Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 7873984, 22 pages
https://doi.org/10.1155/2018/7873984

Research Article

Multiobjective Optimization for a Wireless Ad Hoc Sensor Distribution on Shaped-Bounded Areas

Armando Céspedes-Mota,[1] **Gerardo Castañón**,[1]
Alberto F. Martínez-Herrera,[1] **and Leopoldo Eduardo Cárdenas-Barrón**[2]

[1]*Department of Electrical and Computer Engineering, Tecnológico de Monterrey, 2501 Eugenio Garza Sada Sur, 64849 Monterrey, NL, Mexico*
[2]*School of Engineering and Sciences, Tecnológico de Monterrey, 2501 Eugenio Garza Sada Sur, 64849 Monterrey, NL, Mexico*

Correspondence should be addressed to Leopoldo Eduardo Cárdenas-Barrón; lecarden@tec.mx

Received 30 January 2018; Revised 6 April 2018; Accepted 2 May 2018; Published 3 June 2018

Academic Editor: Abhishek K. Gupta

Resource efficiency in wireless ad hoc networks has become a widely studied NP-problem. This problem may be suboptimally solved by heuristic strategies, focusing on several features like the channel capacity, coverage area, and more. In this work, maximizing coverage area and minimizing energy consumption are suboptimally adjusted with the implementation of two of Storn/Price's Multiobjective Differential Evolution (DE) algorithm versions. Additionally, their extended representations with the use of random-M parameter into the mutation operator were also evaluated. These versions optimize the initial random distribution of the nodes in different shaped areas, by keeping the connectivity of all the network nodes by using the Prim–Dijkstra algorithm. Moreover, the Hungarian algorithm is applied to find the minimum path distance between the initial and final node positions in order to arrange them at the end of the DE algorithm. A case base is analyzed theoretically to check how DE is able to find suboptimal solutions with certain accuracy. The results here computed show that the inclusion of random-M and completion of the algorithm, where the area is pondered with 60% and the energy is pondered with 40%, lead to energy optimization and a total coverage area higher than 90%, by considering the best results on each scenario. Thus, this work shows that the aforementioned strategies are feasible to be applied on this problem with successful results. Finally, these results are compared against two typical bioinspired multiobjective algorithms, where the DE algorithm shows the best tradeoff.

1. Introduction

Nowadays, it is well known that the distribution of sensors in a wireless ad hoc sensor network (WAHSN) is a challenge in the research of wireless communications, because it is necessary to increase both lifetime and coverage area demand of the network. Furthermore, it is required to avoid a weak connectivity.

A WAHSN contains a number of geographically dispersed mobile sensors. Each sensor node has wireless communication, collaborative signal processing, and network capabilities subject to some technical constraints. The mobile sensor nodes are free to move and self-organize in an arbitrary way while the communication is made via radio transceivers. Therefore, a WAHSN is able 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. In consequence, it is necessary that the required data be disseminated to the proper end users [1, 2]. It is important to remark that wireless sensor networks (WSN) are a type of ad hoc network composed of a large number of sensor nodes, which are densely deployed either inside the sense perception area or very close to it. The nodes are not connected to each other. Basically, they are stationary or at most slowly moving [1]. In contrast, a mobile ad hoc network (MANET) is an autonomous collection of mobile users connected by one to one wireless links. The MANETs topology may change continuously in an unpredictably way over time. The mobile nodes in MANETs can renew, replace, or recharge

their batteries or source of energy. Although WAHSNs, WSNs, and MANETs involve multihop communications, many end-to-end routing schemes proposed for MANETs are inappropriate for WAHSN and WSNs for the previously mentioned reasons [3]. Nevertheless, sensor distribution schemes between WAHSNs and MANETs can be compared. A key factor that distinguishes the WAHSNs from MANETs is that the end goal is detection and estimation of some events of interest and not just communication [1, 2].

In spite of the fact that our work is applicable to WAHSNs, MANETs, and also WSNs under certain modifications, the analysis on WAHSNs due to some characteristics and resource constraints such as energy, coverage area, and connectivity optimization is made.

Sensor distribution is one of the fundamental problems in WAHSNs and regularly has been suboptimally solved by heuristic strategies such as genetic algorithms, among several others as shown by [4–8]. It is worth mentioning that sensor distribution with minimum energy connectivity is an NP-complete problem [9–12]. In the literature, different strategies for wireless ad hoc and sensor networks related to topology control [2], node localization [13, 14], distributed coverage [15], and network lifetime [16] are described to provide solutions for sensor distribution, coverage area, and energy consumption problems. Additional related problems in wireless communication such as frequency allocation and interference have been also solved by bioinspired algorithms as it is reported in [17, 18] or in [19], where the authors show an efficient way of solving the minimum number of hops among nodes to be communicated.

To optimize the sensor distribution, DE algorithm is used. DE is a very simple but very powerful stochastic global optimizer for a continuous search domain. It was proposed by [20] and represents a very complex evolution process. Cleverly using the differences between the populations, a simple but fast linear operator called differentiation is created, which makes DE unique. A survey of the applications of DE to different optimization problems shows that DE generally outperforms other evolutionary algorithms [21–23]. An updated reference with novel DE algorithm versions and additional applications are found at [24, 25]. DE exploits a population of potential solutions to effectively probe the search spaces. The algorithm is initialized with a population of random candidate solutions, which are conceptualized as individuals. In this work, an individual includes the coordinates of the sensor nodes (x, y) followed by its communication radii values. For each individual in the population, a descendant is created from three parents. One parent, called the main parent, is disturbed by the vector of the difference of the other two parents. If the descendant has a better performance as measured by the objective function, it replaces the individual; otherwise, the original individual is retained and is passed on to the next generation of the algorithm and the descendant is discarded. This process is repeated until it reaches the termination condition. For a complete theoretical analysis of the DE algorithm, the reader is referred to [20].

Node distribution optimization in mobile sensor networks is presented in [26], where they analyze the sensor network coverage area and its coverage redundant area using DE. Moreover, the optimal power allocation in WSNs using DE is given in [27]. Also, in [28], DE is applied to maximize the lifetime of the WSNs using disjoint sets of sensors to cover a set of targets. However, it is important to mention that [26–28] do not consider ad hoc sensor networks. They use the DE algorithm only for WSNs. On the other hand, a DE based topology control mechanism in MANETs is presented in [29] to deploy the nodes with the maximum coverage area. Here, the topology control is managed with the use of a Markov chain model. A similar approach is followed in [30] to evaluate the node distribution, but now based on Voronoi regions for each node. However, [29, 30] do not consider optimization of energy consumption for maintaining the network connectivity and coverage area simultaneously in a multiobjective fitness function.

The approach followed in [31] analyzed the coverage area of sensor networks, where the main goal is to maximize the coverage area in presence of obstacles such as walls. In [32] the same approach of maximizing the coverage area in the presence of obstacles is extended, including the energy consumed in the objective function. The redundant area is also considered in the objective function. In [33], an improved DEA version based on a best-optimal-solution sorting technique is compared with the classical DEA and the Nondominated Sorting Genetic Algorithm II (NSGA-II), showing that the first one has the best performance when computing the coverage area. Additionally, the number of nodes needed to obtain the maximum coverage area is diminished to determine the minimum feasible. In [34], the coverage area is optimized by using the improved versions of DEA such as the Self-Adaptive Differential Evolution (jDE) and the Adaptive Differential Evolution (JADE). As expected, the jDE and JADE have better performance than the classical DEA versions. In [35], DEA is applied for node position estimation in convex and nonconvex configurations.

Other aspects of wireless sensor networks can be tackled by implementing other bioinspired algorithms. For instance, in [6], the coverage area is optimized with Particle Swarm Optimization (PSO). Coverage area and energy are considered in a multiobjective version of PSO as shown in [5], where the parameters are controlled with a clustering approach. For routing management, Ant Colony is implemented in [36] as well as in [8]. Routing is also solved by the Strength Pareto Evolutionary algorithm (SPEA2) as shown in [37]. Optimal path problem is treated in [38], where DEA is the best solution compared to the classical PSO and GA strategies. Node anchors can be deployed with bioinspired techniques as shown in [39] with Ant Colony. Such scenario is found when RFID readers are deployed to detect tags. In [40], a combination of GA and PSO is generated to do so.

Additionally, in [41], two multiobjective algorithms are compared, the Multiobjective Simulated Annealing (MOSA) and NSGA-II. Such comparison is done to determine the algorithm that gives the best distribution output with fewer nodes. In [42], two algorithms are compared to solve WSN scenarios: SPEA2 and NSGA-II. The objective function includes the energy consumption, the coverage area, and the network reliability, where the energy component is analyzed against the elapsed time. For scenarios where the complexity

is increased, SPEA2 is better than NSGA-II and vice versa. In [43] different WSN scenarios are tested, where NSGA-II and Learning Automata (LA) are combined to create a hybrid multiobjective algorithm. This hybrid version outperforms NSGA-II. The main goal is to minimize the number of active sensors that cover the maximum possible area and minimize the energy consumption while maintaining the network connectivity. In [44] the Multiobjective Evolution algorithm based on Decomposition (MOEA/D) and NSGA-II are compared to determine which is the best option to maximize the coverage area and minimize the energy consumption in WSN, where it is claimed that MOEA/D gives better results than NSGA-II. In [45], three optimization strategies are evaluated, the Multiobjective Particle Swarm Optimization (MOPSO), NSGA-II, and a proposed optimization denoted as Heuristic 3-Phases (H3P). H3P is formed by 3 steps: clustering, Pareto construction, and disassembly. The goal is to minimize the number of sensors when maximizing the number of targets to be detected, where H3P presents the best performance among the presented optimization strategies. In [46], NSGA-II is implemented and compared with MOEA/D to obtain the best routing combination by minimizing the energy consumption while detecting a large number of targets as possible, where it is shown that NSGA-II is better than MOEA/D. In [47], a discrete DEA version and NSGA-II are implemented to reach the best compromise among three goals: the delay of a packet from the origin to the destiny in the network, the packet delivery ratio, and the flow conservation. It is done to help the routing process with time efficiency, where it is shown that DEA outperforms NSGA-II. A detailed survey for more available techniques applied on WSN can be consulted in [48].

Compared to the existing bioinspired techniques in the open literature [26, 32] and the method here presented, the followed approach is based on a simplified version of the MOEA/D techniques presented in [49], where a set of weights is assigned on each function to be optimized and the summation of them is equal to 1. Such variant is known as *Weighted Sum Approach* [49, 50]. For the sake of clarity, the method here implemented is denoted as Multiobjective Differential Evolution algorithm (MODEA).

MODEA is applied to the optimization of network energy consumption to maintain the network connectivity and network coverage area in WAHSNs, following a convex model [35]. The algorithm makes sensor radius adjustments and assumes the mobility of the sensors in order optimize two objectives: maximizing the network coverage area and reducing network energy consumption. It is worth mentioning that consequently also the network lifetime is extended.

Additionally, the proposed algorithm emphasizes the successful communication between sensors in a given neighborhood, with the use of the Prim–Dijkstra algorithm described by [51, 52]. If a tree is not created, the nodes distribution is discarded even when the network fulfills the rest of the restrictions regarding area, energy, and bounds on different shaped areas. Also, the Hungarian algorithm [53] is used to find the shortest distance between the initial and the final node positions. This information is important for the case of having mobile ad hoc sensors; each node knows where to move with the minimal distance.

A summary of the employed techniques and the involved parameters is shown in Table 1.

The rest of this paper is organized as follows: Section 2 presents the methodology. Section 3 describes theoretical models to find the energy lower bound and coverage area's upper bound. Section 4 develops and explains the MODEA for wireless ad hoc sensor distribution. Section 5 provides numerical results and a discussion. Finally, Section 6 gives the conclusions and some future research directions.

2. Description of the Method

2.1. Sensor Coverage Model. The coverage area of a sensor is the region within which the sensor is able to detect or analyze the sensing parameters. There exists a special interest in the sensor distribution for which communication with neighboring nodes is more energy efficient. Also, it is assumed that the coverage radius of each sensor helps to determine its communication link distance through minimum spanning tree to any of its neighbors, and it depends on the kind of sensors considered. Additionally, the sensor distribution on different shaped areas is considered. The main objective of the sensor coverage model is to achieve a balance between the maximum effective coverage area and the minimum communication sensor energy.

According to [26], the node set on the target area $m \times n$ grid can be defined as

$$\mathcal{N} = \{N_1, N_2, \ldots, N_z\}, \tag{1}$$

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

The coverage range of a node N_i can be expressed as a circle centered at its coordinates (x_i, y_i) with sensing radius r_i. The (x_i, y_i) coordinates express the position of sensor nodes and the subindex i represents the sensor node index. A grid point $P(x, y)$ is covered by a sensor node i if and only if its distance to the center (x_i, y_i) of the circle (2) is not larger than the sensing radius r_i:

$$d(N_i, P) = \sqrt{(x - x_i)^2 + (y - y_i)^2}. \tag{2}$$

First of all, a binary detection model is considered. A random variable E_i is introduced to describe the event in which the sensor node i covers a given point $P(x, y)$. The probability of the event E_i, $P\{E_i\}$, is equal to the coverage probability $P_{\text{cov}}(x, y, N_i)$. This probability is a binary valued function such as

$$P_{\text{cov}}(x, y, N_i) = \begin{cases} 1 & \text{if } (x - x_i)^2 + (y - y_i)^2 \leq r_i^2 \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

This model is applied in all cases in Section 5. It is clear that a point should be evaluated against all the nodes present in the network; then

$$S(x, y) = \sum_{i=1}^{z} P_{\text{cov}}(x, y, N_i), \tag{4}$$

Table 1: Summary of the references presented in Section 1.

ID	Cite	Technique	Involved variables
DE	[26]	DE/rand/1	Coverage area, energy
	[27]	DE/rand/1, DE/Best/1	Power allocation
	[27]	DE/Curr-to-Best/1, DE/rand-to-best/1	Power allocation
	[28]	DE/Best/1	Coverage area
	[29]	DE/rand/1, Markov topology control	Coverage area
	[30]	DE/rand/1, Voronoi topology control	Coverage area
	[31]	DE/rand/1	Coverage area, redundant area
	[32]	DE/rand/1 with rand M	Coverage area, energy, redundant area
	[33]	Classical and modified DE/rand/1, NSGA-II	Coverage area with node elimination
	[34]	DE/Best/1, jDE, JADE	Coverage area
	[35]	Modified DE/Curr-to-Best/1	Node position estimation
	[47]	DEA and NSGA-II	Routing
Others	[5]	PSO	Energy and coverage area (by clustering)
	[6]	PSO	Coverage area
	[8]	Ant Colony	Routing
	[36]	Ant Colony	Routing
	[37]	SPEA2	Routing
	[38]	DE, PSO, GA	Optimal path
	[39]	Ant Colony	Node deployment (target coverage)
	[40]	Combined GA and PSO	Node deployment (target coverage)
	[41]	MOSA and NSGA-II	Node deployment
	[42]	SPEA-2 and NSGA-II	Energy, coverage area
	[43]	NSGA-II and LA	Energy, coverage area
	[44]	MOEA/D and NSGA-II	Energy, coverage area
	[45]	NSGA-II, MOPSO, H3P	Node deployment (target coverage)
	[46]	NSGA-II	Node deployment (target coverage), routing

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

Nevertheless, to obtain the proper covered area by the set of nodes, the function $R(x, y)$ is computed to verify if the coordinate is covered at least once or if it is not covered at all N_i.

$$R(x, y) = \begin{cases} 1 & S(x, y) \geq 1 \\ 0 & S(x, y) = 0, \end{cases} \tag{5}$$

and then the covered area is computed as

$$A_{\text{cov}} = \frac{\sum_{y=1}^{n} \sum_{x=1}^{m} R(x, y)}{m \times n}. \tag{6}$$

Equation (6) is implemented in our simulation by sweeping all the (x, y) points of the target area $m \times n$. For different area configurations, this area is replaced by the bounded area of interest.

2.2. Energy Consumption. There are models for efficient energy consumption in MANETs [5], WAHSNs [10], WSNs [54], and many others. The energy consumption model used in this work is based in [55], this model is also used in [26]. In order to show the optimization process in coverage area and energy consumption, the total energy for maintaining the network connectivity is considered as the total energy consumption of the network.

$$E_{\text{csd}} = \mu \sum_{i=1}^{z} r_i^{\alpha}, \tag{7}$$

where μ is the power per unit area in milliwatts per square meter (mW/m^2), considering that the sensors are on the floor, r_i is the sensor radius of the node i in meters, and $\alpha = 2$ since the communication medium is the air (free space) [56]. In this paper, μ equals 0.005 (mW/m^2). The path loss exponent α takes different values according the environment. For urban area cellular radio $\alpha = 2.7$–3.5, for shadowed urban cellular radio $\alpha = 3$–5, for in-building line-of-sight from 1.6 to 1.8, for obstructed in-building from 4 to 6, and for obstructed in factories from 2 to 2.3, the power per unit area changes in a 3D environment to mW/m^2, [56].

3. Base Case: Ideal Distribution of 10 Nodes into a Squared Area

In this section, the methods to obtain the energy and coverage area bounds are presented. In order to find these lower and

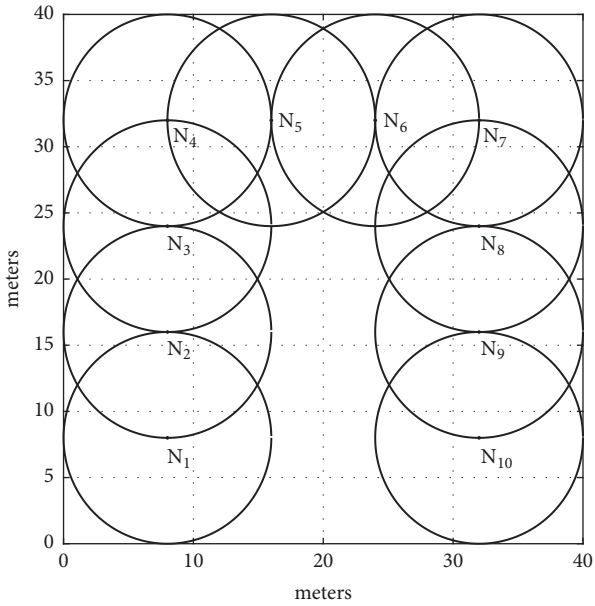

Figure 1: Node distribution to calculate upper bound.

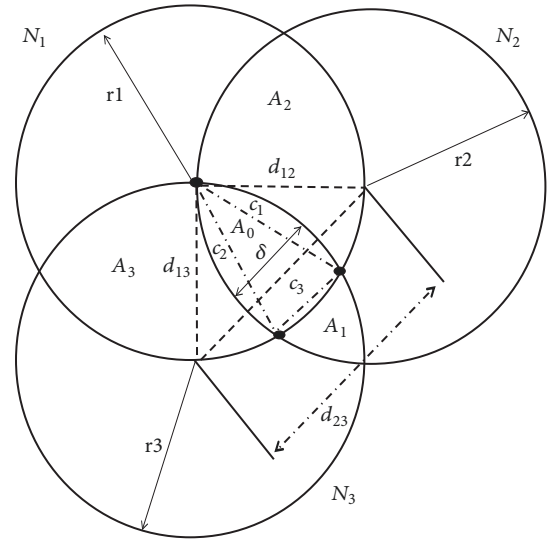

Figure 2: Parameters involved in the intersection of three circles.

upper bounds, an example of 10 nodes distributed in an area of $40 \times 40\,\mathrm{m}^2$—see Figure 1—is considered. It is assumed that the sensors can have a transmitting radius between 6 and 8 meters. Details to compute these bounds are described in the following subsections.

3.1. Energy Lower Bound. The energy lower bound is easily calculated using (7) considering a radius of 6 meters and $\mu = 0.005\,\mathrm{mW/m}^2$. For instance, assuming 10 nodes, the lowest allowed value for energy is $E_{\mathrm{LB}} = (0.005)(10)(6^2) = 1.8\,\mathrm{mW}$.

3.2. Area Upper Bound. The sensor distribution shown in Figure 1 is one among several that provides the nearest value to the maximum coverage area given by 10 nodes using a radius of 8 meters and including maximum three intersections of three circles. The main goal in proposing this case is their comparison to the suboptimal outputs derived of MODEA by having all the nodes' radii equal to 8 meters. If these suboptimal solutions are near to this theoretical case, then the application of MODEA will be considered feasible. Figure 1 is used as the base case model to determine its *theoretical* area upper bound.

At the top corners of Figure 1, the overlap of three circles is observed. Figure 2 shows the details of these three overlapped circles. The covered area of these three circles represents in this paper the coverage area of three overlapped wireless sensors. The set theory helps to determine the total area in a union of sets when some of the sets overlap. To obtain the covered area of the union of three overlapped circles, the inclusion/exclusion rule is used. This rule establishes that the area of the three circles is computed at the beginning. After that, all the intersection areas between two circles $A_{N_1 \cap N_2}$, $A_{N_1 \cap N_3}$, $A_{N_2 \cap N_3}$ are subtracted. At the end, the intersection area of the three circles $A_{N_1 \cap N_2 \cap N_3}$ is added [57, 58]. The equation of the area of the three circles is

$$A_{3C} = 3\pi r^2 - A_{N_1 \cap N_2} - A_{N_1 \cap N_3} - A_{N_2 \cap N_3} + \cdots + A_{N_1 \cap N_2 \cap N_3}, \tag{8}$$

where $A_{N_1 \cap N_2} = A_0 + A_2$, $A_{N_1 \cap N_3} = A_0 + A_3$, $A_{N_2 \cap N_3} = A_0 + A_1$ and $A_{N_1 \cap N_2 \cap N_3} = A_0$.

This result is then used to determine the area of the base case model of 10 sensors. For this model, the area of 10 sensors is computed; then, the nine intersection areas between two circles similar to $(A_0 + A_3)$ and the two intersection areas between two circles similar to $(A_0 + A_1)$ are subtracted; and, finally, the two intersection areas of three circles are added.

It is important to remark that for any finite number of overlapped circles, the covered area can be seen as a set of circles with intersections and is obtained by the use of the expression [57] as

$$A_{\mathrm{cov}} = \sum_{i=1}^{z} A_{N_i} - \sum_{i<j\le z}^{z} A_{N_i \cap N_j} + \cdots + \sum_{i<j<k\le z}^{z} A_{N_i \cap N_j \cap N_k} + (-1)^{z+1} A_{\bigcap_{i=1}^{z} N_i}. \tag{9}$$

According to [59], it is possible to preserve two communication links with an angle of $5\pi/6$ with the minimum power required, but it is valid for directional communication. For omnidirectional communication, it is not necessary. To obtain the maximum coverage area, it is well known that the hexagon configuration is the ideal distribution, where the angle presented on each vertex is equal to $2\pi/3$, but it is valid for circles with equal radius and where there is not a neighbor circle covering each node. Nevertheless, for our case, we have a set of circles with different radius and where each node is covered by at least a neighbor circle.

Now, to find the total area of Figure 1, first find the area of 10 independent circles A_{10C} given by

$$A_{10C} = 10\pi r^2, \tag{10}$$

and then use the intersection between two circles denoted as A_{i2C}, [60].

$$A_{i2C} = 2r^2 \arccos\left(\frac{d}{2r}\right) - \frac{1}{2}d\sqrt{4r^2 - d^2}, \tag{11}$$

where d is the distance between the centers of the two circles. Assume $r_1 = r_2 = r_3 = r$, $d_{12} = d_{13} = r$; therefore $d_{23} = r\sqrt{2}$. Observe that Figure 2 shows the variables of this equation where δ is the width of the intersection $A_{N_2 \cap N_3}$ and $\delta = 2r - D$. Now assuming an intersection with $d = r$, the intersection area $A_{N_1 \cap N_2}$ (A_0 and A_2) or $A_{N_1 \cap N_3}$ (A_0 and A_3) [60],

$$A_{i2C} = 2r^2 \arccos\left(\frac{1}{2}\right) - \frac{1}{2}r^2\sqrt{3}. \tag{12}$$

The area of intersection $A_{N_2 \cap N_3}$ (A_0 and A_1), denoted as $A_{i2C'}$ is an intersection where $d = d_{23} = r\sqrt{2}$, [60],

$$A_{i2C'} = 2r^2 \arccos\left(\frac{1}{\sqrt{2}}\right) - r^2. \tag{13}$$

Figure 2 shows the intersection area between three circles denoted as A_0. This area is known as a *circular triangle*, which is a triangle with arc shaped sides (c_1, c_2, c_3). The area can be obtained using the radius and the length of the three arcs as shown in Figure 2. The area is given by the following equation [61]:

$$A_0 = \frac{1}{4}\sqrt{\kappa\lambda\nu\sigma} + \cdots + \sum_{k=1}^{3}\left[r_k^2 \arcsin\left(\frac{c_k}{2r_k}\right) - \frac{c_k}{4}\sqrt{4r_k^2 - c_k^2}\right], \tag{14}$$

where $\kappa = (c_1 + c_2 + c_3)$, $\lambda = (c_2 + c_3 - c_1)$, $\nu = (c_1 + c_3 - c_2)$, $\sigma = (c_1 + c_2 - c_3)$,

$$c_1^2 = 2r^2 - \frac{d^2}{2} + \sqrt{2r^4 - \frac{3r^2d^2}{2} + \frac{d^4}{4}} - \cdots - d\sqrt{r^2 - \frac{d^2}{4}} - d\sqrt{\frac{r^2}{2} - \frac{d^2}{4}}, \tag{15}$$

and $c_1 = c_2$.

The appendix shows how the equation for c_3 is obtained:

$$c_3 = -\frac{r}{\sqrt{2}} + r\sqrt{\frac{3}{2}}. \tag{16}$$

Therefore, the coverage area upper bound (A_{UB}) of ten wireless sensors as shown in Figure 1 is given by

$$A_{\mathrm{UB}} = A_{10C} - 9A_{i2C} - 2A_{i2C'} + 2A_0. \tag{17}$$

With the completion of this analysis, it is possible to obtain the coverage area upper bound, 1286.7 m^2. In Section 5, the values obtained with the DE algorithm are compared to the bounds.

4. The Multiobjective Differential Evolution Algorithm (MODEA) for Wireless Ad Hoc Sensor Distribution

It is well-know that the DE algorithm [20] begins with an initial population of individuals and then iterates to build new populations until a good solution is found. In DE an individual is a vector of the dimension of the problem D. The individuals characterize specific solutions to the problem under study. In this paper, the vector dimension contains the coordinates of the sensor nodes (x, y) followed by their radii values. Considering that there is not information with respect to the optimal solution, the initial population is built in random manner. DE makes repeated cycles of recombination and selection to move the population in the direction of the vicinity of a global optimum. Probability operators (crossing and mutation) are used to each individual in a population to create new individuals (children). The new individuals have some of the features of their ancestors. The ancestors are retained or removed by selection. The term generation is applied to designate the conversion of all individuals into new ones. In other words, to move from one population to another.

The DE algorithm iterates for a limited number of generations, G. It is important to mention that DE has three key control parameters: the mutation constant M, which controls the mutation strength, the recombination constant C_r, and the population size N_p. The parameters M and C_r take values in the interval $(0, 1)$. During the course of the execution process, the user establishes the population size N_p. At each generation, all individuals in the population are evaluated in turn; in DE's literature, the individual being evaluated is named the target vector. Three other individuals are randomly selected from the population and are mixed with each other: this operation is known as mutation and generates a mutant individual vector. The mutant individual vector is then mixed with the current target vector by an operator named recombination: the result of this recombination process is a vector named the trial vector.

Finally, the selection operator is used. If the trial vector improves the objective function then it is accepted and replaces the current target vector in the new population that is being built. Else, it is rejected and the current target vector passes on to the next generation; in this situation the trial vector is not kept.

The following is a description of the operators utilized in the DE algorithm that obtains the most promising region in the search space.

4.1. Objective Functions. In this research work, the DE algorithm seeks to achieve a balance between the maximal effective coverage rate A_{cov} and the minimum sensor power communication $f_2(x) = E_{\mathrm{csd}}$. To translate the network coverage rate into minimal functions, the opposite effect of the coverage rate is used. Thus, the network effective coverage rate is then defined by

$$\min \quad f_1(x) = 1 - A_{\mathrm{cov}}, \tag{18}$$

and the minimum network energy communication is obtained by

$$\min \quad f_2(x) = \mu \sum_{i=1}^{z} r_i^2. \tag{19}$$

The values of $\min f_1(x)$ and $\min f_2(x)$ are part of the fitness function for measuring the result. A linear combination of the objective functions transforms the original multiobjective function into a single-objective function as follows:

$$\min \quad \sum_{k=1}^{K} w_k f_k(x) \quad \text{where } w_k \geq 0; \ \sum_{k=1}^{K} w_k = 1, \tag{20}$$

where w_k are weight coefficients expressing the relative importance of each objective function and k is the number of objective functions that form the total objective function f.

4.2. The Fitness Function. The fitness function is given by

$$f(x) = w_1 f_1(x) + w_2 n_2 f_2(x), \tag{21}$$

here w_1 and w_2 represent weight coefficients and n_2 is the normalization coefficient of $f_2(x)$. Note that $f_1(x)$ is already a normalized function (18) due to the fact that it takes only values between 0 to 1. To normalize the second term, it is necessary to use the following equation:

$$n_2 = \frac{1}{\mu z r_{\max}^2}, \tag{22}$$

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

Consider $w_2 = 1 - w_1$; therefore if $w_1 = 1$, $w_2 = 0$, then the fitness function optimizes the network effective coverage rate. Conversely if $w_1 = 0$, $w_2 = 1$, then the fitness function optimizes the energy consumption of signal detection. The normalized terms of (21) make it possible to get the same order of magnitude. The overall value of the fitness function is in the interval $[0, 1]$. Obviously, the smaller the value of $f(x)$, the better the node location distribution and the lower the power communication consumption as well.

4.3. The Mutation Operator. For each target vector x^i, $i = 1, \ldots, N_p$, the mutant individual m^i is created with the following equation [23]:

$$m^i = x^{r1} + M\left(x^{r2} - x^{r3}\right), \tag{23}$$

where $x^{r1}, x^{r2}, x^{r3} \in \{1, \ldots, N_p\}$: $x^{r1} \neq x^{r2} \neq x^{r3} \neq x^i$. Here, x^{r1}, x^{r2}, and x^{r3} are three random individuals from the population, mutually different and also different from the current target vector x^i, and M is a scaling factor named the mutation constant which must be $M > 0$. The mutation operator is utilized to manage the magnitude of the difference between two individuals. This operator manages the tradeoff among exploitation and exploration on the search process. This operator is the one guiding the convergence of the algorithm. This version is known as DE/rand/1 version. Aside, there is another version like the one shown in [20, 22, 62]

$$m^i = x^{\text{best}} + M\left(x^{r1} - x^{r1}\right), \tag{24}$$

where the best vector among the population is chosen instead of obtaining it randomly. Expression stated in (24) is known as DE/best/1 [20, 22, 62].

4.4. The Recombination Operator. The recombination operator is applied to increment the diversity in the mutation process. This operator is the final step in the creation of the trial vector. To create the trial vector, the mutant individual is combined with the current target vector. In particular, for each component j, where $j = \{1, 2, \ldots, D\}$, of the mutant individual, a random number rand in the interval $[0, 1]$ is selected. Then, the recombination constant C_r and rand are compared in order to check if rand $< C_r$. Equation (25) briefly shows that if the aforementioned condition is false, the jth element of the target vector is selected as the jth element of the trial vector. Otherwise the jth element of the mutant individual is selected as the jth element of the trial vector [23].

$$t^{i,j} = \begin{cases} m^{i,j} & \text{if rand} < C_r \ \forall j \\ x^{i,j} & \text{otherwise.} \end{cases} \tag{25}$$

The reader is warned that a large value for C_r helps to avoid the cancelation of the mutation operation. Otherwise, C_r would become useless in its application on (25).

4.5. Application of the Prim–Dijkstra Algorithm. The nodes' locations are validated considering the distance communication constraint. The communication range distance equal to the sensor coverage radius is used. To satisfy this constraint, the Prim–Dijkstra algorithm is applied [51]. The input to the algorithm is a root node from the set of nodes. This root node may act as the information concentrator and is chosen arbitrarily by the network designer. An external controller starts the algorithm and the algorithm takes action based on each actual position of all nodes. The Prim algorithm [63] determines the minimum spanning tree (MST) whereas the Dijkstra algorithm [64] obtains a star topology. Prim–Dijkstra [51, 52] uses the location of the sensor nodes and obtains a topology that is a mixture of MST and a star topology. As mentioned before, information about the shortest sensor mobility is determined with the Hungarian algorithm [53] finding the minimum distance between the initial and the final node positions.

4.6. The Selection Operator. The last step of the algorithm is the application of the selection operator, where the fitness computed with the trial vector t^i is compared to the fitness computed with the target vector x^i. In a simplified way, (26) summarizes the aforementioned statement, where t^i is chosen if its computed fitness is lower than the fitness computed on x^i and vice versa.

$$\text{pop}_i = \begin{cases} t^i & \text{if } f\left(t^i\right) < f\left(x^i\right) \\ x^i & \text{otherwise.} \end{cases} \tag{26}$$

```
(1) procedure MODEA
(2)     Set the control parameters for MODEA;
(3)     Create initial population;
(4)     Evaluate fitness, area and energy of each member;
(5)     for g = 1 to G do
(6)         for i = 1 to N_p do
(7)             Select from Pop_i (see version: -R (Eq. (23)), -RM (Eq. (27)), -B (Eq. (24)), -BM (Eq. (28));
(8)             Obtain mutation m^i (see previous chosen version);
(9)             Apply recombination to obtain t^i (Eq. (25));
(10)            Validate t^i according to preliminary bounds;
(11)            Update (x, y) and radius (Eq. (30)) if needed;
(12)            while a tree is not obtained do
(13)                Validate t^i;
(14)                Update (x, y) and radius (Eq. (30)) if needed;
(15)                Apply MST Prim-Dijkstra to array of links;
(16)                if some nodes are not connected then
(17)                    Update position and radius (Eq. (30)) of unlinked nodes;
(18)                else nodes are connected
(19)                    Tree has been built;
(20)                end if
(21)            end while
(22)            Apply selection operator (Eq. (26));
(23)        end for
(24)        Pop_best solution is obtained for fitness, area and energy;
(25)    end for
(26)    Apply Hungarian algorithm to Pop_best regarding to Pop_initial;
(27) end procedure
```

Algorithm 1: MODEA applied to WAHSN.

Here, pop_i is the population of the next generation that changes by accepting or rejecting new individuals. The global best individual is retained at the end of each generation, to keep track of the best solution obtained when DEA is executed.

4.7. The Random-M Mutation Parameter. The parameter M is responsible for the exploration and exploitation of the solution spaces. To fix the problem of stagnation for small values of M, the random-M parameter is added to (23). This parameter consists of a random number rand that multiplies the constant M at each evaluation, [22, 65]. Equation (27) shows how the main operator changes with this addition:

$$m^i = x^{r1} + \text{rand}\,()\, M\left(x^{r2} - x^{r3}\right), \tag{27}$$

where rand is a random number in the range of $(0 < \text{rand} < 1)$ that changes during the evolution process. By multiplying M by rand, M modifies its value. In this way, rand helps DEA to have a strong reduction of stagnation. Thus, it is also possible to achieve an improved convergence speed, as presented in Section 5. The same approach may be applied to (24) as shown in

$$m^i = x^{\text{best}} + \text{rand}\,()\, M\left(x^{r1} - x^{r2}\right). \tag{28}$$

4.8. Pseudocode for MODEA in a WAHSN. Employing these considerations, here the pseudocode for the MODEA is presented. The algorithm takes the initial parameters of DE, performs the mutation and recombination, validates the coordinates (x, y) and the radius values, and, if necessary, updates the positions of some nodes. Thereafter, the links are created for each node, proceeding to form a tree, and in the case of nodes not connected to the tree, their positions and radii are updated. After a tree is obtained, the solution may be selected from the best trees generated, based on the best fitness value. The algorithm is considered and denoted as *MODEA* and also is summarized in pseudocode Algorithm 1. The details of the pseudocode are listed as follows:

(1) MODEA starts with the following parameters: M, C_r, N_p, G.

(2) Each individual of population starts as follows: $\text{Pop}_i = x_{i,1}, \ldots, x_{i,z}; y_{i,1}, \ldots, y_{i,z}; r_{i,1}, \ldots, r_{i,z}$ and $\text{Pop}_{\text{initial}} = \text{Pop}_1$. Each $r_{i,j} = r_{\max}$.

(3) Validate t^i based on coordinates (x, y) and r for each member of t^i is performed according to the boundaries, $r_{\max}$ and $r_{\min}$. If $r > r_{\max}$ or $r_{\min} > r$, (30) is applied to regenerate r. If (x, y) are outside the boundaries, (x, y) are regenerated.

(4) The Hungarian algorithm receives Pop_{best} (the best result obtained from final population at the end of MODEA) and $\text{Pop}_{\text{initial}}$ to determine the final position (e) of the sensors regarding the initial positions (i).

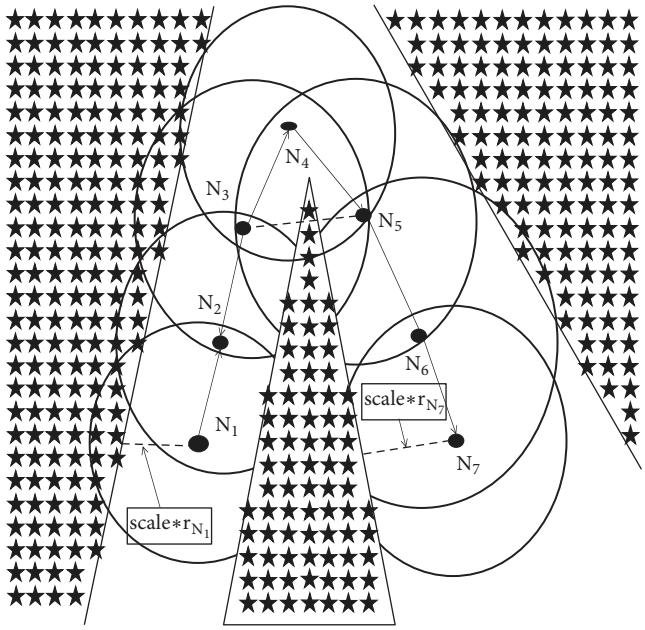

Figure 3: Forbidden zone.

(5) The MST is formed by using central node (cnode = 5), $\alpha = 0$, and the distance to generate a link between the nodes ($d(N_i, N_j)$, $i \neq j$) as

$$d\left(N_i, N_j\right) = \sqrt{\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2}. \tag{29}$$

If $d(N_i, N_j) > \min(r_i, r_j)$ or if there is a forbidden zone between the nodes, the link does not exist.

(6) For each generation $g \in \{1, \ldots, G\}$, all the individuals of population are updated according to selection operator.

For simplification purposes, each MODEA version will be treated as follows for the next sections of this manuscript:

MODEA rand (23) ↣ MODEA-R,

MODEA rand (with random-M) (27) ↣ MODEA-RM,

MODEA best (24) ↣ MODEA-B,

MODEA best (with random-M) (28) ↣ MODEA-BM.

Figure 3 shows the parameter *scale* which scales the radius of the current node to separate it from the boundaries of the forbidden zone (see nodes N_1 and N_7 with their respective radii r_{N_1} and r_{N_7}) and the limits of the area of interest. Even when N_3 and N_5 are close to each other, if there is a forbidden area between them, these nodes must use N_4 as the communication node. Throughout these restrictions, the nodes are separated while maintaining the MST. To separate the nodes from the main contour area ($L \times H$), $r_{\min}$ is used instead as a reference.

The MODEA (all rand and best versions and their corresponding random-M versions) adjusts the radius value according to its rules and the given restrictions. The algorithm uses (30) to generate the radius adjustment r_{adj} when validating t^i. In this algorithm, the radius of each population member starts at its maximum value; as a result, the average radius at the end of the generations is a value above the minimum radius.

$$r_{\text{adj}} = r_{\min} + \text{rand}\,()\left(r_{\max} - r_{\min}\right). \tag{30}$$

5. Numerical Results and Discussion

The methodology followed in this manuscript was based on the one presented in [26]. Nevertheless, the treatment of the scenarios and constraints here presented were stated by following the convex configurations [35]. Compared to [26] and other similar works [34], the main differences are that the center of a circle should be inside in at least one of its neighbor circles, where all the circle centers must display a connected graph. The detailed rules are described in Section 4.8. This approach was adopted before testing more elaborated bioinspired techniques for future works, in order to have a baseline and then to determine the feasibility of MODEA with stricter constraints. A previous successful approach with the use of nonconvex constraints and obstacles can be seen in [32]. In future works, such elaborated bioinspired techniques will be tested (more details are available in the Conclusion).

The algorithm was implemented in MATLAB to optimize the distribution of sensors. The effective sensor radius was assigned between $r_{\min} = 6$ and $r_{\max} = 8$ meters, except the *multiple-bounded* case, where $r_{\min} = 4\,\text{m}$, $r_{\max} = 6\,\text{m}$. Different shaped-variant target zones were also tested. The same starting random node distribution was implemented on each scenario, with the purpose of having fair performance comparisons and the computation of the corresponding averages, that is, a unique starting node distribution for squared areas, a unique starting node distribution for polygons, etc., where it is clear that all the starting node distributions are not equal among the analyzed scenarios.

The control parameters were set as $M = 0.8$, $NP = 35$, and $C_r = 0.9$, where M and C_r were assigned taking the best reported performance for both parameters in the open literature [62]. Figure 4 shows the results of the fitness function against the number of generations of *MODEA-R, MODEA-RM, MODEA-B,* and *MODEA-BM*. These tests are the average of 50 independent tests on a squared area of $40\,\text{m} \times 40\,\text{m}$, $G = 1000$, with the maximum possible radius; that is, $r_{\min} = r_{\max} = 8\,\text{m}$. Note that *MODEA-BM* outperforms *MODEA-R*, *MODEA-RM*, and *MODEA-B* obtaining the lowest fitness values. In this specific case, a value of $w_1 = 1.0$ is used to obtain a result near to the area with 10 nodes computed in Section 3.

Comparing the averages obtained among the four versions of MODEA shown in Figure 4, the authors of this manuscript decided to use MODEA-BM because the average obtained on all the 50 tested cases is the best. The best case for MODEA-BM among these 50 tests was obtained in the test 23/50, where the notation is denoted as *numtest*/*numfulltests*. This choice was taken even when

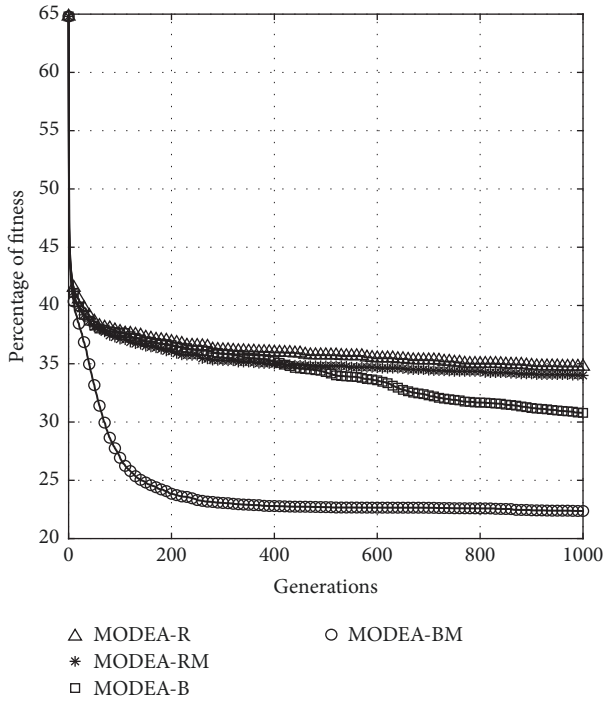

Figure 4: Fitness versus generations.

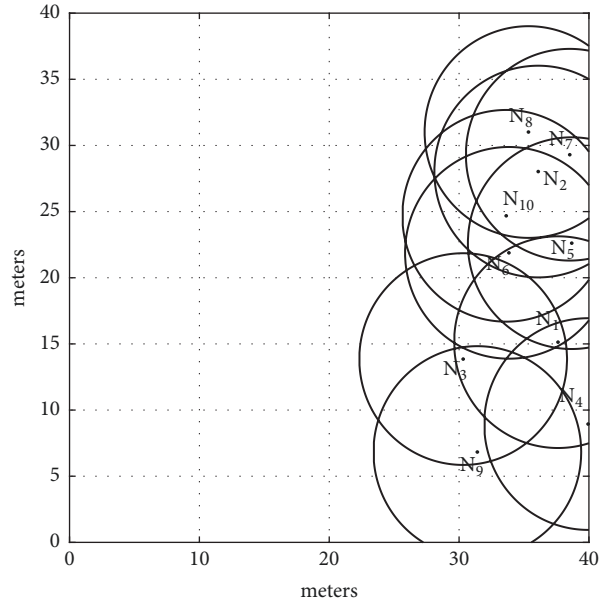

Figure 5: Initial random distribution of 10 sensor nodes.

there is a best result obtained with MODEA-B. Even so, the previous mentioned results are consistent with those presented in [34], where MODEA-B became the best possible option by considering the classical MODEA variants presented in [20, 22].

Figure 5 shows the initial position of 10 sensors randomly distributed. The sensor nodes are identified as $N_1, N_2, \ldots, N_{10}$, and the circles are the sensor ranges of the sensor nodes. As explained before, this starting node

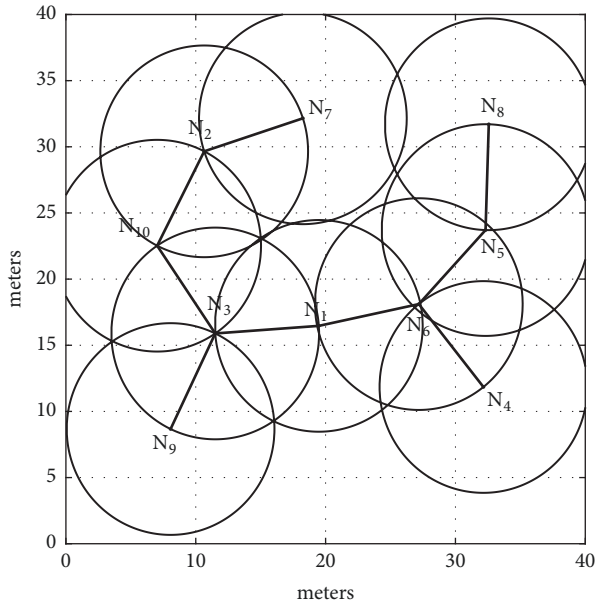

Figure 6: Optimized distribution of 10 sensors.

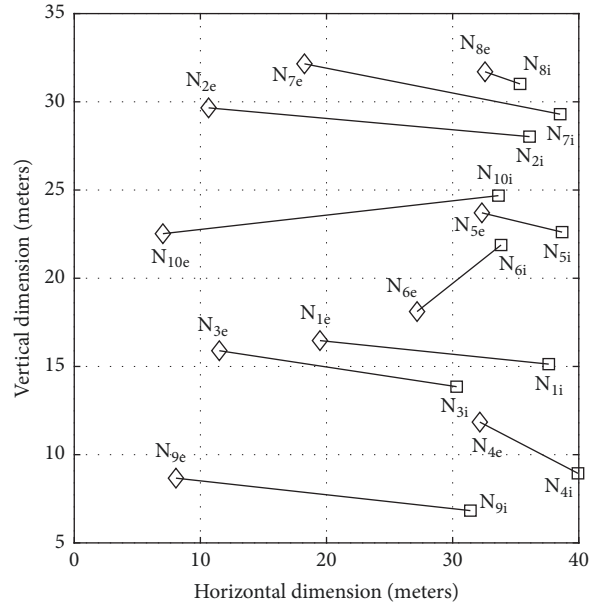

Figure 7: Shortest distance from initial (i) to final (e) node positions generated by the Hungarian algorithm.

distribution was implemented for all the MODEA versions for all the 50 tests applied on this squared scenario.

Figure 6 shows the optimized sensor distribution using the initial sensor positions of Figure 5. Also, the wireless communication links between the nodes are shown. The presence of a link between two nodes in the network depends on the relative distance between them (lower than the minimum radius between two nodes).

Figure 7 shows the initial and final position of the sensors nodes, identified with the subindexes (i) and (e), respectively. Note that the final node positions are at the shortest distance

Table 2: Bounds: theoretical model (TM) and MODEA-BM results.

10 Nodes Bounds	Area TM (m^2)	Energy TM (mW)	Area (m^2) MODEA Best	Energy (mW) MODEA Best
Lower	113.09	1.8	-	**1.8**
Upper	1286.7	3.2	**1274.6**	3.2

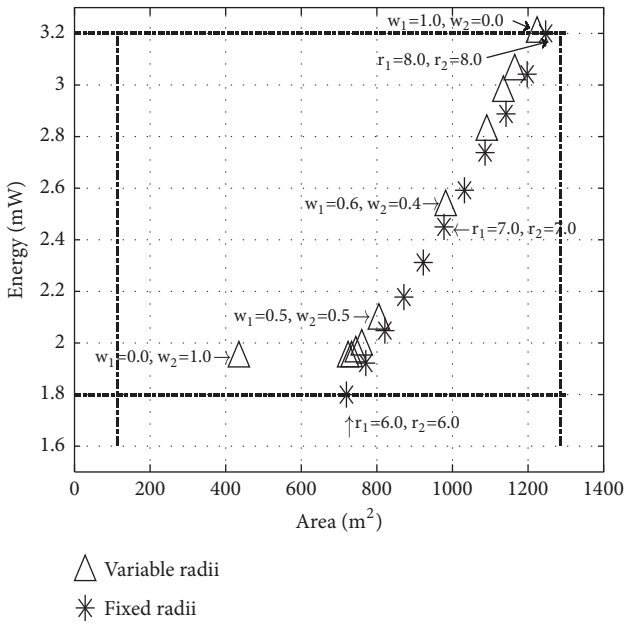

Figure 8: Bounds comparison of 10 nodes, varying w_1 and w_2.

from the initial node positions. For mobile sensors, this information about mobility is useful to globally minimize sensor path distance. A minimum distance matching algorithm after MODEA-BM is used to obtain the minimum global traveled distance between the initial and the final node positions through the Hungarian algorithm, [53].

Table 2 shows a summary of the results obtained using the lower energy and the upper area bound equations presented in Section 3. Also, the results obtained with the MODEA-BM are shown for comparison purposes (case 23/50). The results obtained with the use of MODEA-BM are within 99.05% of the theoretical bounds (1274.6 m^2/1286.7 m^2). With the use of the random-M parameter, the MODEA-BM algorithm has a high convergence speed and stagnation is avoided, as shown in (27). The upper bounded area is 80.41% of the total area to be covered (1286.7 m^2/1600 m^2); the computed area with MODEA-BM is 1274.6 m^2 which represents 79.66% of the total area to be covered (1274.6 m^2/1600 m^2).

Now, in Figure 8 an analysis of the weights w_1 and w_2 of (30) and how these values affect the optimization of energy and coverage area is presented for MODEA-BM. It was done by following the methodology presented in [65, 66]. Here, w_1 varies from 0 to 1 in increments of 0.1; meanwhile the value of w_2 varies from 1 to 0 so that $w_2 = 1 - w_1$. The fixed radii curve ($*$) shows the bounds comparison when the values of

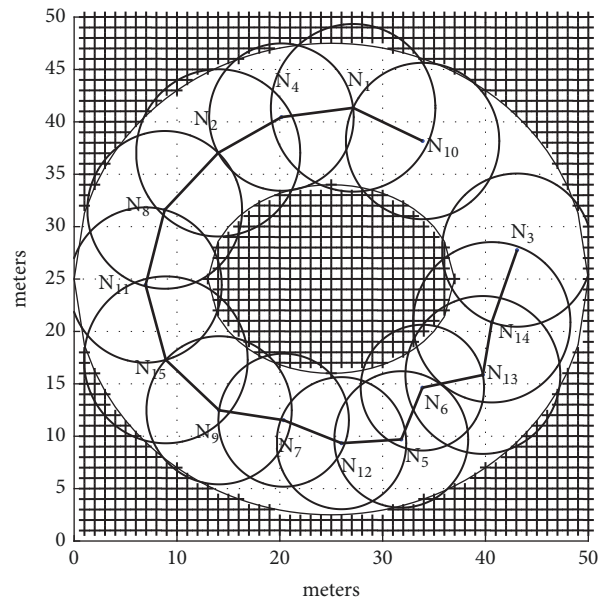

Figure 9: Area formed with two ellipses.

radii between r_{max} and r_{min} are equal. The variable radii curve ($\triangle$) shows the behavior of bounds comparison when the value of radii varies between $r_{min} = 6$ m and $r_{max} = 8$ m. Here, the observed trend is to maximize the coverage area; nevertheless the restrictions help to reduce the radii of the nodes. It was observed that values of $w_1 = 0.6$ and $w_2 = 0.4$ are able to produce results in which the covered area is maximized and, meanwhile, the energy is minimized. This is because those values are near to the case where $r_{min} = r_{max} = 7$ m for the fixed scenario. The results presented here are the average of 50 independent tests of $G = 1000$ for each point at each scenario. The tests to obtain the point $w_1 = 1.0$, $w_2 = 0.0$ were performed independently of those used to compute the output for Figure 6.

5.1. Different Scenarios of Testing. Now the results of the algorithm MODEA-BM parameter applied to the sensor distribution problem in different shaped-bounded areas are presented. The number of nodes is obtained by incrementing the number of sensors depending on the coverage area. Using $w_1 = 0.6$, $w_2 = 0.4$, and $G = 1000$, MODEA-BM was able to find good sensor distributions. For *multiple-bounded* case, $G = 2500$. Restrictions were tuned and set to obtain a suboptimal result for each of the following scenarios. Figures 11–16 were obtained by MATLAB with the command *inpolygon* as described in the corresponding subsections. The best minimum fitness was considered for all the presented cases.

5.1.1. Scenario 1. Ellipse. To form the ellipse area shown in Figure 9, the parameters are $h_A = h_B = 25$, $k_A = k_B = 25$, $r_A = 15$, $r_B = 25$, $a_A = 0.8$, $a_B = 1.0$, $b_A = 0.6$, $b_B = 0.9$, where A is the inner ellipse and B is the external ellipse, displayed into an area of 50 m × 50 m. The corresponding equations are listed as follows:

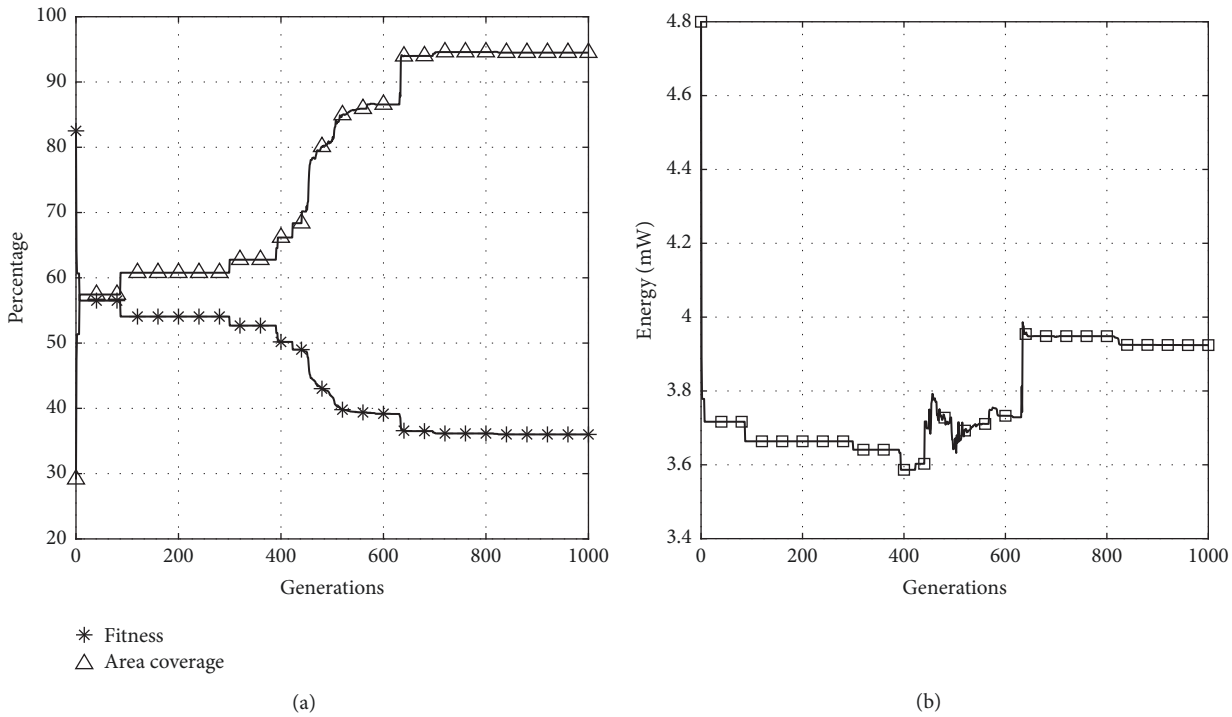

Figure 10: Results for (a) fitness, area, and (b) energy consumption on the scenario with two ellipses.

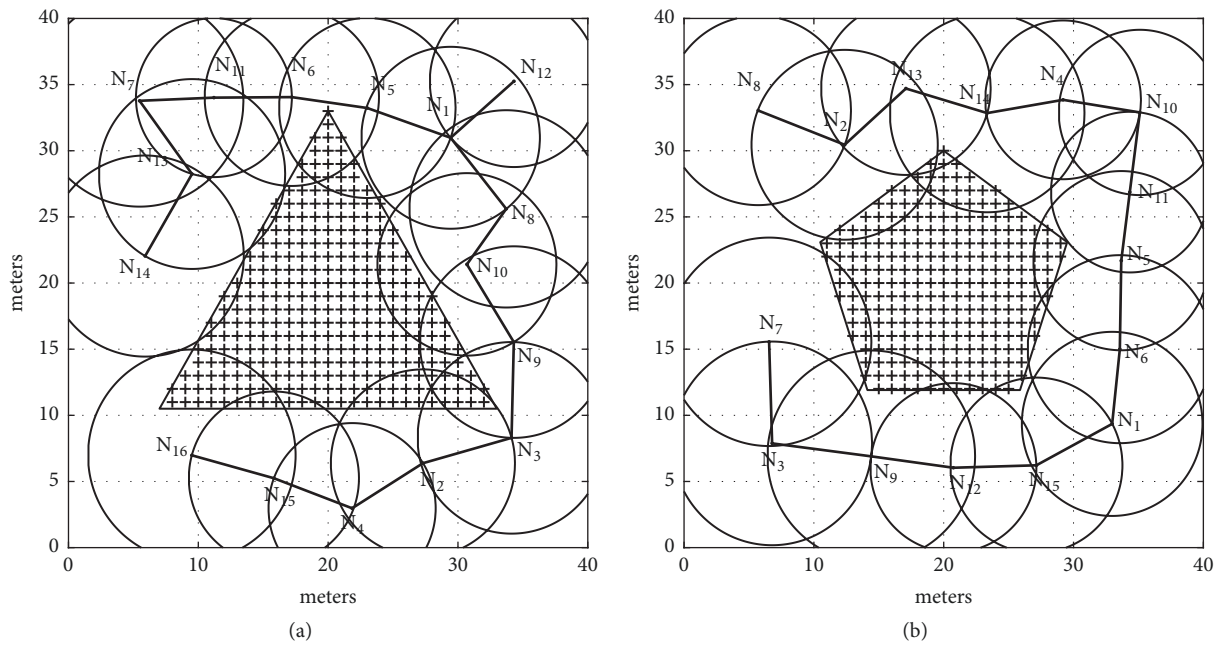

Figure 11: (a) Triangle and (b) pentagon scenarios, best output.

$$\frac{(x-h_A)^2}{a_A^2}+\frac{(y-k_A)^2}{b_A^2}=r_A^2,$$
$$\frac{(x-h_B)^2}{a_B^2}+\frac{(y-k_B)^2}{b_B^2}=r_B^2. \quad (31)$$

It is observed in Figure 9 that the circles follow the contour of the given ellipses, maximizing the coverage area. The best minimum fitness was obtained for the test 9/50.

Figure 10 shows the total energy consumption of the best curved-bounded scenario. The covered area is more than 90% while the energy is below 4 units. The fitness curve indicates

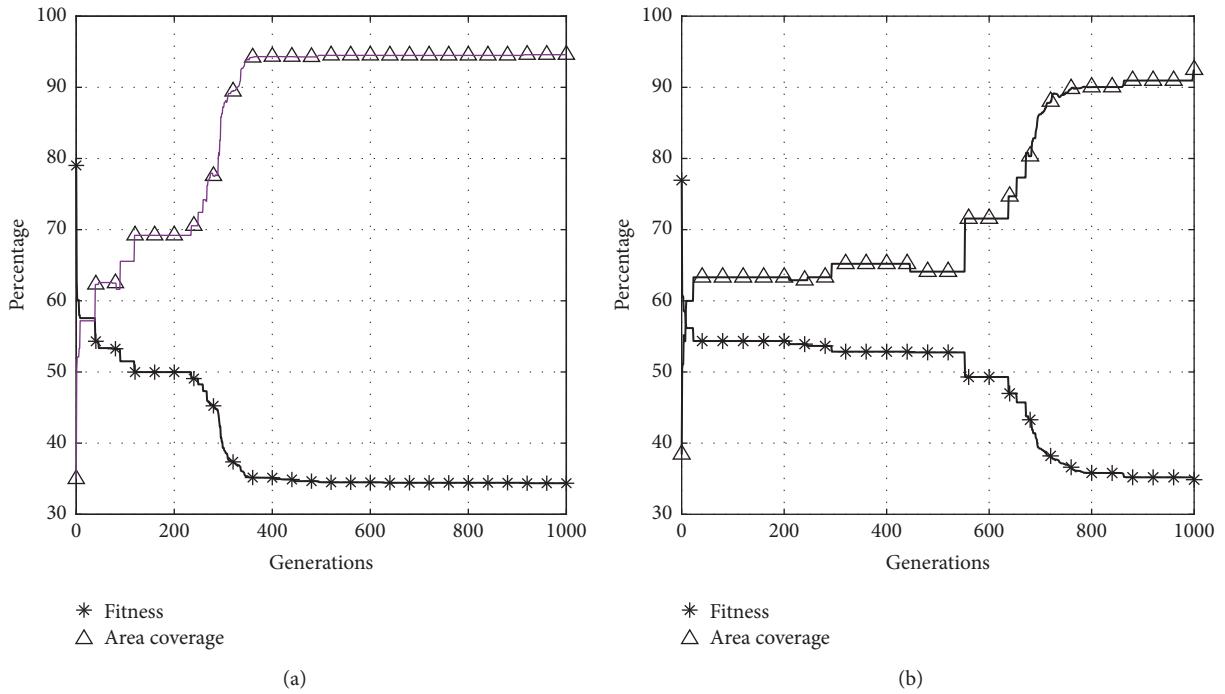

Figure 12: (a) Triangle and (b) pentagon scenarios, fitness and area curve.

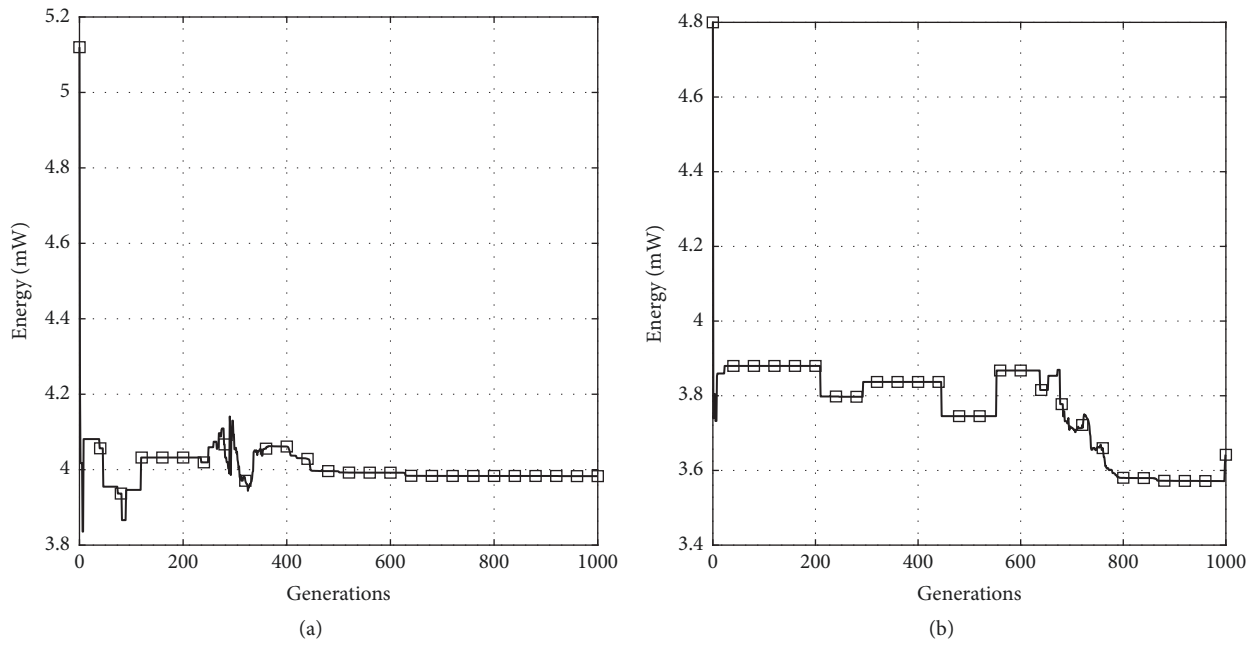

Figure 13: (a) Triangle and (b) pentagon scenarios, energy curve.

how both parameters are evolving to be optimized. After 700 generations, the fitness curve becomes stable, maintaining the area/energy limit values also stable.

The line resolution between two nodes was left as 0.01. The complete circle contour was considered to determine the separation of the boundaries. That is, the angles to compute the radius circle were set as $0 : 0.01 : 360$.

5.1.2. Regular Polygons, External Area. MODEA-BM is applied to the pentagon shown in Figure 11(b) and also to the

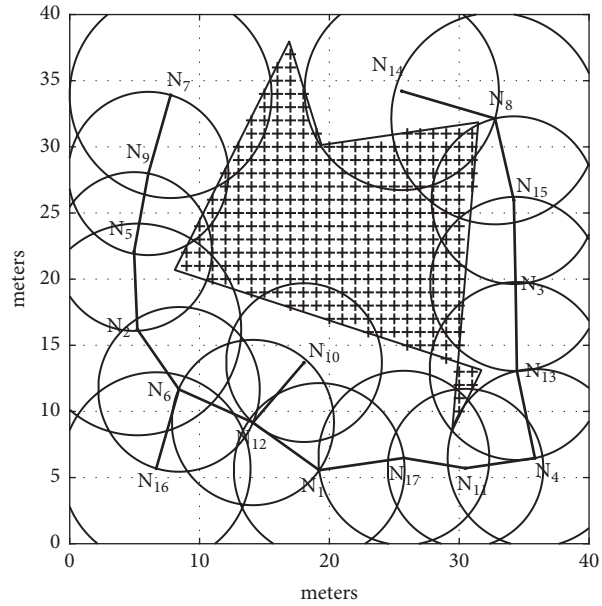

Figure 14: Irregular area outside an amorphous zone.

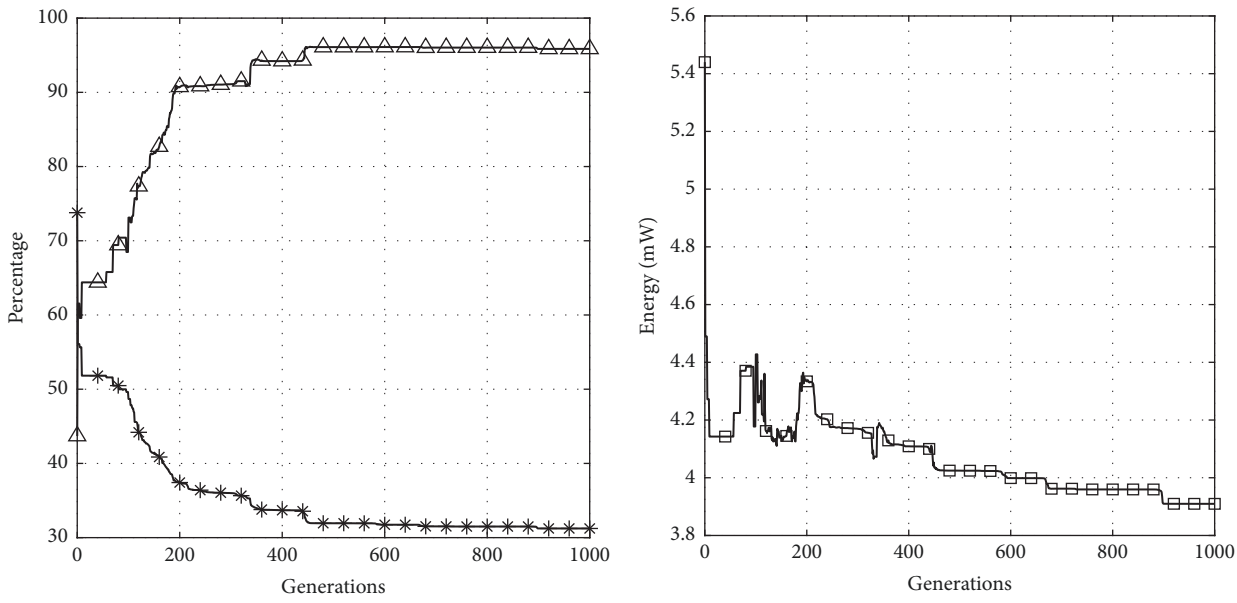

Figure 15: (a) Area, fitness, and (b) energy curves for amorphous figure.

triangle shown in Figure 11(a). The figures here exposed are obtained using the expression $LT = \text{linspace}(0, 2\pi, \text{points})$ and the equations given below:

$$x = A\left[\cos\left(LT + \frac{\pi}{2}\right)\right] + U,$$
$$y = A\left[\sin\left(LT + \frac{\pi}{2}\right)\right] + V, \tag{32}$$

where points = 4, $A = 15$, $U = 20$, $V = 18$ in Figure 11(a) and points = 6, $A = 10$, $U = V = 20$ in Figure 11(b). Vectors x and y define the vertices of the polygon. Both polygons are displayed into an area of 40 m × 40 m.

Figure 12 shows the area and fitness obtained for both scenarios. While for triangle the fitness converges before $G = 400$, for pentagon the fitness delays to converge before $G = 900$. For both cases, the coverage area is greater than 90%.

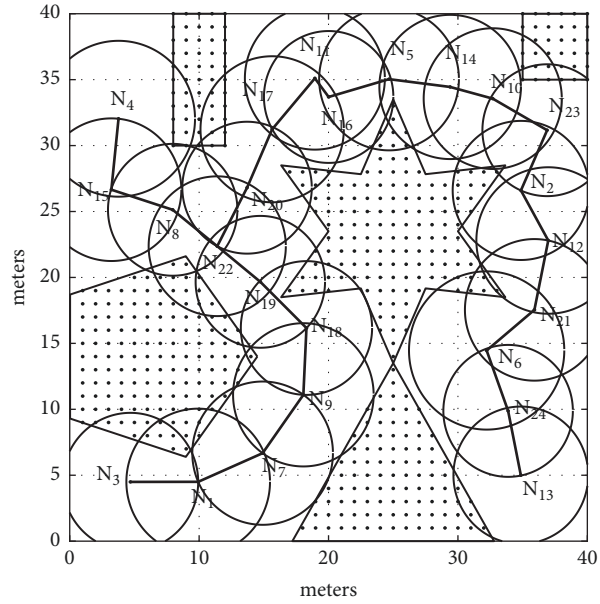

Figure 16: Multiple figures scenario.

Figure 13 shows the energy obtained for both scenarios. While for triangle the energy becomes stable after $G = 400$ with 4 units, for the pentagon the energy holds the same after $G = 800$ with 3.6 units.

The best minimum fitness was obtained as 41/50 for triangle and 45/50 for pentagon. To speed up the algorithm, each circle was sampled in angles as $0:45:360$ to compute the radius. The line resolution between two nodes was left as 0.001 in order to detect forbidden crosses on the shaded area.

5.1.3. Irregular External Area. In this scenario, MODEA-BM is applied to the irregular area shown in Figure 14, which is obtained with

$$x = \frac{(2H)\,x}{2}$$
$$y = \frac{(2H)\,y}{2} + 1, \tag{33}$$

where the user can choose the values of vectors x and y to generate the irregular polygon of Figure 14. This scenario is presented to show how the algorithm is able to work with true irregular areas. Figure 14 is formed by two linked irregular polygons. The figure is displayed into an area of 40 m × 40 m.

As shown in Figure 14, the algorithm prevents that links of the MST from crossing the small polygon, also keeping a distance between the nodes and the entire polygon. With 17 nodes, it is possible to cover more than 90% of the amorphous figure (Figure 15(a)) while the energy is below 4 units (Figure 15(b)). The best minimum fitness was obtained as 10/50.

To speed up the algorithm, each circle was sampled in angles as $0:90:360$ to compute the radius. The line resolution between two nodes was left as 0.001 to check if a link crosses forbidden areas, specially the small triangle of the scenario.

5.1.4. Multiple-Bounded Area. A multiple-bounded area is shown in Figure 16 with a different radii values, $r_{\min} = 4$ m, $r_{\max} = 6$ m, for speeding up the computation of results. The number of tests here are equal to 20. All the polygons are displayed into an area of 40 m × 40 m. This case shows how the algorithm can be adapted to cover an area with multiple bounds (forbidden areas). These bounds represent a triangle, pentagon, star, and two rectangles. For instance, to get the star figure, the following parameters: $v = 12$, $t = (-1/4 : 1/v : 3/4) * 2 * \pi$, $r_1 = 10$, $r_2 = 5$, $p = (0 : v)$, $r = (r_1 + r_2)/2 + (r_1 - r_2)/2 * (-1)^p$ and (34) are used

$$x = r\cos(t) + 25,$$
$$y = r\sin(t) + 23.5. \tag{34}$$

Here, the algorithm is able to arrange the obtained MST as a contour to the given bounds, preventing MST links from crossing forbidden areas. This effect is seen in the case of the star, where the links of the nodes do not cross the peaks of the star. A similar effect is seen for the rectangles, where the nodes forming the circle go nearly in parallel with 24 nodes. This property is maintained even in the presence of the thinnest region that connects the star and the triangle, where there is no link that crosses it. Thus, it is possible to cover more than 90% of the area of Figure 16 (see Figure 17(a)) with less than 3.5 units of energy (see Figure 17(b)).

The best minimum fitness was obtained as 2/20. To speed up the algorithm, each circle was sampled in angles as $0:30:360$ to compute the radius and the line between two nodes was sampled with 0.1 of resolution. When constructing the final MST, that line is sampled with 0.001 of resolution to detect if it crosses forbidden zones. The *scale* parameter guarantees that all the cases do not have issues with the line resolution when finishing MODEA-BM.

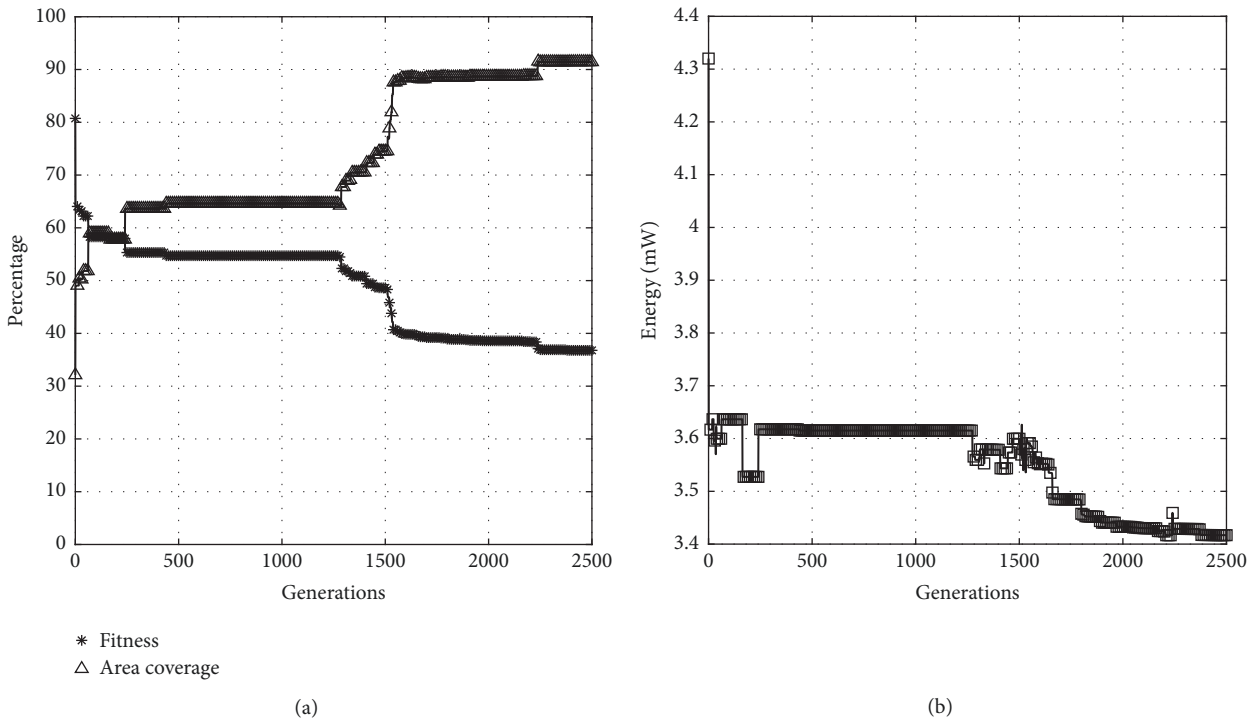

Figure 17: (a) Area, fitness, and (b) energy curves for multiple figures.

5.1.5. Summary of the Results. Table 3 shows a summary of the average area results obtained from the different figures comparing them with the maximum area to be covered. The results obtained using MODEA-BM parameter are very close to the area to be covered. On each case, 50 tests were made to obtain their mean and standard deviation, except for the multiple case, where only 20 tests were done. The energy for all the cases remains below 4 units, while the coverage area is greater than 90% for the best solutions obtained. The computation of the average and standard deviation on the area parameter shows that the *irregular external area* case remains greater than 90% with the lowest standard deviation, which is consistent with the fitness value (34.98%). In contrast, the *multiple* case shows the worst average (77.35%) and the largest standard deviation (135.10) and also presents the largest fitness value (46.17%) with the largest standard deviation (8.00). Even so, for *multiple* case, MODEA-BM is able to obtain results where the covered area is greater than 90%.

Table 4 shows the comparison of the DEA here proposed to well-known multiobjective optimization algorithms such as NSGA-II and MOEA/D. The authors modified the generic versions of both algorithms, where the corresponding source codes are available in [67]. The default configuration parameters were left unmodified except $Np = 35$. Here, only the base case scenario was tested. For the sake of consistency and fairness, the same 50 tests were done for the following presented cases. For MODEA, the radii-variable coordinate obtained with $w_1 = 0.6$ and $w_2 = 0.4$ was included for the comparison (see Figure 8). For the other multiobjective algorithms, four different scenarios were tested. Here, $g_1(x)$ is the nonnormalized version of $f_1(x)$ and $g_2(x)$ is the nonnormalized energy version of $f_2(x)$. Additionally, $A_{\max}$ is the area to be covered and $E_{\max}$ is given by (22). These variants are described as follows:

(i) Version-I (V-I): the coordinates of the algorithm are tested without normalizing them. That means

$$g(x) = \begin{bmatrix} g_1(x) & g_2(x) \end{bmatrix}. \tag{35}$$

(ii) Version-II (V-II): the coordinates of the algorithm are tested, but in a normalized way as

$$g(x) = \begin{bmatrix} \dfrac{g_1(x)}{A_{\max}} & \dfrac{g_2(x)}{E_{\max}} \end{bmatrix}. \tag{36}$$

(iii) Version-III (V-III): each normalized coordinate is weighted by 0.5 as

$$g(x) = \begin{bmatrix} \dfrac{0.5g_1(x)}{A_{\max}} & \dfrac{0.5g_2(x)}{E_{\max}} \end{bmatrix}. \tag{37}$$

(iv) Version-IV (V-IV): the normalized area ($g_1(x)$) is weighted by 0.6. Meanwhile, the normalized energy ($g_2(x)$) is weighted by 0.4 as

$$g(x) = \begin{bmatrix} \dfrac{0.6g_1(x)}{A_{\max}} & \dfrac{0.4g_2(x)}{E_{\max}} \end{bmatrix}. \tag{38}$$

Table 3: Area to cover and area covered by MODEA-BM, plus energy consumption. Averages, standard deviation, and best cases, where BC means best case, $\widehat{\mu}$ is the average, and σ is the standard deviation.

Figure Number	Data	Fitness Percent %	Fitness σ	Area (m^2) MODEA-BM	Area (m^2) to cover	Area Percentage %	Area σ	Energy (mW)	Energy σ	Scale parameter	Number of Nodes
Figure 9	Stats $(\widehat{\mu}, \sigma)$	42.25	5.04	1151.63	1427.72	80.66	124.16	3.67	0.19	0.500	15
	BC 9/50	36.00	-	1349.12		94.49	-	3.92	-		
Figure 11(a)	Stats $(\widehat{\mu}, \sigma)$	37.30	1.97	1120.02	1306.40	85.73	59.75	3.67	0.13	0.500	16
	BC 41/50	34.36	-	1235.63		94.58	-	3.98	-		
Figure 11(b)	Stats $(\widehat{\mu}, \sigma)$	39.76	4.66	1136.88	1362.71	83.42	103.41	3.57	0.18	0.200	15
	BC 45/50	34.86	-	1260.21		92.47	-	3.64	-		
Figure 14	Stats $(\widehat{\mu}, \sigma)$	34.98	1.80	1159.72	1288.5	90.00	53.79	3.94	0.15	0.200	17
	BC 10/50	31.23	-	1235.06		95.85	-	3.90	-		
Figure 16	Stats $(\widehat{\mu}, \sigma)$	46.17	8.00	871.15	1126.21	77.35	135.10	3.51	0.13	0.400	24
	BC 2/20	36.78	-	1029.67		91.42	-	3.41	-		

Table 4: Comparison of DEA algorithm with some known multiobjective algorithms. Average on 50 tests.

Algorithm	Area (m^2) mean	Energy (mW) mean
MODEA (ours) ($w_1 = 0.6, w_2 = 0.4$)	981.9858	2.5266
NSGA-II V-I	1028.5674	2.8120
NSGA-II V-II	667.3309	2.1155
NSGA-II V-III	721.3496	1.9494
NSGA-II V-IV	875.0040	2.3059
MOEA/D V-I	868.0113	2.9287
MOEA/D V-II	708.6827	1.9229
MOEA/D V-III	648.9909	2.0030
MOEA/D V-IV	809.0555	2.3831

According to the results shown in Table 4, MODEA presents the best average performance among all the cases, with exception of NSGA-II version V-I. NSGA-II V-I obtains a coverage area 4.53% larger than MODEA; however the energy is 10.15% larger. It is shown that MODEA is more efficient in terms of the energy consumption; meanwhile NSGA-II increases the coverage area but raises the energy consumption. The rest of the multiobjective algorithms versions do not reach a similar area to that obtained by MODEA.

6. Conclusion

Firstly, a base case was thoroughly studied in a theoretical way to determine the minimum and maximum limits of that base configuration. This was done to show how MODEA is able to find a suboptimal solution near to its maximum theoretical configuration with great accuracy (99.05%). At the same time on the same base case, several feasible optimization strategies were tested with the use of MODEA, denoted as MODEA-R, MODEA-RM, MODEA-B, and MODEA-BM. It can be seen that the addition of the random-M parameter in Storn's algorithm helps to converge toward the optimal solution quickly, particularly for MODEA-BM. For that reason, MODEA-BM was chosen to evaluate the rest of the cases here presented, i.e., different area shapes, either one shape or multiple shapes. Results obtained with 1000 iterations and $w_1 = 0.6, w_2 = 0.4$ showed that more than 90% of the targeted area is covered for the best outputs, except for *multiple* case. For a highly restricted scenarios like *multiple* case, the best outputs showed that the suboptimal solution is reached in 2500 iterations, even when in general the low average (77.35%) shows that some of the cases do not reach the desired output. The MST was adapted to avoid crossing forbidden areas, creating feasible communication trees while optimizing the objective function. Nevertheless, the readers are warned that an adequate restriction tuning leads to a feasible solution, especially in cases where the area shapes are irregular, multiple, or a combination of both. Thus, MODEA-BM prevents the existence of nodes in forbidden areas, near these areas' borders, or near each other with a certain tolerance, thus maximizing the covered area and reducing energy consumption. MODEA-BM is complemented with the use of the Prim–Dijkstra algorithm to find on-the-fly the best possible MST for each population member. MODEA-BM is also complemented with the Hungarian algorithm at the end in order to find which movements should be performed to arrange the initial to the end node positions.

As further work, the following lines of research can be done: (1) The application of the MODEA algorithm should be investigated in real indoor coverage areas considering wireless propagation effects such as shadowing, interference, obstacles, and multipath. Those restrictions shown by [17, 18] will complement the results presented in this work. (2) With some further adaptations, MODEA could be applied for guiding autonomous sensorial robots in an unexplored geographical area or (3) in search and rescue operations, where it is common to have different sensor nodes with different transmission rates, ranges, and sampling rates. (4) It would be interesting to compare MODEA against other well-known bioinspired algorithms such as Ant Colony Optimization or the latest MODEA versions found in the open literature (jDE [68], JADE [69], SADE [70], and the like [62]) and obtain similar results like those shown in [34]. (5) A simulation to evaluate MODEA should be done using random variables to generate irregular coverage areas and simultaneously determine its impact in forming links in an ad hoc network, reflected in the construction of routing tables.

Appendix

Figure 18 given in [61] shows the intersection of three circles called a circular triangle. This figure is used to obtain (14) and (16), which are used to obtain (16), which is the area of the circular triangle. The vertices of the circular triangle are labeled by (x_{ij}, y_{ij}).

The separation between the centers of circle 1 and circle 2, d_{12}, must satisfy

$$r_1 - r_2 \leq d_{12} \leq r_1 + r_2. \tag{A.1}$$

If not satisfied, then there is no circular triangle. However, if satisfied, the circles intersect in two points. Each intersection point satisfies the equation of these two circles.

$$\begin{aligned} x_{12}^2 + y_{12}^2 &= r_1^2, \\ (x_{12} - d_{12})^2 + y_{12}^2 &= r_2^2, \end{aligned} \tag{A.2}$$

where (x_{12}, y_{12}) are the coordinate variables of the intersection points that correspond to

$$\begin{aligned} x_{12} &= \frac{r_1^2 - r_2^2 + d_{12}^2}{2d_{12}}, \\ y_{12} &= \frac{1}{2d_{12}} \sqrt{2d_{12}^2 \left(r_1^2 + r_2^2\right) - \left(r_1^2 - r_2^2\right)^2 - d_{12}^4}. \end{aligned} \tag{A.3}$$

Now considering the third circle and using the coordinate system of (x', y'), the equations of the intersection point

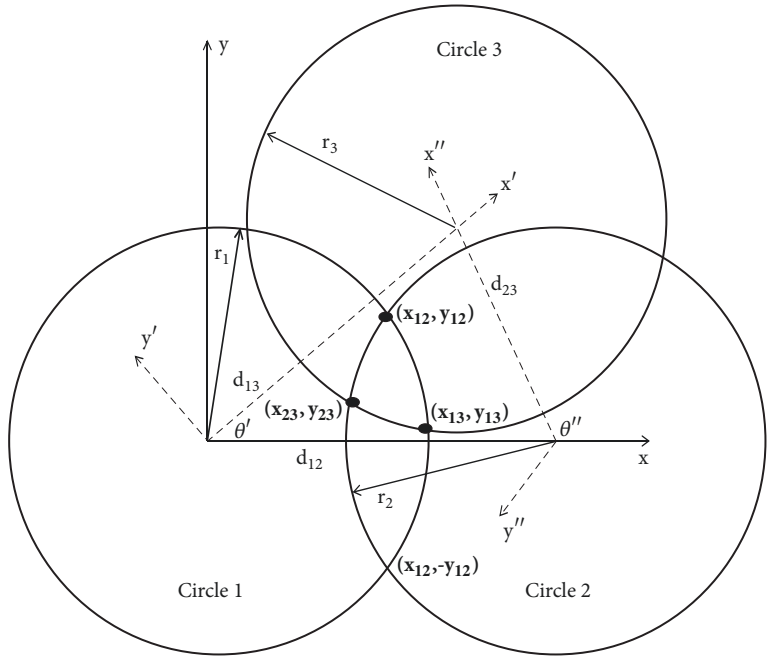

Figure 18: Vertices of the circular triangle.

between circle 1 and circle 3 are derived. Note that y'_{13} has a negative value.

$$x'_{13} = \frac{r_1^2 - r_3^2 + d_{13}^2}{2d_{13}},$$
$$y'_{13} = \frac{-1}{2d_{13}}\sqrt{2d_{13}^2\left(r_1^2 + r_3^2\right) - \left(r_1^2 - r_3^2\right)^2 - d_{13}^4}. \tag{A.4}$$

Assuming that $r_1 = r_2 = r_3 = r = d$, $d_{12} = r\sqrt{2}$, $d_{13} = r$, and $d_{23} = r$, it is found that

$$x'_{13} = \frac{r}{2},$$
$$y'_{13} = \frac{-r\sqrt{3}}{2}. \tag{A.5}$$

Now, considering that

$$x_{13} = x'_{13}\cos\theta' - y'_{13}\sin\theta',$$
$$y_{13} = x'_{13}\sin\theta' + y'_{13}\cos\theta' \tag{A.6}$$

and that

$$\cos\theta' = \frac{d_{12}^2 + d_{13}^2 - d_{23}^2}{2d_{12}d_{13}},$$
$$\sin\theta' = \sqrt{1 - \cos^2\theta'} \tag{A.7}$$

and substituting values, the following expressions are obtained

$$\cos\theta' = \frac{\sqrt{2}}{2},$$
$$\sin\theta' = \frac{\sqrt{2}}{2},$$
$$x_{13} = r\left(\frac{\sqrt{2}}{4} + \frac{\sqrt{6}}{4}\right),$$
$$y_{13} = r\left(\frac{\sqrt{2}}{4} - \frac{\sqrt{6}}{4}\right). \tag{A.8}$$

Using the following equations

$$x''_{23} = \frac{r_2^2 - r_3^2 + d_{23}^2}{2d_{23}},$$
$$y''_{23} = \frac{1}{2d_{23}}\sqrt{2d_{23}^2\left(r_2^2 + r_3^2\right) - \left(r_2^2 - r_3^2\right)^2 - d_{23}^4},$$
$$x_{23} = x''_{23}\cos\theta'' - y''_{23}\sin\theta'' + d_{12},$$
$$y_{23} = x''_{23}\sin\theta'' + y''_{23}\cos\theta'',$$
$$\cos\theta'' = \frac{-d_{12}^2 + d_{23}^2 - d_{13}^2}{2d_{12}d_{23}},$$
$$\sin\theta'' = \sqrt{1 - \cos^2\theta''} \tag{A.9}$$

and substituting values, the following results are obtained

$$\begin{aligned} \cos\theta' &= \frac{\sqrt{2}}{2}, \\ \sin\theta' &= \frac{\sqrt{2}}{2}, \\ x_{13} &= r\left(\frac{\sqrt{2}}{4} + \frac{\sqrt{6}}{4}\right), \\ y_{13} &= r\left(\frac{\sqrt{2}}{4} - \frac{\sqrt{6}}{4}\right). \end{aligned} \tag{A.10}$$

The use of (16) is to verify that circle 3 forms a circular triangle under the conditions

$$\begin{aligned} \left(x_{12} - d_{13}\cos\theta'\right)^2 + \left(y_{12} - d_{13}\sin\theta'\right)^2 &< r_3^2, \\ \left(x_{12} - d_{13}\cos\theta'\right)^2 + \left(y_{12} + d_{13}\sin\theta'\right)^2 &> r_3^2. \end{aligned} \tag{A.11}$$

If these conditions are satisfied, then there is a circular triangle. Now, substituting values,

$$\begin{aligned} \cos\theta'' &= \frac{-\sqrt{2}}{2}, \\ \sin\theta'' &= \frac{\sqrt{2}}{2}, \\ x_{23} &= r\left(\sqrt{2} - \frac{\sqrt{2}}{4} - \frac{\sqrt{6}}{4}\right), \\ y_{23} &= r\left(\frac{\sqrt{2}}{4} - \frac{\sqrt{6}}{4}\right). \end{aligned} \tag{A.12}$$

Now, the length of the chords c_1, c_2, c_3 is found through

$$c_k^2 = \left(x_{ik} - x_{jk}\right)^2 + \left(y_{ik} - y_{jk}\right)^2. \tag{A.13}$$

For the case of c_3

$$c_3^2 = \left(x_{13} - x_{23}\right)^2 + \left(y_{13} - y_{23}\right)^2. \tag{A.14}$$

Substituting values, c_3 is given by

$$c_3 = -\frac{r}{\sqrt{2}} + r\sqrt{\frac{3}{2}}. \tag{A.15}$$

This is (16) of Section 3 assuming $d = r$ and is different from the value of c_3 given in [61].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

Armando Céspedes-Mota gives thanks to CONACyT-México, the research council of Mexico, for the scholarship granted to him. This research was partially supported by the 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] I. Mahgoub and M. Ilyas, *Sensor Network Protocols*, CRC Press, 2006.

[4] 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.

[5] 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*, vol. 12, no. 7, pp. 1913–1928, 2012.

[6] 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.

[7] Y. Chen, C. Chin, and D. Deng, "Efficient localization algorithm in wireless ad hoc sensor networks by utilizing radical centers," in *Proceedings of the ICC 2012 - 2012 IEEE International Conference on Communications*, pp. 27–31, Ottawa, ON, Canada, June 2012.

[8] K. Saleem, N. Fisal, and J. Al-Muhtadi, "Empirical studies of bio-inspired self-organized secure autonomous routing protocol," *IEEE Sensors Journal*, vol. 14, no. 7, pp. 2232–2239, 2014.

[9] 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.

[10] 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.

[11] K. P. Eswaran and R. E. Tarjan, "Augmentation problems," *SIAM Journal on Computing*, vol. 5, no. 4, pp. 653–665, 1976.

[12] 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.

[13] Z. Yang and Y. Liu, "Understanding node localizability of wireless ad hoc and sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 8, pp. 1249–1260, 2012.

[14] T. Chen, Z. Yang, Y. Liu, D. Guo, and X. Luo, "Localization-oriented network adjustment in wireless ad-hoc and sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 146–155, 2014.

[15] D. Dong, X. Liao, K. Liu, Y. Liu, and W. Xu, "Distributed coverage in wireless ad hoc and sensor networks by topological graph approaches," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 61, no. 10, pp. 1417–1428, 2012.

[16] B. Panigrahi, S. De, B. S. Panda, and J. L. Luk, "Network lifetime maximising distributed forwarding strategies in ad hoc wireless sensor networks," *IET Communications*, vol. 6, no. 14, pp. 2138–2148, 2012.

[17] P. Di Lorenzo and S. Barbarossa, "A bio-inspired swarming algorithm for decentralized access in cognitive radio," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 6160–6174, 2011.

[18] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, "Bio-inspired decentralized radio access based on swarming mechanisms over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 12, pp. 3183–3197, 2013.

[19] G. Kumar and M. K. Rai, "An energy efficient and optimized load balanced localization method using CDS with one-hop neighbourhood and genetic algorithm in WSNs," *Journal of Network and Computer Applications*, vol. 78, pp. 73–82, 2017.

[20] 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.

[21] 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.

[22] V. Feoktistov, *Differential Evolution: In Search of Solutions*, vol. 5 of *Springer Optimization and Its Applications*, New York, NY, USA, Springer, 2006.

[23] 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.

[24] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution-An updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.

[25] A. W. Mohamed and A. S. Almazyad, "Differential evolution with novel mutation and adaptive crossover strategies for solving large scale global optimization problems," *Applied Computational Intelligence and Soft Computing*, vol. 2017, Article ID 7974218, 18 pages, 2017.

[26] 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*, pp. 51–54, Shenzhen, China, 2010.

[27] W. Gong and Z. Cai, "An empirical study on differential evolution for optimal power allocation in WSNs," in *Proceedings of the 2012 8th International Conference on Natural Computation (ICNC)*, pp. 635–639, Chongqing, Sichuan, China, May 2012.

[28] Y. Xu, J. Fang, W. Zhu, and W. Cui, "Differential evolution for lifetime maximization of heterogeneous wireless sensor networks," *Mathematical Problems in Engineering*, vol. 2013, Article ID 172783, 12 pages, 2013.

[29] S. Gundry, J. Zou, J. Kusyk, C. S. Sahin, and M. U. Uyar, "Markov chain model for differential evolution based topology control in MANETs," in *Proceedings of the 2012 35th IEEE Sarnoff Symposium*, pp. 1–5, Newark, NJ, USA, May 2012.

[30] S. Gundry, J. Kusyk, J. Zou, C. S. Sahin, and M. U. Uyar, "Performance evaluation of differential evolution based topology control method for autonomous MANET nodes," in *Proceedings of the 2012 IEEE Symposium on Computers and Communications (ISCC)*, pp. 000228–000233, Cappadocia, Turkey, July 2012.

[31] 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.

[32] A. Céspedes-Mota, G. Castañón, A. F. Martnez-Herrera, and L. E. Cárdenas-Barrón, "Optimization of the distribution and localization of wireless sensor networks based on differential evolution approach," *Mathematical Problems in Engineering*, Article ID 7918581, 12 pages, 2016.

[33] Y. Xu, X. Wang, and H. Zhang, "Improved differential evolution to solve the two-objective coverage problem of wireless sensor networks," in *Proceedings of the 28th Chinese Control and Decision Conference, CCDC 2016*, pp. 2379–2384, May 2016.

[34] Y. Xu, T. Duan, W. Chen, and J. Chen, "Based on differential evolution to research the control problem of area-coverage in WSNs," in *Proceedings of the 29th Chinese Control and Decision Conference, CCDC 2017*, pp. 7298–7303, May 2017.

[35] 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*, vol. 65, no. 3, pp. 1676–1679, 2015.

[36] B. Nancharaiah and B. C. Mohan, "Modified ant colony optimization to enhance manet routing in adhoc on demand distance vector," in *Proceedings of the 2014 2nd International Conference on Business and Information Management, ICBIM 2014*, pp. 81–85, January 2014.

[37] F. Hajjej, R. Ejbali, and M. Zaied, "Quality of services based routing using evolutionary algorithms for wireless sensor network," in *Proceedings of the 15th International Conference on Intelligent Systems Design and Applications, ISDA 2015*, pp. 237–242, December 2015.

[38] X. Yulong, W. Xiaopeng, and Z. Han, "Comparative study on the optimal path problem of wireless sensor networks," in *Proceedings of the 13th IEEE International Conference on Mechatronics and Automation, IEEE ICMA 2016*, pp. 2234–2239, August 2016.

[39] D. S. Deif and Y. Gadallah, "An ant colony optimization approach for the deployment of reliable wireless sensor networks," *IEEE Access*, vol. 5, pp. 10744–10756, 2017.

[40] H. Feng and J. Qi, "Radio frequency identification networks planning using a new hybrid evolutionary algorithm," in *Proceedings of the 5th International Conference Advanced Communication Technology (ICACT)*, pp. 179–188, 2013.

[41] P. Li, Y. Wang, J. Hu, and J. Zhou, "Sensors distribution optimization for impact localization using NSGA-II," *Sensor Review*, vol. 35, no. 4, pp. 409–418, 2015.

[42] J. M. L. Gutiérrez, J. A. G. Pulido, and M. A. V. Rodríguez, "A new realistic approach for the relay node placement problem in wireless sensor networks by means of evolutionary computation," *Ad-Hoc and Sensor Wireless Networks*, vol. 26, no. 1-4, pp. 193–209, 2015.

[43] S. M. Jameii, K. Faez, and M. Dehghan, "AMOF: adaptive multi-objective optimization framework for coverage and topology control in heterogeneous wireless sensor networks," *Telecommunication Systems*, vol. 61, no. 3, pp. 515–530, 2016.

[44] S. Özdemir, B. A. Attea, and Ö. A. Khalil, "Multi-objective evolutionary algorithm based on decomposition for energy efficient coverage in wireless sensor networks," *Wireless Personal Communications*, vol. 71, no. 1, pp. 195–215, 2013.

[45] M. Le Berre, M. Rebai, F. Hnaien, and H. Snoussi, "A specific heuristic dedicated to a coverage/tracking bi-objective problem for wireless sensor deployment," *Wireless Personal Communications*, vol. 84, no. 3, pp. 2187–2213, 2015.

[46] B. A. A. Attea, E. A. Khalil, and A. Cosar, "Multi-objective evolutionary routing protocol for efficient coverage in mobile sensor networks," *Soft Computing*, vol. 19, no. 10, pp. 2983–2995, 2015.

[47] R. Murugeswari and S. Radhakrishnan, "Discrete multi-objective differential evolution algorithm for routing in wireless mesh network," *Soft Computing*, vol. 20, no. 9, pp. 3687–3698, 2016.

[48] S. Singh and R. M. Sharma, "Optimization techniques in wireless sensor networks," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, ICTCS '16*, pp. 1–7, New York, NY, USA, March 2016.

[49] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[50] K. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, Norwell, Mass, USA, 1999.

[51] R. S. Cahn, *Wide Area Network Design: Concepts and Tools for Optimization*, The Morgan Kaufmann Series in Networking, Morgan Kaufmann Publishers, 1998.

[52] A. S. Nepomniaschaya, "An associative version of the Prim-Dijkstra algorithm and its application to some graph problems," in *Perspectives of System Informatics*, vol. 1181 of *Lecture Notes in Computer Science*, pp. 203–213, Springer Berlin Heidelberg, Berlin, Germany, 1996.

[53] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society For Industrial & Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[54] 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.

[55] J. Jia, J. Chen, G. Chang, Y. Wen, and J. Song, "Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius," *Computers & Mathematics with Applications. An International Journal*, vol. 57, no. 11-12, pp. 1767–1775, 2009.

[56] T. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

[57] A. L. Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering*, Pearson/Prentice Hall, 2008.

[58] S. Epp, *Discrete Mathematics with Applications*, Brooks/Cole-Thompson Learning, Belmont, CA, USA, 3rd edition, 2004.

[59] 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.

[60] E. W. Weisstein, "Circle-Circle Intersection," http://mathworld.wolfram.com/Circle-CircleIntersection.html, 2017.

[61] P. M. Fewell, "Area of common overlap of three circles," Technical report, DSTO Defence Science and Technology Organization, Edinburg South Australia, 2006.

[62] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 482–500, 2012.

[63] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Labs Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957.

[64] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[65] F. Lezama, G. Castañón, and A. M. Sarmiento, "Routing and wavelength assignment in all optical networks using differential evolution optimization," *Photonic Network Communications*, vol. 26, no. 2-3, pp. 103–119, 2013.

[66] F. Lezama, G. Castañón, A. M. Sarmiento, and I. B. Martins, "Differential evolution optimization applied to the routing and spectrum allocation problem in flexgrid optical networks," *Photonic Network Communications*, vol. 31, no. 1, pp. 129–146, 2016.

[67] S. M. K. Heris, "Generic implementations of MOEA/D and NSGA-II," http://yarpiz.com, 2018.

[68] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[69] J. Q. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[70] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.