordinate  $(x, y)$  is covered by the node set.

However, to find the proper covered area by the set of nodes, the function  $R(x, y)$  is calculated to check if the coordinate is covered at least once or if it is not covered at all  $N_i$ ;

$$R(x, y) = \begin{cases} 1 & S(x, y) \geq 1 \\ 0 & S(x, y) = 0. \end{cases} \quad (4)$$

The covered area is calculated as

$$A_{\text{cov}}(N) = \frac{\sum_{y=1}^n \sum_{x=1}^m R(x, y)}{A_t}. \quad (5)$$

Equation (5) is applied in our simulation by sweeping all  $(x, y)$  points of the target area  $m \times n$ . For different area configurations, this area is replaced by the bounded area of interest. Now,  $f_1(x)$  is expressed as

$$f_1(x) = 1 - A_{\text{cov}}(N). \quad (6)$$

**2.2. Redundant Area.** The uncovered nonnormalized area is computed as

$$A_{\text{zeros}} = A_t (1 - A_{\text{cov}}(N)). \quad (7)$$

The redundant area is computed as

$$\text{Red}(x, y) = \begin{cases} 1 & S(x, y) > 1 \\ 0 & S(x, y) = \{0, 1\}, \end{cases} \quad (8)$$

which verify if a coordinate is covered by more than one node. As a result,

$$A_{\text{red}}(N) = \frac{\sum_{y=1}^n \sum_{x=1}^m \text{Red}(x, y)}{A_t - A_{\text{zeros}}}, \quad (9)$$

where  $A_{\text{cov}}(N)$  is the area covered by the nodes,  $A_{\text{red}}(N)$  is the redundant area, and  $A_{\text{zeros}}(N)$  is the area not covered by the nodes. Now,  $f_2(x)$  is defined as

$$f_2(x) = A_{\text{red}}(N). \quad (10)$$

The function  $f_2(x)$  is equivalent to the one given by [28].

**2.3. Energy Consumption.** There exist several models for wireless networks efficient energy consumption; for example, see [6, 35, 36], among others. In this paper, the total energy consumption in maintaining the network connectivity is given by

$$E_{\text{cst}} = \mu \sum_{i=1}^z r_i^\alpha, \quad (11)$$

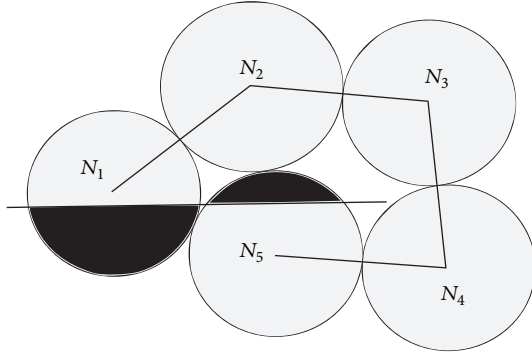


FIGURE 1: Wall analysis to compute area coverage and to form a MST.

and here  $\mu$  is the node sensing parameter in milliwatts per square meter ( $\text{mW}/\text{m}^2$ ) and  $r_i$  is the sensor radius of  $N_i$  in meters (m), and  $\alpha = 2$  since our communication medium is the air [37]. In this research work  $\mu = 0.005$  ( $\text{mW}/\text{m}^2$ ) is used. Now,  $f_3(x)$  is expressed as

$$f_3(x) = E_{\text{cst}}(N). \quad (12)$$

Nevertheless, to normalize  $f_3(x)$ , it is needed to use a constant which considers the maximum value that  $f_3(x)$  may obtain. Then, the possible maximum value is used in its inverse form as

$$n_3 = \frac{1}{\mu z r_{\text{max}}^2}, \quad (13)$$

and here  $r_{\text{max}}$  is the maximum sensor radius in meters (m).

**2.4. Wall Behavior Analysis.** To simplify the analysis of the obstacles, it has adopted on-off criterion to compute the area coverage in the same way that a channel is modelled with the well-known binary symmetric channel model [38], by omitting existing effects such as signal propagation through barriers or obstacles. A similar approach is used in [28]. Figure 1 shows how the MST should be formed. Additionally, it also shows how the area can be treated and computed if a wall crosses the area coverage of a given node. It can see that  $N_4$  is used to connect  $N_3$  and  $N_5$  to prevent the wall. That means a link should not cross a wall. About the computed area coverage, dark areas in  $N_1$  and  $N_5$  are not computed because they are not located aside from the center of the area coverage; that is, they are located at the other side of the wall. Such restriction is considered to compute the area coverage in order to give feedback on this research work.

### 3. The Differential Evolution Algorithm

The DEA generates a set of  $N_p$  individuals, each one of dimension  $D$ . Particular for this problem, the dimension  $D$  includes the coordinates  $(x_i, y_i)$  and the radii  $(r_i)$  of the sensor nodes in the same vector  $(x_i, y_i, r_i)$ . Each individual is a potential solution of the problem, generated randomly but bounded by the constraints of the problem to solve. To perform its duties, the algorithm tunes the solutions

with the following constants: the mutation constant ( $M$ ), the recombination constant ( $C_r$ ), and the population size ( $N_p$ ) which are defined by the user, where  $0 \leq M \leq 1$  and  $0 \leq C_r \leq 1$ . Given  $G$  generations (cycles), operations like mutation, recombination, and selection are applied to each individual to obtain new and improved individuals that should be able to substitute the old ones. If it is successful, each time a generation is executed, a better suboptimal solution is obtained, which is near to the global optimal solution. The following is a description of the operators used in the DE algorithm that finds the most promising region in the search space.

**3.1. Objective Functions.** In this paper, the DE algorithm seeks to obtain a balance among the maximal effective coverage rate, the minimum sensor power communication, and the overlap. To convert the network coverage rate into minimal functions the opposite effect of the coverage rate is applied. Therefore, the network effective coverage rate is expressed by

$$\text{Min } f_1(x) = 1 - A_{\text{cov}}(N), \quad (14)$$

the redundant area is determined by

$$\text{Min } f_2(x) = A_{\text{red}}(N), \quad (15)$$

and the minimum network energy communication is determined by

$$\text{Min } f_3(x) = E_{\text{cst}}(N). \quad (16)$$

The values of  $\text{Min } f_1(x)$ ,  $\text{Min } f_2(x)$ , and  $\text{Min } f_3(x)$  are part of the fitness function for measuring the result. A linear combination of the objective functions converts the original multiobjective function into a single-objective function as follows:

$$\text{Min } \sum_{k=1}^K w_k f_k(x), \quad \text{where } w_k \geq 0; \sum_{k=1}^K w_k = 1, \quad (17)$$

and here  $w_k$  are weight coefficients expressing the relative importance of each objective function and  $k$  is the number of objective functions that the total objective function  $f(x)$  has.

**3.2. The Fitness Function.** The scenario where DEA is to be applied is limited by the area boundaries and the small amount of energy for each node. Additionally, it is important, aside from the optimization of the area coverage and the energy, to improve the separation among the nodes by adding an extra function which helps to prevent the overlap. Then, the fitness function is defined by

$$f(x) = w_1 f_1(x) + w_2 f_2(x) + w_3 n_3 f_3(x), \quad (18)$$

where  $w_1$ ,  $w_2$ , and  $w_3$  are weight coefficients and  $n_3$  is the normalization coefficient of  $f_3(x)$ . These values are chosen in such a way that the whole objective function is in the interval  $[0, 1]$  and  $w_1 + w_2 + w_3 = 1$ . If the value is minimal, it means that a good combination of maximum area, minimum energy, and minimum overlap is achieved.

**3.3. The Mutation Operator.** Let  $x^{r1}, x^{r2}, x^{r3} \in \{1, \dots, N_p\}$ ;  $x^{r1} \neq x^{r2} \neq x^{r3} \neq x^i$ .  $x^{r1}, x^{r2}$ , and  $x^{r3}$ , different from the current target vector  $x^i$ , and  $M$  is a scaling factor named as the mutation constant, which must satisfy  $M > 0$ . For each target vector  $x^i$ ,  $i = 1, \dots, N_p$ , the mutant individual  $m^i$  is created using (19) [25]:

$$m^i = x^{r1} + M(x^{r2} - x^{r3}), \quad (19)$$

where the mutation operator is used to control the magnitude of the difference between two individuals, and it has a strong influence into the convergence of DEA.

**3.4. The Recombination Operator.** Let  $t^i$  be the trial vector, rand a uniform random number in the interval  $[0, 1]$ , and  $C_r$  the recombination constant. To obtain the trial vector from the mutation vector and the target vector, (20) holds. Consider

$$t^{i,j} = \begin{cases} m^{i,j} & \text{if rand} < C_r \quad \forall j \\ x^{i,j} & \text{otherwise.} \end{cases} \quad (20)$$

As it is known, each individual is formed for  $D$  components in the way that  $j = \{1, 2, \dots, D\}$ . If  $\text{rand} \leq C_r$ , then the  $j$ th element of the mutation vector is chosen to be the  $j$ th component of the trial vector; otherwise, the  $j$ th of the trial vector is selected instead. The recombination operator helps to increase the diversity in the mutation process, so an adequate assignment will lead to obtaining accurate solutions [25].

**3.5. The Selection Operator.** Let  $\text{Pop}_i$  be the vector population of the next generation. To obtain it, the selection operation checks that

$$\text{Pop}_i = \begin{cases} t^i & \text{if } f(t^i) < f(x^i) \\ x^i & \text{otherwise.} \end{cases} \quad (21)$$

If  $f(t^i) < f(x^i)$ , it means that the trial vector is a better individual than the target vector, thus being accepted as the new individual [25]. From these improved individuals either  $t^i$  or  $x^i$ , the best solution is registered until DEA finishes its duties.

**3.6. The Random- $M$  Mutation Parameter.** Let rand be a random number in the range of  $(0 < \text{rand} < 1)$  that changes during the evolution process. Then, (22) is a modified version of (19) and shows how the main operator changes with this addition:

$$m^i = x^{r1} + \text{rand} * M(x^{r2} - x^{r3}). \quad (22)$$

By multiplying  $M$  by rand, randomly increment or decrement its value. This permits the algorithm to get benefits from both a high convergence speed and a strong reduction of the probability of stagnation. After introducing rand, a significant improvement was perceived in the convergence of the algorithm. About other successful results, readers can read [25, 39].

## 4. Pseudocode for DEA in a WSN

As explained in Section 3 DEA is tuned with  $M$ ,  $C_r$ , and  $N_p$ , and the number of generations is given by  $G$ . The complete population set is stored at Pop. Each individual is formed by  $\text{Pop}_i = x_{i,1}, \dots, x_{i,z}; y_{i,1}, \dots, y_{i,z}; r_{i,1}, \dots, r_{i,z}$ , where each member belongs to each  $N_i$  formed by a triad  $(x_i, y_i, r_i)$ ,  $(x_i, y_i)$  being random coordinates. To start with DEA, it is necessary to consider an initial population  $P_{\text{initial}}$ , where  $P_{\text{initial}} = \text{Pop}_1$  for the sake of simplicity. Remembering the steps followed by DEA, three different individuals  $x^{r1}, x^{r2}, x^{r3}$  are obtained. Here, these individuals are obtained from Pop, where  $x^{r1}, x^{r2}, x^{r3}$  are equivalent to having  $x_{r1,j}, y_{r2,j}, r_{r1,j}$ .

The validation process of  $t^i$  is based on the triad  $(x_i, y_i, r_i)$  regarding the boundaries,  $r_{\text{max}}$  and  $r_{\text{min}}$  (or the smallest radio available in the node set  $r_{\text{min}}$ ). If  $r > r_{\text{max}}$ , (23) is applied to regenerate  $r$  as

$$r_{\text{adj}} = r_{\text{min}} + w_1 * \left( \frac{\text{rand}()}{2} + \frac{1}{2} \right) * (r_{\text{max}} - r_{\text{min}}). \quad (23)$$

On the other hand, if  $(x, y)$  are outside the boundaries,  $(x, y)$  coordinates are regenerated. One goal is to reduce the overlap of these nodes on each population member, and then in this way better energy and area combinations will be obtained. A distance among the nodes should be maintained to prevent excessive separations and this condition is controlled by the largest radius of the involved nodes multiplied by a parameter *scale*, typically  $r_{\text{max}}$ . The easiest way is to check the triad  $(x_i, y_i, r_i)$  of  $N_i$  regarding to its neighbors. A strategy to prevent wasting area is to move  $(x_i, y_i)$  away from the boundaries. This is also controlled by *scale* that multiplies  $r_i$ . In this work, wall scenarios are included, and it is important to note that it is possible that an area coverage of  $N_i$  can be cut out when it is crossing a wall. Then, the effective area coverage will be identified in the side where  $(x_i, y_i)$  is located. Additionally, to check if the obtained area covers enough surface,  $A_{\text{th}}$  is used as the minimum percentage acceptable area. Here, the parameter *scale* can be treated as fixed *scale* or variable *scale* = *scale* + *delta*.

The Prim algorithm [40] obtains the minimum spanning tree (MST) while the Dijkstra algorithm [41] gives a star topology. Prim-Dijkstra [42] uses the location of the sensor nodes and determines a topology that is a mixture of both MST and a star topology. In this research work, the MST is formed by using arbitrarily a central node ( $\text{cnode} = 5$ ) and the maximum distance to generate a link ( $r_i + r_j, i \neq j$ ). If  $d(N_i, N_j) > r_i + r_j$ , the link does not exist. To satisfy this constraint the Prim-Dijkstra algorithm [42] is used. Prim-Dijkstra will perform its duties when each link of the MST does not cross forbidden zones, boundaries, or walls; otherwise the nodes that do not fulfill these restrictions will not be neighbors of  $N_i$ .

The objective function launches the area, the energy, and the redundancy aside from the fitness. As it has been described through this work, the minimum fitness indicates that efficient combination of area usage, energy, and redundancy has been obtained.

Finally, the initial and final location of sensor nodes are given by  $\text{Pop}_{\text{initial}}$  and  $\text{Pop}_{\text{best}}$ , respectively. The Hungarian

```

procedure DEA
  Set the control parameters
  Create Pop
  Evaluate fitness, area, energy, redundancy
  Start generations counter  $s = 0$ 
  Repeat until stop conditions are completed:
  (tree is built and  $s > G$ ) or
  (tree is built and area coverage  $\geq A_{th}$ )
  for  $i = 1$  to  $N_p$  do
    Select from Pop
    Mutation  $m^i$ , equation (19);
    Recombination  $t^i$ , equation (20);
    Validate  $t^i$ .
    Update  $(x, y, r)$  if needed;
    De-overlap  $t^i$ , update  $(x, y)$  if needed;
    Selection, equation (21)
  end for
  Popbest fitness, area, energy and redundancy
  Apply Hungarian algorithm to Popbest versus Popinitial.
  Apply Prim-Dijkstra(cnode,  $\alpha$ ) to array of links;
  if Some nodes are not connected to the tree then
    Continue;
  else nodes are connected
    Tree has been build.
  end if
   $s = s + 1$ 
   $scale = scale + \delta$  (Variable option until  $G$ )
  End repeat
  return Popbest, Fitness, Area, Energy, redundancy
end procedure

```

ALGORITHM 1: DEA applied to WSN.

Algorithm [34] helps to find the minimal distance that a node will move away to the ending location from its original location. It receives Pop<sub>best</sub> and Pop<sub>initial</sub> to determine the final location of the sensors with respect to the initial location. By using  $(x_i, y_i)$ , the Hungarian Algorithm uses a matrix of distances, where each distance is calculated by using the initial coordinates with respect to the final coordinates. The initial coordinate of a node is evaluated against all the final coordinates, to obtain the desired minimum distances. Then, the solution with the minimum distance in which each node goes from the initial to the final position is chosen. In this work, in scenarios with walls, the distance from one node to any other one is through the branches of the MST preventing crossing any wall or obstacle. Algorithm 1 summarizes the behavior of DEA.

## 5. Numerical Results and Discussion

The algorithm is implemented in machines with the following characteristics:

MATLAB R2013a.  
 Windows 7 Enterprise 2009, service pack 1.  
 Processor: Intel (R) Xeon (R) CPU.  
 W3690 @ 3.47 GHz 3.46 GHz.

Installed memory (RAM): 8.00 GB.

System type: 64-bit operating system.

For the inner star scenario, the effective sensor radius is between 3.5 and 8 meters, and, the weights are  $w_1 = 3 * 0.90/4$ ,  $w_2 = 0.90/4$ , and  $w_3 = 0.10$ . Here, two different strategies were tested. In the first one, the value for *scale* is fixed to 0.7000, while in the second one *scale* is increased in steps of 0.0070.

For the wall scenarios, the effective sensor radius of the sensor nodes is between 6 and 8 meters and the target areas are 80 (m)  $\times$  80 (m). DEA was tested based on the increment of *scale* in steps of 0.008 units when each generation is executed, thus controlling the separation among nodes instead of spreading them immediately and also helping with the speed of the algorithm. Here,  $w_1 = 3 * 0.8/4$ ,  $w_2 = 0.8/4$ , and  $w_3 = 0.2$ . If the node coverage area is crossed by a wall, the region of the coverage area that does not contain the center of the node should not be considered in the computation of the area because the wall reduces the signal propagation. For simplification purposes in the calculation of the area coverage on each node, a criteria on-off is adopted. If the center is behind a wall, the area crossed by a wall is not considered.

As it was explained in the previous section, the algorithm stops when the number of established generations is reached and the tree is formed, or a threshold area is reached and the tree is formed. The sensor nodes are identified as  $N_1, N_2, \dots, N_j, \dots, N_z$ , and the circles show their sensor range. The values of  $M = 0.8$ ,  $N_p = 35$ , and  $C_r = 0.2$  are used.

For all, the inner star and wall scenarios, 40 tests were computed using both variable and fixed criteria. The percentage of covered area, energy consumption in maintaining the network connectivity, redundant area, and radii values are determined and analyzed. Also, the execution time for each test is determined. Increasing the number of sensor nodes, the noncovered areas of all scenarios can be solved. The initial and final location of sensor nodes can be obtained.

**5.1. Inner Bounded Star.** The set of Figure 2 show the final distribution obtained with fixed and variable criteria, when DEA is applied. The initial positions of the nodes are randomly set, without forming a tree. The star is delimited by the following parameters:  $s = 12$ ,  $p = (0 : s)$ ,  $r_1 = 40$ ,  $r_2 = 20$ ,  $t = (-1/4 : 1/s : 3/4) * 2 * \pi$ ,  $r = (r_1 + r_2)/2 + (r_1 - r_2)/2 * (-1)^p$ , and

$$\begin{aligned}
 xv &= r * \cos(t) + 40, \\
 yv &= r * \sin(t) + 40,
 \end{aligned} \tag{24}$$

where  $xv$  and  $yv$  are used in the Matlab command `inpolygon` (`inpolygon(x, y, xv, yv)`) and  $(x, y)$  are the coordinates of the point to be tested whether it is inside the figure or not.

Testing the fixed criterion, the final distribution fits the inner star as it can be seen in Figure 2(a), the smallest nodes appear as connector between the nodes with larger area, and sometimes they appear at peaks of the star. Additionally, the nodes are distributed regarding the overlaps, showing that the algorithm tries to arrange the smallest nodes into the peaks or at least to arrange these nodes with an average area coverage. It is noted that the tree is formed successfully. Also, the results

TABLE 1: Results for inner star. Tests # 24, 30 nodes.

Criteria	Area%	Energy (mW)	Radii (m)	Redundancy%	Time (s)	Gen.
Variable	91.83	6.1192	6.2856	48.37	546.9349	100
Fixed	93.33	5.3197	5.8639	37.19	889.1790	100

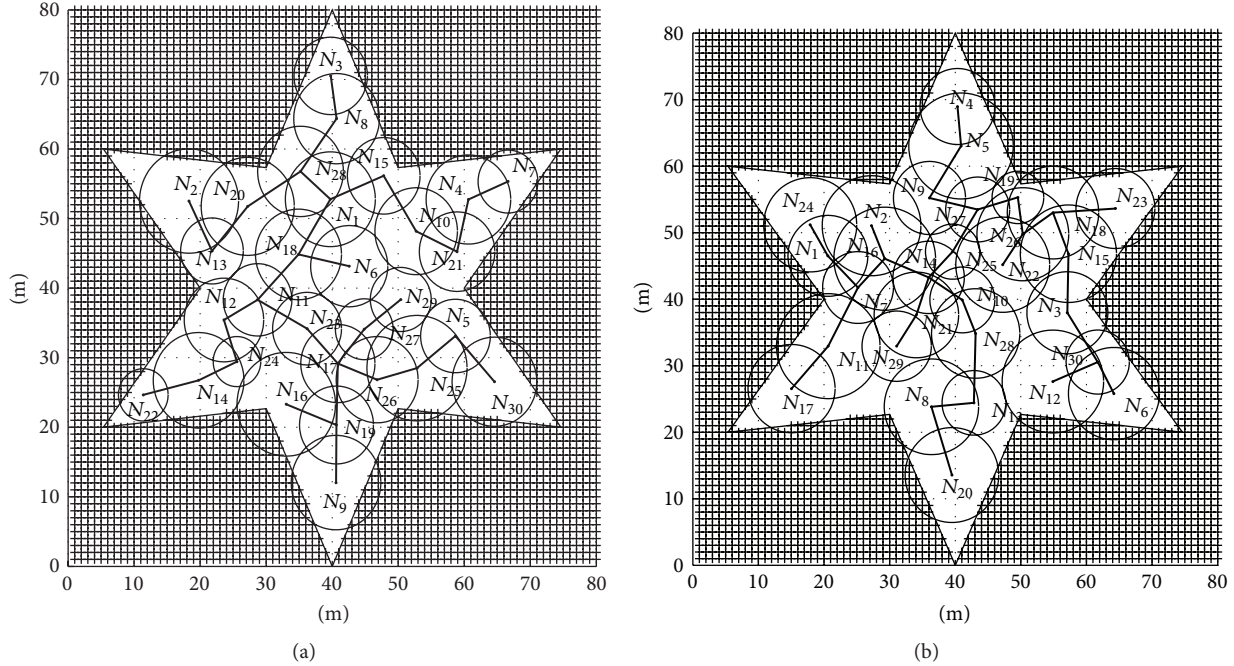


FIGURE 2: Examples of final distribution inner star, (a) fixed criterion, (b) variable criterion.

of the variable criterion (Figure 2(b)) show that the nodes are distributed; nevertheless the overlap is increased.

Table 1 shows examples of fixed and variable criteria with 30 nodes and effective area of  $2399.9200 \text{ m}^2$  under the scenario shown in Figure 2. It can be seen that both cases converge to 100 generations. Even when the fixed case presents better metrics than the variable one, the time spent to compute the solution is larger (fixed case has 889.1790 (s) while variable case has 546.9349 (s)). To state and prove this trend, it is needed to perform more tests.

Table 2 shows the results using 30 nodes and an effective area of  $2399.9200 \text{ m}^2$  for the tests numbers 31/40 and 35/40 of the 40 tests under the scenario shown in Figure 2; the mean and variance values of time, radii, percentage of redundancy, energy, and the covered area of the network were obtained. Here, both fixed and variable criteria are compared under the same conditions. The results are the average of these tests using 100 generations. The variable criterion is faster than the fixed one in 1.6527 times. It can be explained because the separation among the nodes (*scale* parameter) is relaxed for the variable criteria at most of the generations, even when the fixed criterion has a better solution than the variable one (which is reflected in the area coverage 89.66% versus 88.90%, energy 5.0037 (mW) versus 5.3347 (mW), and overlap 34.39% versus 40.94%).

For the next scenarios one will choose the variable criterion due to its high speed and low variance in terms of

TABLE 2: Inner star. Mean and variance results, variable and fixed criteria.

Test	Variable	Fixed	Gen.
Success test/40	35/40	31/40	100
Mean time (s)	544.8065	900.4085	100
Mean redundant (%)	40.94	34.39	100
Mean radii (m)	5.8493	5.6670	100
Mean energy (mW)	5.3347	5.0037	100
Mean area (%)	88.90	89.66	100
Var time	6.6988	117.3945	100
Var redundant	0.0016	0.0019	100
Var radii	0.0330	0.0294	100
Var energy	0.0955	0.0877	100
Var area	$2.3411e - 004$	$2.6960e - 004$	100

area coverage and time execution in the search of the solution space. The solution here obtained with the variable criterion is as good as the fixed one. Adequate tuned parameters and weights will help the variable criteria to be near the fixed one.

It is important to note that Figure 2(a) shows only the optimized location of the sensor nodes. The initial random location and the final optimized location of sensor nodes in Figure 2(a) in terms of  $(x, y)$  coordinates are shown in Table 3.

TABLE 3: Inner star, fixed criterion. Initial and final location of sensors nodes.

Node	$X_{\text{initial}}$ (m)	$Y_{\text{initial}}$ (m)	$X_{\text{final}}$ (m)	$Y_{\text{final}}$ (m)
$N_1$	38.9	45.8	39.7	52.8
$N_2$	14.2	58.7	18.3	52.5
$N_3$	42.8	73.4	39.8	70.5
$N_4$	60.9	49.0	60.6	52.7
$N_5$	54.2	40.1	58.7	33.1
$N_6$	40.4	40.6	42.6	43.1
$N_7$	60.4	58.2	66.6	55.3
$N_8$	41.6	62.8	40.7	64.4
$N_9$	43.8	13.1	40.6	12.0
$N_{10}$	52.4	51.8	52.7	48.2
$N_{11}$	28.8	39.2	28.7	38.3
$N_{12}$	28.4	36.0	23.6	35.4
$N_{13}$	15.3	47.1	21.8	45.3
$N_{14}$	28.0	26.5	19.8	26.7
$N_{15}$	47.1	57.7	47.8	56.1
$N_{16}$	27.5	24.1	33.0	23.3
$N_{17}$	41.7	29.0	40.9	29.2
$N_{18}$	18.0	49.0	34.9	49.8
$N_{19}$	37.9	19.5	40.6	20.3
$N_{20}$	19.8	51.4	27.2	51.5
$N_{21}$	58.3	43.8	58.9	45.2
$N_{22}$	9.5	25.3	11.4	24.7
$N_{23}$	31.9	35.7	36.2	34.2
$N_{24}$	27.1	28.1	25.6	29.4
$N_{25}$	46.8	28.9	52.8	28.5
$N_{26}$	41.2	21.9	46.8	26.8
$N_{27}$	29.7	28.2	44.9	34.1
$N_{28}$	25.8	54.2	35.2	56.8
$N_{29}$	48.2	49.9	50.4	38.4
$N_{30}$	57.3	39.0	65.6	26.5

Now, four sets of walls are analyzed. The results were obtained using 100 generations for the first, second, and fourth set of walls and 120 generations for the third set of walls in the DEA.

**5.2. First Set of Walls.** The set of walls shown in Figure 3 can be constructed by setting the walls through the coordinates  $x_a = 56 : 80$ ,  $y_a = 40$ ,  $x_b = 50$ ,  $y_b = 0 : 24$ ,  $x_c = 30$ ,  $y_c = 0 : 24$ ,  $x_d = 0 : 24$ ,  $y_d = 40$ ,  $x_e = 30$ ,  $y_e = 56 : 80$ , and  $x_f = 50$ ,  $y_f = 56 : 80$ . The distribution of the nodes is affected by the position of the walls. It is also noted that the final tree avoids crossing the walls, accomplishing the restriction of not crossing any obstacle. Some nodes lost their area coverage because of the walls; nevertheless other nodes can compensate that effect by covering the affected area. It is important to note that the distances between the walls and nodes 28 and 59 are 70 (cm) and 120 (cm), respectively. The circles are arranged far from the area boundary.

Table 4 shows the results obtained in the test number 5 of the 40 tests under the scenario shown in Figure 3. The covered

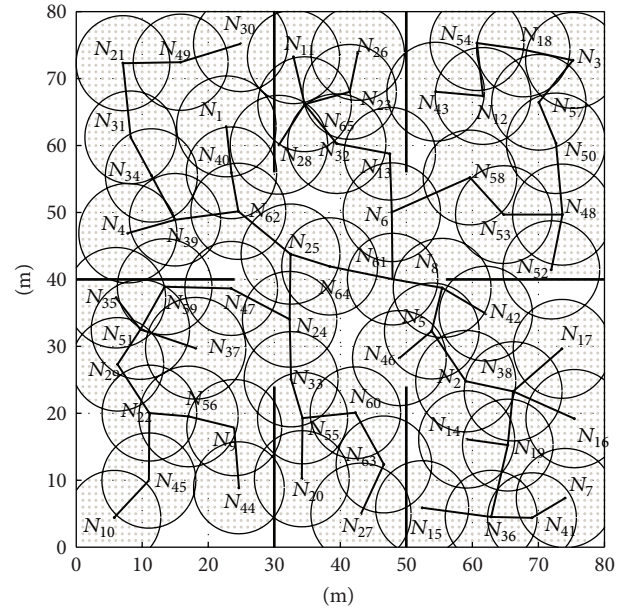


FIGURE 3: Final distribution of nodes for the first set of walls.

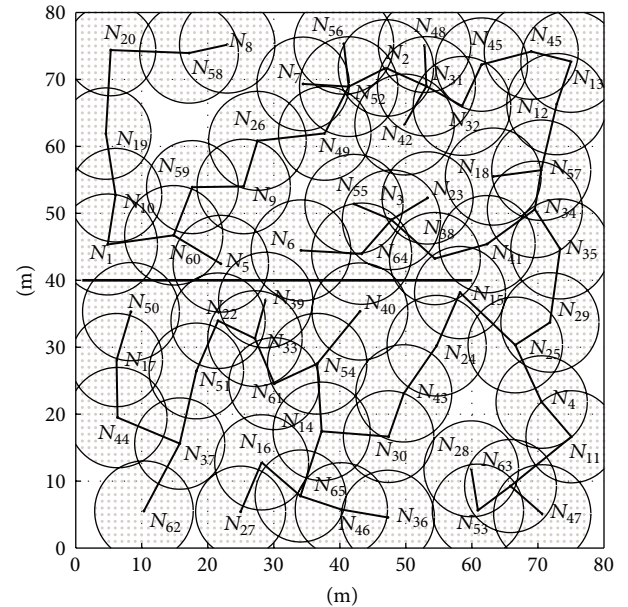


FIGURE 4: Final distribution of nodes for the second set of walls.

area is greater than 92%, the energy is 16.8013 (mW), and the redundancy is lower than 53%.

**5.3. Second Set of Walls.** The wall shown in Figure 4 can be constructed by setting the wall through the coordinates  $x_a = 1 : 60$ ,  $y_a = 45$ . It can be seen that the MST goes around the wall and the nodes are arranged away from the boundaries.

Table 5 shows the obtained parameters with the test number 24 of the 40 tests under the scenario shown in Figure 4. The area is greater than 93%, the energy is 16.8589 (mW), and the redundancy is lower than 52%.



TABLE 4: Results for the first set of walls, test # 5.

Criteria	Area%	Energy (mW)	Radii (m)	Redundancy%	Time (s)	Gen.
Variable	92.18	16.8013	7.1866	52.24	1226.3465	100

TABLE 5: Results for the second set of walls, test # 24.

Criteria	Area%	Energy (mW)	Radii (m)	Redundancy%	Time (s)	Gen.
Variable	93.87	16.8589	7.1981	51.32	1498.9526	100

TABLE 6: Results for the third set of walls, test 38.

Criteria	Area%	Energy (mW)	Radii (m)	Redundancy%	Time (s)	Gen.
Variable	92.88	16.5964	7.1421	50.69	1739.6244	120

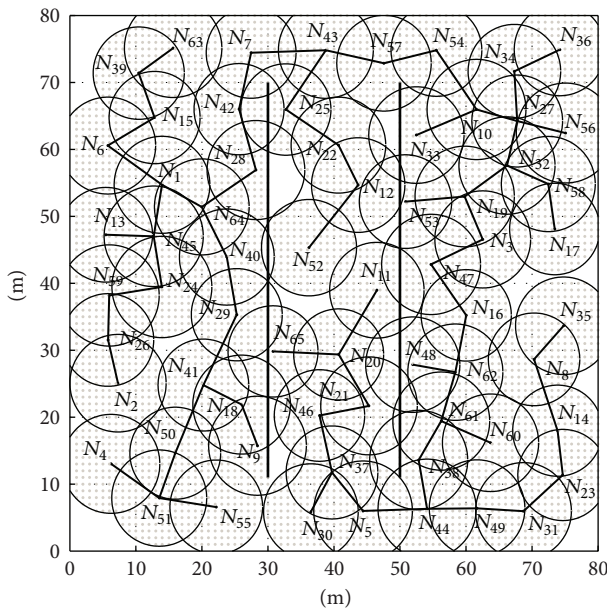


FIGURE 5: Final distribution of nodes for the third set of walls.

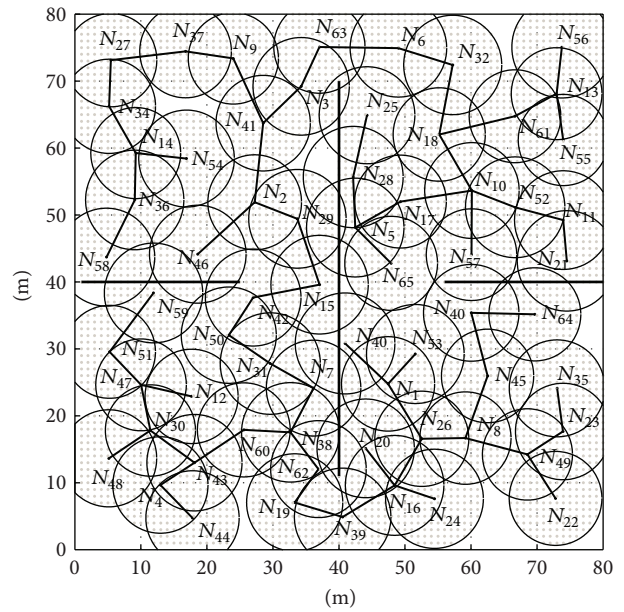


FIGURE 6: Final distribution of nodes for the fourth set of walls.

5.4. *Third Set of Walls.* Case three was configured as  $x_a = 30$ ,  $y_a = 11 : 70$ ,  $x_b = 50$ , and  $y_b = 11 : 70$ , which seems as a set of parallel walls, shown in Figure 5. It can be seen that the tree goes around the wall and the nodes are arranged away from the boundaries. Here, also, it is important to note that the distances between the walls and nodes 53 and 65 are 80 (cm) and 70 (cm), respectively.

Table 6 shows the obtained parameters with the test number 38 of the 40 tests under the scenario shown in Figure 5. The area is greater than 92%, the energy is 16.5964 (mW), and the redundancy is lower than 51%. Here, a case is presented where the number of generations is larger compared to the established one, 120 generations instead of 100 generations. As it can be shown through this paper, sometimes the algorithm needs more generations to obtain a tree.

5.5. *Fourth Set of Walls.* Case four was configured as  $x_a = 1 : 25$ ,  $y_a = 40$ ,  $x_b = 76 : 80$ ,  $y_b = 40$ , and  $x_c = 40$ ,  $y_c = 26 : 75$  which seems as a divided corridors inside a room, shown in

Figure 6. It can be seen that the tree goes around the wall and the nodes are arranged away from the boundaries.

Table 7 shows the obtained parameters with case 38 of the 40 tests under the scenario shown in Figure 6. The area is greater than 93%, the energy is 16.7526 (mW), and the redundancy is lower than 52%.

It is important to remember that, to locate one sensor respect any other one, it is simple through the branches of the MST. For example, the distance between sensor nodes  $N_7$  and  $N_{20}$ , in Figure 6, is not the euclidian distance between them. It is necessary to follow the route  $N_7 \rightarrow N_{38} \rightarrow N_{62} \rightarrow N_{19} \rightarrow N_{39} \rightarrow N_{16} \rightarrow N_{20}$  and add each distance to obtain the total distance between sensor nodes  $N_7$  and  $N_{20}$ .

5.6. *Summary.* Table 8 shows a comparison summary of the parameters obtained with DEA such as the generation number in which the sensors configuration was obtained, the area covered, the energy average of the network, and the degree of overlap. The same 40 tests were done for each

TABLE 7: Results for the fourth set of walls, test 38.

Criteria	Area%	Energy (mW)	Radii (m)	Redundancy%	Time (s)	Gen.
Variable	93.19	16.7526	7.1761	51.99	1480.5491	100

TABLE 8: Mean and variance results for the four set of walls, with 65 nodes and effective area of 6400.0000 (m<sup>2</sup>).

Case	Case 1	Case 2	Case 3	Case 4
Success test/40	34/40	40/40	36/40	39/40
Mean time (s)	1228.6005	1417.2937	1531.0622	1515.7459
Mean redundant (%)	50.13	53.05	50.50	51.40
Mean radii (m)	7.1963	7.1891	7.1879	7.1858
Mean energy (mW)	16.8495	16.8144	16.8092	16.7988
Mean area (%)	92.25	93.83	93.24	93.41
Var time	4.4281	1187.1325	1735.6760	1129.1838
Var redundant	$1.7137e - 004$	$1.2889e - 004$	$1.8643e - 004$	$1.7708e - 004$
Var radii	$9.4720e - 004$	$4.4480e - 004$	$9.6199e - 004$	$6.5851e - 004$
Var energy	0.0213	0.0097	0.0210	0.0143
Var area	$3.2485e - 005$	$1.9754e - 005$	$2.9491e - 005$	$3.6057e - 005$

case, using 100 generations. The area average is larger than 93% with an energy average less than 16.85 (mW) and the redundancy average less than 54%. However, as it can be shown in Table 8, the covered area is quite similar in all cases. The rest of the nodes near the wall are compensated with the nodes at the other side of the wall.

## 6. Conclusions

This research work shows the feasibility to apply the DEA in presence of obstacles or strict restrictions. Results obtained with the inner star show that the circles can be arranged even into the peaks of the star by using either large or small circles. Results could be improved if more restrictions are included, for instance, the area management on the peaks, where the smallest circle can be placed instead of the largest circles. Even so, the results seem promising.

With respect to the presented cases with obstacles, the construction of the trees and the distribution and location of the nodes are feasible. The trees avoid crossing the obstacles and the circles have a separation among them and regarding the area boundaries. This is shown because the area obtained is greater than 90% for all the cases with a maximum overlap of 51%. The presented results can be improved if a restriction regarding the obstacles is adopted moving the circles away of the barriers, obtaining more useful area for each circle. Nonetheless, the DEA minimizes the wasted area by putting the neighbor nodes of the circle that lose its area coverage. This work is the initial step to study the distribution of sensor nodes in presence of obstacles, represented in this work as a set of given walls. An accurate modelling regarding the signal propagation and a larger separation among the nodes could also improve the results presented in this research work. As further work it is recommended to implement the flocking of multiagents with virtual leader technic described in Section 1 in order to move the sensors in the network considering the position and velocity of the centre of mass

of all sensors (agents). Also, it is important to implement the case with multiple leaders (containment control) focused on both switching networks and fixed networks described in Section 1, in which the input signals of sensors may be saturated in practice.

## Conflict of Interests

The authors declare that they have no competing interests.

## Acknowledgment

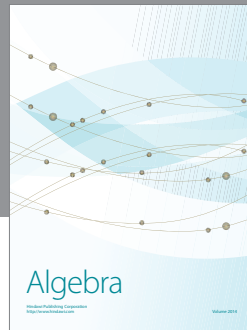
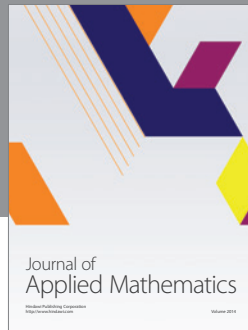
This research was partially supported by School of Engineering and Sciences of Tecnológico de Monterrey.

## References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.
- [2] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Computing Surveys*, vol. 37, no. 2, pp. 164–194, 2005.
- [3] K. P. Eswaran and R. E. Tarjan, "Augmentation problems," *SIAM Journal on Computing*, vol. 5, no. 4, pp. 653–665, 1976.
- [4] W.-T. Chen and N.-F. Huang, "The strongly connecting problem on multihop packet radio networks," *IEEE Transactions on Communications*, vol. 37, no. 3, pp. 293–295, 1989.
- [5] L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc, "Power consumption in packet radio networks," *Theoretical Computer Science*, vol. 243, no. 1-2, pp. 289–305, 2000.
- [6] M. X. Cheng, M. Cardei, J. Sun et al., "Topology control of ad hoc wireless networks for energy efficiency," *IEEE Transactions on Computers*, vol. 53, no. 12, pp. 1629–1635, 2004.
- [7] B. Xiao, L. Chen, Q. Xiao, and M. Li, "Reliable anchor-based sensor localization in irregular areas," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 60–72, 2010.

- [8] W. Shi and V. W. S. Wong, "A two-phase algorithm for locating sensors in irregular areas," in *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM '10)*, pp. 1–6, Miami, Fla, USA, December 2010.
- [9] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1333–1344, 1999.
- [10] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, "A cone-based distributed topology-control algorithm for wireless multi-hop networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 1, pp. 147–159, 2005.
- [11] A. S. Majid and E. Joelianto, "Optimal sensor deployment in non-convex region using discrete particle swarm optimization algorithm," in *Proceedings of the 1st IEEE Conference on Control, Systems and Industrial Informatics (ICCSII '12)*, pp. 109–113, IEEE, Bandung, Indonesia, September 2012.
- [12] N. A. B. Ab Aziz, A. W. Mohemmed, and M. Y. Alias, "A wireless sensor network coverage optimization algorithm based on particle swarm optimization and voronoi diagram," in *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC '09)*, pp. 602–607, Okayama, Japan, March 2009.
- [13] H. Safa, W. El-Hajj, and H. Zoubian, "Particle swarm optimization based approach to solve the multiple sink placement problem in WSNs," in *Proceedings of the IEEE International Conference on Communications (ICC '12)*, pp. 5445–5450, Ottawa, Canada, June 2012.
- [14] D. S. Deif and Y. Gadallah, "Classification of wireless sensor networks deployment techniques," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 2, pp. 834–855, 2014.
- [15] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: a brief survey," *IEEE Transactions on Systems, Man and Cybernetics—Part C: Applications and Reviews*, vol. 41, no. 2, pp. 262–267, 2011.
- [16] A. Tamizharasi, R. Arthi, and K. Murugan, "Bio-inspired algorithm for optimizing the localization of wireless sensor networks," in *Proceedings of the 4th International Conference on Computing, Communications and Networking Technologies (ICCCNT '13)*, pp. 1–5, Tiruchengode, India, July 2013.
- [17] X. Zhang, T. Wang, and J. Fang, "A node localization approach using particle swarm optimization in wireless sensor networks," in *Proceedings of the International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI '14)*, pp. 84–87, IEEE, Beijing, China, October 2014.
- [18] H. Su, X. Wang, and Z. Lin, "Flocking of multi-agents with a virtual leader," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 293–307, 2009.
- [19] H. Su and M. Z. Chen, "Multi-agent containment control with input saturation on switching topologies," *IET Control Theory & Applications*, vol. 9, no. 3, pp. 399–409, 2015.
- [20] H. Su, G. Jia, and M. Z. Q. Chen, "Semi-global containment control of multi-agent systems with intermittent input saturation," *Journal of the Franklin Institute*, vol. 352, no. 9, pp. 3504–3525, 2015.
- [21] A. Nordio and C.-F. Chiasserini, "Field reconstruction in sensor networks with coverage holes and packet losses," *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3943–3953, 2011.
- [22] Z. Mi, Y. Yang, and H. Ding, "Self-organized connectivity control and optimization subjected to dispersion of mobile ad hoc sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 672436, 15 pages, 2012.
- [23] L. Liu and H. Ma, "On coverage of wireless sensor networks for rolling terrains," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 118–125, 2012.
- [24] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [25] V. Feoktistov, *Differential Evolution: In Search of Solutions*, Springer Optimization and Its Applications, Springer, New York, NY, USA, 2006.
- [26] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
- [27] L. Jin, J. Jia, and D. Sun, "Node distribution optimization in mobile sensor network based on multi-objective differential evolution algorithm," in *Proceedings of the 4th International Conference on Genetic and Evolutionary Computing (ICGEC '10)*, pp. 51–54, IEEE, Shenzhen, China, December 2010.
- [28] J.-H. Seok, J.-Y. Lee, C. Oh, J.-J. Lee, and H. J. Lee, "RFID sensor deployment using differential evolution for indoor mobile robot localization," in *Proceedings of the 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 3719–3724, Taipei, Taiwan, October 2010.
- [29] S. Hojjatoleslami, V. Aghazarian, and A. Aliabadi, "DE based node placement optimization for wireless sensor networks," in *Proceedings of the 3rd International Workshop on Intelligent Systems and Applications (ISA '11)*, pp. 1–4, Wuhan, China, May 2011.
- [30] D. Qiao and G. Pang, "A modified differential evolution with heuristics algorithm for non-convex optimization on sensor network localization," *IEEE Transactions on Vehicular Technology*, 2015.
- [31] R. Hari Krishnan, V. J. S. Kumar, and P. S. Ponmalar, "Differential evolution approach for localization in wireless sensor networks," in *Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (ICIC '14)*, pp. 1–4, Coimbatore, India, December 2014.
- [32] J. Kuriakose, V. Amruth, and N. S. Nandhini, "A survey on localization of wireless sensor nodes," in *Proceedings of the International Conference on Information Communication and Embedded Systems (ICICES '14)*, pp. 1–6, Chennai, India, February 2014.
- [33] L. Cheng, C. Wu, Y. Zhang, H. Wu, M. Li, and C. Maple, "A survey of localization in wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 962523, 12 pages, 2012.
- [34] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.
- [35] U. K. Chakraborty, S. K. Das, and T. E. Abbott, "Energy-efficient routing in hierarchical wireless sensor networks using differential-evolution-based memetic algorithm," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '12)*, pp. 1–8, Brisbane, Australia, June 2012.
- [36] H. Ali, W. Shahzad, and F. A. Khan, "Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization," *Applied Soft Computing Journal*, vol. 12, no. 7, pp. 1913–1928, 2012.
- [37] T. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2001.
- [38] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

- [39] F. Lezama, G. Castañón, and A. M. Sarmiento, "Differential evolution optimization applied to the wavelength converters placement problem in all optical networks," *Computer Networks*, vol. 56, no. 9, pp. 2262–2275, 2012.
- [40] R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957.
- [41] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [42] R. S. Cahn, *Wide Area Network Design: Concepts and Tools for Optimization*, The Morgan Kaufmann Series in Networking, Morgan Kaufmann, Boston, Mass, USA, 1998.



# Hindawi

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

