

Article

A Coral Reef Algorithm Based on Learning Automata for the Coverage Control Problem of Heterogeneous Directional Sensor Networks

Ming Li ^{1,2,3,*}, Chunyan Miao ^{3,†} and Cyril Leung ^{4,†}

Received: 20 September 2015; Accepted: 1 December 2015; Published: 4 December 2015

Academic Editor: Leonhard M. Reindl

¹ Detection and Control of Integrated Systems Engineering Laboratory in Chongqing Technology and Business University, Chongqing 400067, China

² School of Computer Science and Information Engineering, Chongqing Technology and Business University, Chongqing 400067, China

³ School of Computer Engineering, Nanyang Technological University, Nanyang Avenue 639798, Singapore; asycymiao@ntu.edu.sg

⁴ Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada; cleung@ntu.edu.sg

* Correspondence: sshjlm@gmail.com; Tel.: +86-136-4841-8549

† These authors contributed equally to this work.

Abstract: Coverage control is one of the most fundamental issues in directional sensor networks. In this paper, the coverage optimization problem in a directional sensor network is formulated as a multi-objective optimization problem. It takes into account the coverage rate of the network, the number of working sensor nodes and the connectivity of the network. The coverage problem considered in this paper is characterized by the geographical irregularity of the sensed events and heterogeneity of the sensor nodes in terms of sensing radius, field of angle and communication radius. To solve this multi-objective problem, we introduce a learning automata-based coral reef algorithm for adaptive parameter selection and use a novel Tchebycheff decomposition method to decompose the multi-objective problem into a single-objective problem. Simulation results show the consistent superiority of the proposed algorithm over alternative approaches.

Keywords: directional sensor network; coverage control; coral reef algorithm; learning automata; multi-objective optimization

1. Introduction

In recent years, with the development of micro-electromechanical systems (MEMS) directional sensor networks (DSNs) have received much attention due to their wide and significant applications [1], which offer important economic benefits. A DSN is a wireless network which is equipped with directional sensors such as video sensors, ultrasound and infrared sensors. Differing from the sensor nodes of the traditional wireless sensor networks (WSN) which have omni-directional sensing ranges, the sensing range of DSN sensor nodes are restricted by their directions and specific angular dimensions.

Since directional sensing and directional communication have great impact on the performance of a DSN, several difficulties have emerged in DSN protocol design. One of the most fundamental problems is the coverage issue. Although there are extensive amounts of research about the coverage problems in WSNs, however, these WSN research results cannot be directly applied to DSNs.

In this paper, a coverage control algorithm based on a multi-objective optimization method is proposed to reduce the power consumption of networks and prolong the lifetime of the network. It should be noted that two assumptions which are generally considered in the published papers about

the coverage problems of DSNs are relaxed in order to formulate a more realistic DSN deployment problem. The first assumption concerns the types of sensor nodes in the DSN. Indeed, in most research works, sensor nodes are identical in terms of energy and hardware complexity. We consider in the paper a heterogeneous directional sensor network (HDSN), in which each node has different sensing radius, communication radius and field of angle. To the best of our knowledge, research on the network coverage in HDSNs is less evident. The second assumption is the sensing requirement within the monitored area. In most of the published papers, the whole supervised region is considered to be of the same importance. In other words, the sensing requirement is uniformly distributed within the area [2]. However, this assumption is not always valid for some applications. For example, in water quality monitoring, the risky area near the chemical deposits or animal/human habitats needs high detection levels, while for other areas a low detection level is sufficient. To sum up, the key contributions of this paper are as follows:

- Firstly, we propose a realistic case study of a coverage algorithm for the placement field characterized by a geographical irregularity of the sensed events in a HDSN by ensuring the connectivity of the network, reducing the cost of deployment.
- Secondly, this paper proposes a learning automata-based coral reef algorithm for adaptive parameter selection. Learning capabilities are used in the coral reef algorithm to select its parameters. As a result, the convergence rate and the search abilities of the algorithm are enhanced. Experimental results show the superiority of the proposed algorithm.
- Thirdly, a novel Tchebycheff decomposition approach is introduced to decompose the multi-objective problem into a single-objective problem. Theoretical proofs and numerical results indicate the efficiency of the proposed method.

The rest of the paper is organized as follows: Section 2 reviews some related works on DSN deployment issues. Mathematical formulation of the coverage model in HDSN is formalized in Section 3. In Sections 4 and 5, fundamentals of the coral reef algorithm and learning automata are introduced, respectively. An improved Tchebycheff decomposition method and the proposed learning automata-based coral reef algorithm are detailed in Section 6. The experimental results of the proposed approaches are given in Section 7. Finally, Section 8 ends the paper with some concluding remarks and future research directions.

2. Previous Works

As some of the most promising tools, multi-objective optimization algorithms have been applied to wireless sensor networks to balance various trade-offs among different conflicting objectives. A detailed survey can be found in [3]. By decomposing the multi-objective optimization problem into a set of single objective subproblems which are tackled in parallel, the Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) [4] has been used in coverage optimization problems relating to WSNs. In [5], the authors proposed a generalized subproblem-dependent heuristic (GSH) and successfully hybridized it with MOEA/D for tackling the multi-objective dense deployment and power assignment problem in WSNs. In [6], the authors deal with how to efficiently deploy energy-harvesting relay nodes in a WSN. However, it should be noted that the weighted Tchebycheff Approach used in the above two papers is very sensitive to the scale of the objectives. To correct this situation an enhanced version of the Tchebycheff Approach is proposed in this paper.

The parameter settings of the evolutionary algorithm have a great impact on the performance of the algorithm. The self-adaptive parameters adjustment method is an effective way to improve the diversity of the individuals and avoid premature convergence. Since learning automata are adaptive decision making devices that run in an unknown environment and progressively enhance their performance via a learning process, it has been proved that the method that combines the learning ability of the learning automata with the parameter adjustment of the evolutionary algorithm will enhance the efficiency of the evolutionary algorithm [7,8]. Inspired by this idea, a learning automata-based evolutionary algorithm is proposed to solve our multi-objective problem.

In recent years, a considerable amount of work on coverage in directional sensor networks (DSNs) has been reported in the literature [9]. In [10], the authors designed two greedy-based scheduling algorithms that aim to select the appropriated sensor direction and sensing range in a way to meet the requirements of the target coverage problem and at the same time maximize the network lifetime. However, the performance of the algorithms is extremely dependent on the closeness of the initial candidates to the optimal solution. In [11–14], they all aimed to maximize the network lifetime by finding cover sets in each of which the directions cover all the targets. In [15] and [16], the authors utilized the geometrical features of a Voronoi diagram to propose a distributed Voronoi-based self-redeployment algorithm aiming to improve the overall field coverage of directional sensor networks. In [17], the authors proposed heuristic algorithms to solve the multiple directional cover set problem. To model an application scene more accurately, Ma et al. have proposed a 3D sensor coverage-control model with tunable orientations [18]. In [19], learning automata algorithms are employed to find a near-optimal solution for solving the target coverage problem in DSNs. However, it should be noted that all the abovementioned algorithms assume that the coverage requirement is uniformly distributed within the area, therefore these algorithms cannot perform well in real scenarios characterized by the geographical irregularity of the sensed events. To solve such a problem, several algorithms were proposed in [20–22]. In [20], Wang *et al.* proposed to choose a minimum subset of directional sensors that is able to satisfy the prescribed priorities of all the targets. In [21], the authors proposed a greedy-based scheduling algorithm to find a sequence of feasible cover sets in order to prolong the network lifetime. In [22], the authors propose a learning automata-based algorithm to organize the directional sensors into several cover sets in such a way that each cover set can satisfy coverage quality requirement of all the targets.

While coverage maintenance protocols for DSNs have been extensively studied in the literature, no attempt, however, has been made on tackling coverage problems, energy consumption and connectivity problems for DSNs, especially for DSNs with heterogeneous nodes in terms of field of angle view, sensing and communication radius. To this end, a multi-objective optimization-based coverage control algorithm for HDSNs is proposed to deal with such problems in HDSNs. As the performance of this algorithm is sensitive to the proper setting of the parameters, the learning capabilities of the learning automata are employed for adaptive parameter selection. Moreover, an improved Tchebycheff decomposition method is proposed to assess the solutions obtained from the evolutionary algorithm, which can provide a collection of non-dominant solutions in a single run. Besides, to our knowledge, no attempt has adopted multi-objective evolutionary algorithms for coverage and efficient design issues in HDSNs on the condition of the geographical irregularity of the sensed events. Thus, in this paper we deal with the development of an evolutionary algorithm-based coverage control protocol that efficiently sets up the minimum number of active sensors, while satisfying the coverage and connectivity requirements. Consequently, unwanted interference at the MAC layer is avoided and energy consumption is reduced. The characteristics optimized by the evolutionary algorithm in this paper include network coverage, network connectivity and the status of the sensor nodes (whether they are active or inactive).

3. Problem Statements

This section first demonstrates the system models and assumptions based upon which the problem is formulated, and then presents the problem formulation.

3.1. System Model and Assumptions

The system model and general assumptions are as follows:

- (1) There is a rectangular monitoring area A in which the sensors are placed. In order to reduce computational complexity of the problem, A is divided into $m \times n$ uniform consecutive small grids and each size of grid is 1.
- (2) A DSN is formed by N heterogeneous directional sensors $s_i (i = 1, 2, \dots, N)$ in terms of the sensing radius, communication radius and field angles of view.

- (3) The directional sensor nodes are scattered randomly and uniformly within the field. Each sensor has only one sensing radius and sensing direction. All the sensors are static after deployment and their locations are known.
- (4) If a sensor S_a is within the communication radius of a sensor S_b , then S_b can communicate with sensor S_a .
- (5) A target can be sensed by a sensor node if the target is located within the sensing radius and field of view of the node.
- (6) The geographical distribution of the sensing events or targets is supposed to be non-uniformly distributed within the zone. That is to say, the subareas of the monitoring area are not of the same importance. The area where targets or monitoring events appear frequently is called a hot spot area.

3.2. Problem Formulation

The objective of the coverage control algorithm is to obtain a set of sensor nodes that fulfill the following optimization criteria:

(1) Maximize the coverage rate of area. Since the targets are usually non-uniformly distributed in the area, more sensors are needed in some subareas where the targets tend to be clustered around to enhance the quality of service for the WSN and fault tolerance of the network. Based upon this, we need more sensors to cover any hot spots of the supervised area (often called k -coverage), and in the remaining area, we only need to guarantee that there are some sensors which can monitor it. In the mathematical description, the coverage rate of the area is defined as the percentage of the covered grids over the total grids of A . It is evaluated as follows:

Max:

$$f_1 = \frac{\sum_{(x,y) \in A} c(x,y)}{W}$$

where W is the number of grids in surveillance zone A and $c(x, y)$ is calculated using following equation:

$$c(x, y) = \begin{cases} 1 & \text{grid}(x, y) \text{ is achieved coverage requirement} \\ 0 & \end{cases}$$

where grid (x, y) is achieved coverage requirement means if the grid (x, y) is located within a hot spot subarea, the coverage requirement is K -coverage ($K > 1$); otherwise, the coverage requirement is covered by one sensor node.

(2) Minimize the number of working sensors nodes, or equivalently, minimize the financial cost of working nodes, which is defined as:

Min:

$$f_2 = \frac{\left| \sum_{s_i \in S'} s_i \right|}{N}$$

where S' is set of working sensors, and $|s_i|$ is evaluated in the following way:

$$|s_i| = \begin{cases} 1 & s_i \in S' \\ 0 & \text{otherwise} \end{cases}$$

(3) Maximize the connection status of working nodes, *i.e.* for each node s_i ($i = 1, 2, \dots, N$) belonging to a set of S' , if s_i is located in the hot area, K -connect ($K > 1$) is necessary for network reliability and fault tolerant; otherwise, in the rest of the monitor area, that is non-hot area, s_i can connect another sensor in the set of S' , that is OK. Connect status of working nodes is denoted as:

Max:

$$f_3 = \frac{\left| \sum_{s_i \in S'} Cn(s_i) \right|}{|S'|}$$

where $Cn(s_i)$ is the connect status of sensor s_i and is defined as:

$$Cn(s_i) = \begin{cases} 1 & s_i \text{ located within hotspot region and } s_i \text{ is } k\text{-connected } (k > 1) \\ 1 & s_i \text{ located at non-hotspot region and } s_i \text{ is 1-connected} \\ 0 & \text{otherwise} \end{cases}$$

In the above formula, if s_i is located within the hotspot region and it is k -connected (that means it has more than one sensor within its communication radius), then its value of $Cn(s_i)$ is 1. If s_i is placed within the non-hotspot region and it has only one sensor to directly communicate with, then its value of $Cn(s_i)$ is 1. Except for the cases presented, the value of $Cn(s_i)$ is 0.

The problem we aim to solve is multi-objective optimization, which can be described as follows:

$$F = (\max(f_1), \min(f_2), \max(f_3))$$

The problem presented above is a NP -hard problem [23]. Its solution space is as large as 2^N . To resolve such a combinatorial optimization problem, it is difficult to use an exact method as Branch & Bound due to its exponential complexity. In this paper we use a bio-inspired algorithm to solve this multi-objective optimization problem.

4. Original Coral Reef Optimization Algorithm

The coral reef optimization (CRO) algorithm is a novel bio-inspired algorithm for solving optimization problems, which mimics the behaviors of corals' reproduction and coral reef formation. It was originally introduced by Salcedo and his colleagues [24] and thereafter has found a lot of applications in the optimization of mobile network deployment problems [25], global solar radiation prediction [26], wind speed prediction [27,28], *etc.* Algorithm 1 summarizes the skeleton of the CRO algorithm applied to a combinatorial optimization problem for minimizing one objective function.

Algorithm 1 The CRO algorithm

1. Initialize the reef
 2. While not stopCriteria () do {
 3. A fraction of p_k coral larvae formed by external reproduction(also called broadcast spawning)
 4. The rest of $(1-p_k)$ coral larvae formed by internal sexual reproduction(brooding)
 5. Larvae setting procedure
 6. Asexual reproduction
 7. Depredation in polyp phase
 8. End while }
 9. Return solution to the problem
-

In the CRO implementation there are two main phases [28]:

Phase 1: Initialization of CRO parameters.

The purpose of CRO initialization is to set parameters to fill in the algorithm. The main control parameters of CRO are presented as follows: coral reef, Λ , consisting of a $T \times M$ grid similar to the population size in the evolutionary algorithm (EA), the grids are selected randomly and can assign a coral or colony of coral, representing a solution to the given problem, the rate ρ_0 between the selected grids and not selected ones which is an important factor to control the exploration ability of the algorithm. The health function f is similar to the fitness of EA. The underlying idea behind CRO is that as the reef progresses, the more healthy the corals are (which represents a better solution to the mentioned problem), the better the chance they can survive.

Phase 2: reef formation.

The reef formation consists of the following steps:

(1) Broadcast Spawning (external sexual reproduction):

a. Choose a fraction of the existing corals p_k to become broadcast spawners. The remaining ones will be formed by internal sexual reproduction.

b. Broadcast spawner couples will produce a coral larva by sexual crossover. It should be noted that once two corals have been selected to be the parents of a larvae, they are not selected any more in Broadcast Spawning phase.

(2) Brooding (internal sexual reproduction)

As Step (1) describes, the fraction $1 - p_k$ of corals will form in this stage by means of a random mutation. The newly produced larvae together with that of larvae formed in Step (1) are released to the water.

(3) Larvae setting

When all the larvae are formed by broadcast spawning or by brooding, the process of setting and growing in the reef will begin. The health function of each larva is evaluated firstly. Then, each larvae will settle down in the grid (i, j) of the reef randomly. If the grid is free space in the reef, the coral can grow whatever its value of health function. However, if the grid is already occupied by a coral, the new larvae will set in the occupied grid only when its health function is better than the existing one. A variable κ is defined to represent the maximum number of attempts for the larvae to set in the reef before it is preyed on by other animals.

(4) Asexual reproduction

According to the values of health function, the existing corals in the reef are rearranged. A proportion F_a of the existing corals will copy itself and attempt to settle in a different part of reef by Step (3) introduced above.

(5) Depredation in polyp phase

In the process of reef formulation, some corals may die. As a result, space is freed up for the newly generated corals. The probability of the depredation process is F_d , and only be applicable for a proportion F_d of corals with the worst health function values. It should be noted that, $F_a + F_d \leq 1$, that is to say, no overlap between the coral set formed by Step (4) and depredated by Step (5).

5. Learning Automata

A learning automaton [29] is considered as an adaptive decision making unit that is continually interacting with the environment to learn how to choose an optimal action from its action set. The action selected by the learning automaton is evaluated by the environment and a reinforcement signal will return as a feedback for the selected action. The automaton uses the feedback to update its action probability vector set. By continuing the above process, the optimal action from the action-set will be found and the average reward feedback signals received from the environment could be maximized [30,31]. The four-tuple $\{\alpha, \beta, P, T\}$ can be used to represent a learning automaton, where $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ is a set of available actions which can be selected by the learning automaton. $\beta = \{\beta_1, \beta_2, \dots, \beta_r\}$ signifies the set of the potential values of the reinforcement signal, $p = \{p_1, p_2, \dots, p_r\}$ is a state probability vector and each p_r associated with the selected action α_r , T is a learning algorithm used to update the state probability vector

In the basic learning algorithm, named Linear Reward Penalty Algorithm, the action probability vector is updated applying to Equation (1) for favorable responses and Equation (2) otherwise:

$$p_i(n+1) = p_i(n) + \alpha[1 - p_i(n)] \quad p_j(n+1) = (1 - \alpha)p_j(n) \quad \forall j, j \neq i \quad (1)$$

$$p_i(n+1) = (1-b)p_i(n) \quad p_j(n+1) = \frac{b}{r-1} + (1-b)p_j(n) \quad \forall j, j \neq i \quad (2)$$

where r is the cardinality of the action set, a and b are the reward and penalty parameters, respectively.

6. The Proposed Algorithm

This section details a novel fitness evaluation method for the multi-objective optimization problem and the proposed learning automata-based coral reefs algorithm, respectively. It should be noted that the underlying idea behind the proposed method might shed light on the design of another heuristic method for multi-objective optimization problems.

6.1. Proposed Tchebycheff Decomposition Approach

In this subsection, a newly improved multi-objective optimization fitness evaluation method based on the Tchebycheff decomposition approach is proposed. Firstly, a few concepts for the multi-objective optimization are defined to facilitate our presentation. A general multi-objective optimization problem is described as [32]:

Min:

$$\begin{aligned} F(x) &= ((f_1(x), f_2(x), \dots, f_t(x)))^T \\ \text{s.t. } x &\in S, x = ((x_1, x_2, \dots, x_n))^n \end{aligned} \quad (3)$$

where $f_1(x), f_2(x), \dots, f_t(x)$ are the t objective functions, (x_1, x_2, \dots, x_n) are the n optimization parameters and $j \in \{1, 2, \dots, t\}$ $S \in R^n$ is the solution space. The definition of Pareto dominance and Pareto optimality are presented as follows:

Definition 1. (Pareto dominance) [33]: Suppose x and y are two decision variables, x is said to dominate y , denoted by $x \succ y$, if and only if $f_i(x) \geq f_i(y)$ for every $i \in \{1, 2, \dots, t\}$ and $f_j(x) > f_j(y)$ for at least one index $j \in \{1, 2, \dots, t\}$.

Definition 2. (Pareto optimality) [33]: $x^* \in S$ is said to be Pareto optimal (or non-dominated) if there is no another $x \in S$ so that x dominates x^* . The set of all Pareto optimal solutions in the decision space is called the Pareto optimal Set (PS).

An improved fitness evaluation based on Tchebycheff decomposition approach is:

$$\min T_s(f(x) | (\lambda^i, y^*, y^\#)) = \max_{1 \leq j \leq t} \left\{ \lambda_j^i \left(\frac{f_j(x) - y_j^*}{y_j^\# - y_j^*} \right) \right\} \quad (A)$$

where λ^i , y^* , $y^\#$ are constant values; $\lambda^i = (\lambda_1^i, \lambda_2^i, \dots, \lambda_t^i)$ is the aggregation coefficient vector, $\sum_{j=1}^t \lambda_j^i = 1$; $y^* = (y_1^*, y_2^*, \dots, y_t^*)$ and $y^\# = (y_1^\#, y_2^\#, \dots, y_t^\#)$ are the lower bounds and upper bounds of the objective functions, $\forall j, j \in \{1, 2, \dots, t\}$, $y_j^* = \min\{f_j(x)\}$, $y_j^\# = \max\{f_j(x)\}$. y^* and $y^\#$ are introduced to compare the values of the different objective functions.

In the improved Tchebycheff approach, let $\lambda^1, \dots, \lambda^W$ be the aggregation coefficient vectors, subproblem i corresponds to the coefficient vector, where $\lambda_j^i \geq 0, j = 1, \dots, t, \sum_{j=1}^t \lambda_j^i = 1, i = 1, \dots, W$. t is the number of objectives, and W is the number of subproblems. Population size W (i.e., the number of subproblems) and aggregation coefficient vectors $\lambda^1, \dots, \lambda^W$ are controlled by t and an integer H . More precisely, $\lambda^1, \dots, \lambda^W$ are all the weight vectors in which each individual weight takes a value from $\{0/H, 1/H, \dots, H/H\}$. Therefore, the number of such vectors is $W = C_{H+t-1}^{t-1}$.

We use Formula (A) to change them into the same range of value in order to make comparisons. The proof of Proposition (A) will be given in Theorem 2.

Lemma: For $\forall y^1, y^2 \in R^n$, if $y_j^1 \leq y_j^2$ and $y^1 \neq y^2$ for all j ($j = 1, 2, \dots, t$), then $T_s(y^1) < T_s(y^2)$.

Proof: Let us take y^1, y^2 into the Formula (A), note that the two parts in the right side of the equation for the Formula (A) are positive. Therefore, it is easy to get the conclusion that $T_s(y^1) < T_s(y^2)$. \square

Theorem 1. If x^* is an optimal solution for the single objective Problem (A), then x^* is the Pareto optimal solution for the Problem (1).

Proof: We will prove it by the contraction method. Suppose x^* is not Pareto optimal solution to the Problem (1), according to Definition 2, $\exists x' f(x') < f(x^*)$. From Definition 1, we know, $\forall j \in \{1, 2, \dots, n\}$, $f_j(x') \leq f_j(x^*)$ and $f_j(x') \neq f_j(x^*)$. From the Lemma, we can get $T_s(f_j(x')) < T_s(f_j(x^*))$, that is to say, x^* is not an optimal solution to the Problem (A). Therefore, it apparently contradicts the hypothesis that x^* is the single objective Problem (A). \square

Theorem 2. If x^* is a Pareto optimal solution to Problem (1), then $\exists \lambda^* = (\lambda_1^*, \lambda_2^*, \dots, \lambda_t^*)$ makes $T_s(f(x^*)) \leq T_s(f(x))$, wherein $\forall \lambda_j^* \geq 0, \sum_{j=1}^t \lambda_j^* = 1$.

Proof: We will prove it by the contraction method. Suppose that $\exists \lambda^* T_s(f(x^*)) \leq T_s(f(x))$. In other words, for $\forall \lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$, $\exists x', T_s(f(x')) < T_s(f(x^*))$, where $\forall \lambda_i \geq 0, \sum_{i=1}^t \lambda_i = 1$. By Formula (A), and thus $\forall j, f_j(x') < f_j(x^*)$, that is to say $f(x') < f(x^*)$. Therefore, it apparently contradicts the hypothesis. \square

From the theorems presented above, we can see that the optimal solutions to the single objective function optimization Problem (A) are the Pareto optimal solutions to the multi-objective function. By using Theorem 2, a multi-objective function optimization problem can be changed into a single-objective function optimization problem.

6.2. Proposed Hybrid Learning Automata—CRO Algorithm

The coral reef algorithm described in Section 4 shows that the parameters of CRO, that is, Broadcast Spawning radio F_b and brooding radio $1-F_b$ play important roles in the exploration/exploitation capabilities of the algorithm. In this subsection a new learning automata mechanism for adaptive parameter selection for the CRO is given.

The proposed algorithm, that is the learning automata based coral reefs optimization algorithm (LACRO for short), performs like the original CRO algorithm with the auxiliary section at the end of the each iteration to select CRO parameters value. In order to enhance the ability of the original CRO algorithm, we borrow the idea from the differential evolution [34] and the genetic algorithm to make some modifications to the basic process of the CRO algorithm. The great difference between the original CRO algorithm and the proposed method lies in that the mutation of the selected brooding coral is controlled by the parameter B_d , which is adaptive and adjusted by learning automata according by the diversity and evolutionary status of the population. In contrast to the proposed algorithm, the fraction of corals that reproduce by brooding is $1-F_b$ in the original CRO. In addition to the parameter of the brooding radio controlled by the learning automata, the parameter of Broadcast Spawning radio F_b is also adaptive and adjusted by the learning automata. The details of the proposed algorithm are presented as follows:

Step 1: Coral reefs initialization. Set the initial parameter values for the coral reef population, the learning automata and determine the way of encoding the solution. Considering the characteristics of the solved problem, we use the binary code method. A solution to the problem is demonstrated in Figure 1, where “1” stands for the sensor which is selected to be in the working status, otherwise, the bit is “0”.



Figure 1. Solution representation in LACRO.

Step 2: Equally discretize two parameters, namely, the value of F_b and B_d , into the m_1 distant value and m_2 distant value, respectively. This method is called the adventurous method [7] which allows a parameter to change radically from one end of its range to the other in the consecutive iterations and not to be restricted by its previous value.

Step 3: To equip each parameter with one learning automaton LA_i ($i = F_b, B_d$), in which the corresponding actions number is m_1 and m_2 , respectively. During every iteration, LA_i ($i = F_b, B_d$) chooses one from its action set, then the corresponding value of the selected action will be set as the new value for the parameter. In our proposed method, all the coral reefs have the same values for the parameter F_b and B_d . In each iteration the roulette-wheel selection method is used to select the corresponding action of each learning automaton.

The pseudo codes for LACRO are listed in Algorithm 2:

Algorithm 2. Pseudocode for the proposed LACRO algorithm

Input: parameters F_b and B_d , success threshold θ , the parameters of learning automata and CRO

Output: The solutions to the problem

1. Coral reefs initialization
 2. Equally divide parameter F_b into m_1 parts and parameter B_d into m_2 parts
 3. Assign LA_{F_b} with m_1 actions to F_b and the probability of each action is $1/m_1$
 4. Assign LA_{B_d} with m_2 actions to B_d and the probability of each action is $1/m_2$
 5. N is the number of coral reefs and $N'(t)$ is the number of coral reefs which their fitness has been improved since iteration $k-1$
 6. While not stopCriteria () do {
 7. Using the roulette-wheel method to select Active Action p_{F_b} for LA_{F_b} and p_{B_d} for LA_{B_d}
 8. Select the value of parameter F_b and B_d according to p_{F_b} and p_{B_d} , assuming the selected value is f_b and b_d respectively
 9. A fraction of f_b coral larvae formed by external reproduction(also called broadcast spawning)
 10. The rest of b_d coral larvae formed by internal sexual reproduction(brooding)
 11. Fitness evaluation of newly produced coral larvae by broadcast spawning and brooding
 12. If $\frac{N'(t)}{N} \geq \theta$ {
 13. Reward the selected action for LA_{F_b} and LA_{B_d} according to Equation (1)
 13. Reward the selected action for LA_{F_b} and LA_{B_d} according to Equation (1)
 14. Else
 15. Punish the selected action for LA_{F_b} and LA_{B_d} according to Equation (2) }
 16. Larvae setting procedure
 17. Asexual reproduction
 18. Depredation in polyp phase
 19. End while }
-

Step 4: Fitness evaluation and learning automata updating. Please see Subsection 6.1 for the detailed fitness evaluation method. The learning automata are updated according to the evaluation result of parameter selection. It is considered to be a “successful” selection on the condition that the fraction of improved coral reef in the previous iteration is greater than the specific threshold. In this case, the two learning automata are rewarded; otherwise they are penalized. As a result, the actions with greater probability stand a good chance of being selected.

Step 5: Larvae setting process.

Step 6: Asexual reproduction process.

Step 7: Depredation in polyp phase.

Steps 5–7 are the same as described in Section 4. Please see the relevant parts for detailed information.

Step 8: To determine whether the goal of stopping criteria is achieved. If it is true, return the best solution, otherwise, return to Step 4.

7. Performance Evaluation

In this section, the performance of the proposed algorithm is investigated. Firstly, performance comparison in test functions including single-objective functions and multi-objective functions were made. Subsequently, the proposed algorithms were applied to solve the coverage problem of directional heterogeneous sensor network. All the algorithms were implemented using MATLAB 2013b and executed on a computer with a Core i5 2.3 GHz Quad CPU, 4 GB RAM and Windows 7 64 bit operating system.

7.1. Performance Evaluation in Test Functions

7.1.1. Single-Objective Function

Three well-studied test functions in benchmarking optimization algorithms, namely Rastrigrin, Rosenbrock, and Griewank are used to compare the performances of the LACRO and the original CRO algorithm. In all the simulation, the termination criterion is set to 800 iterations. The range of values for F_b and b_d are [0.75, 0.95] [35] and [0.5, 0.8] [28], respectively. The value of m_1 and m_2 are 5 and 6. That is to say, the values of F_b and b_d are selected by the learning automata from the set {0.75 0.8 0.85 0.9 0.95} and set {0.5 0.55 0.60 0.65 0.7 0.75 0.8}, respectively. The success threshold parameter θ is 0.8, and the population size is 300 ($H = 23$, $t = 3$, $W = C_{H+t-1}^{t-1} = 300$). The parameters in the learning automata are $a = b = 0.01$ [8]. The parameter of the function dimension D is 30 and the details of the test functions are as follows:

$$(1) \text{ Rastrigrin } \min f_1(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) + 10 \quad x_i \in [-5, 5]$$

$$(2) \text{ Rosenbrock } \min f_2(x) = \sum_{i=1}^n 100(x_{i+1} - x_i)^2 + (1 - x_i)^2 \quad x_i \in [-100, 100]$$

$$(3) \text{ Griewank } \min f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad x_i \in [-600, 600]$$

Every test function has its own characteristics that make it suitable for evaluating the efficiency of the proposed algorithm. Rastrigrin and Griewank are multimodal functions with many local optima that increase in the exponential way with problem dimensions, while Rosenbrock is a unimodal function.

Table 1. Results on three functions based on 10 independent runs.

Function	LACRO	CRO
f_1	Mean 0.0531 SD 0.02257	Mean 0.0733 SD 0.0247
f_2	Mean 3.3884 SD 91.8525	Mean 28.4710 SD 0.7326
f_3	Mean 0.1344 SD 0.1446	Mean 0.2578 SD 0.15297

Table 1 demonstrates the global mean values and the standard deviation (SD) of the final solutions during 10 rounds of simulation. The results in Table 1 show that the proposed LACRO algorithm obtains the goal of the best solutions for the testing functions. The comparisons prove that the mechanism of the learning automata-based parameters self-adaptive indeed makes the LACRO achieve better performance than the original CRO algorithm. It can effectively avoid local optima and premature convergence, finding optimal solution accuracy in test multimodal functions. Figure 2a–c illustrate the evolution of optimal fitness for two algorithms. As can be seen from Figure 2a–c, the proposed algorithm performs better and converges more rapidly than the original CRO algorithm. The proposed algorithm LACRO successfully gets the local optimal solution (0, 0... 0) in the end. Therefore, the proposed LACRO performs better than CRO in terms of convergence speed and final solution.

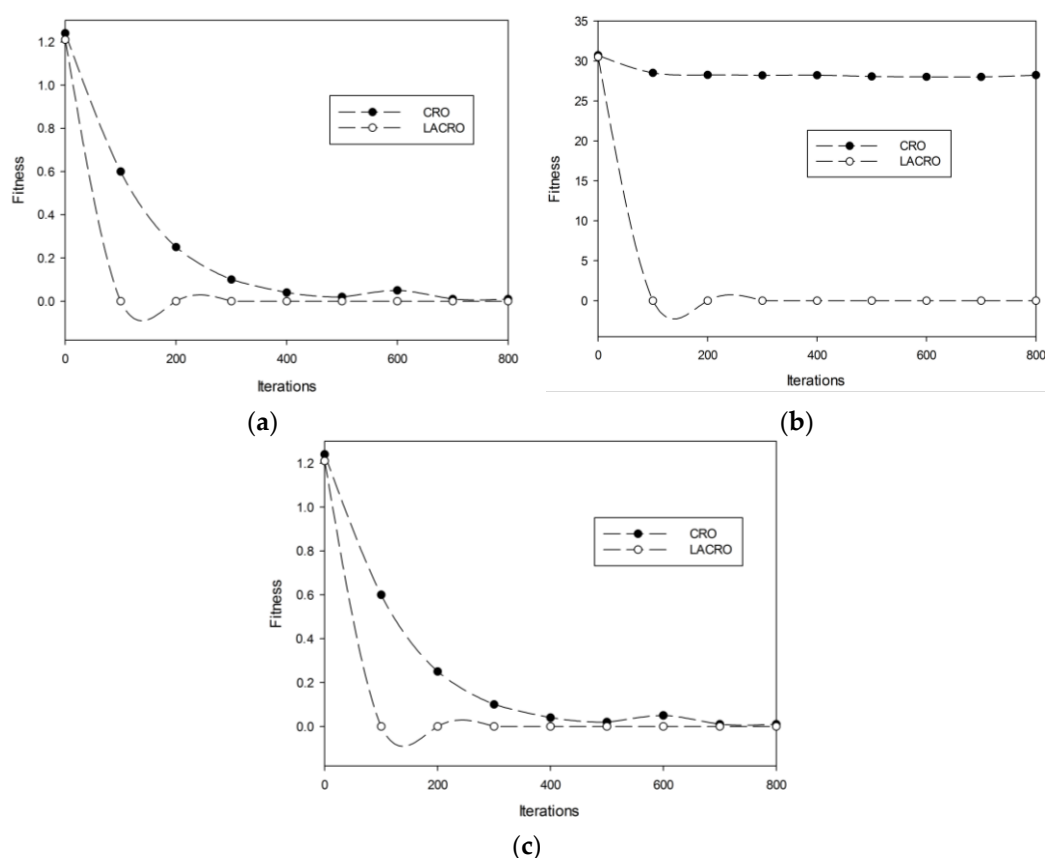


Figure 2. Comparison of LACRO and CRO. (a) f_1 Rastrigrin; (b) f_2 Rosenbrock; (c) f_3 Griewank.

7.1.2. Multi-Objective Function

To better verify the validity of the proposed fitness evaluation method and the performance of LACRO algorithm, we carried out a number of experiments on multi-objective function optimization. In this subsection, performance metrics for multi-objective optimization are firstly introduced. Then three test functions are presented. Finally, the results will be given at the end of this subsection.

(1) Performance metrics

In our experiments, the following performance index is used:

Set coverage(C-metric): The C-metric [36] evaluates the rate between dominance of a Pareto front over another Pareto front. Assuming that A and B are two approximated Pareto-optimal sets, the coverage of two sets $C(A, B)$ is defined as the percentage of the solutions in B that are dominated by at least one solution in A , *i.e.*:

$$C(A, B) = \frac{|A - \{x \in A \mid \exists y \in B : y \prec x\}|}{|A|}$$

The smallest $C(A, B)$ is, the better the A . It should be noted that $C(A, B)$ is not necessarily equal to $1 - C(B, A)$. $C(A, B) = 1$ means that all solutions in B are dominated by some solutions in A , while $C(A, B) = 0$ implies that no solution in B is dominated by a solution in A .

(2) Test functions

We use three widely used bio-objective test instances [37–39] to compare LACRO with the original CRO. All these test instances are minimization of the objectives. They are selected based on following properties: (i) the maximum value and minimum value of each sub problem in the test functions is easy to obtain; (ii) Exact shape and location of resulting PF for these problems are known. Table 2 presents the details of the test problems used in the paper. The experiment environment is the same as the performance simulation in single objective function section.

Table 2. Test problems used in this study.

Problem	Objective Functions	Domain
F1	$f_1 = x^2$ $f_2 = (x - 2)^2$	$[-5, 7]$
F2	$f_1(x, y) = (x^2 + y^2)^{\frac{1}{8}}$ $f_2(x, y) = [(x - 0.5)^2 + (y - 0.5)^2]^{\frac{1}{4}}$	$[-5, 10]$
F3	$f_1(x) = \begin{cases} -x & x \leq 1 \\ -2 + x & 1 < x \leq 3 \\ 4 - x & 3 < x \leq 4 \\ -4 + x & x > 4 \end{cases}$ $f_2(x) = (x - 5)^2$	$[-5, 10]$

(3) Results

The parameter setting of the algorithm is the same as the single-objective test section. In formula (A), each individual weight $\lambda_1, \lambda_2, \dots, \lambda_W$ (W is the population size) takes a value from the set of $\{0/300, 1/300, \dots, 300/300\}$. Table 3 describes of the C-metric values of the final approximations obtained by LACRO and original CRO. The values in the brackets are the standard deviation of the solutions. As can be seen from Table 3, LACRO performs better than CRO in terms of C-metric. Taking Function F1 as an example, on average, 97% of the final solutions generated by CRO are dominated by LACRO and 89% vice versa.

Table 3. Average value of C-metric based on 10 independent runs.

Function	C (LACRO, CRO)	C (CRO, LACRO)
F1	0.97 (0, 00005)	0.89 (0, 00042)
F2	0.59 (0, 00008)	0.5 (0, 00032)
F3	0.27 (0, 00018)	0.2 (0, 0004)

7.2. Application to the HDSN Coverage Problem

7.2.1. Experimental Setup

In this section, the performance of the proposed algorithm is analyzed and the parameters of the algorithm are the same as the previous section in performance evaluation in test functions. The

following network parameters are set: $m = n = 40$, $N = 300$, sensor parameters are presented in Table 4.

Table 4. Sensor specifications.

Type	Sensing Radius	Radius Communication	Angle of View	Quantity
1	10	20	$\pi/3$	150
2	15	30	$\pi/2$	150

In order to model the hotspots, we divide the monitor area into a number of squared subareas. For each subarea, we assign a number of targets to that subarea according to a bounded Pareto distribution. A subarea with a lot of targets can be viewed as a hot spot. Once the number of targets in a specific subarea is determined, they are randomly distributed within that subarea. The cumulative density function of the bounded Pareto distribution is:

$$F(k) = \frac{1 - (u/k)^\alpha}{1 - (u/v)^\alpha}$$

where $u < k < v$, $0 < \alpha < 2$ [40]. In this paper, $u = 3$, $v = 100$, and $\alpha = 1.1$, the number of targets and subareas are 150 and 36, respectively. Figure 3 illustrates the targets distribution of the clustered layout generated by the above-mentioned method. In our experiment, the parameter K , that is coverage requirement and connect requirement is in the hot area, which is set 2, otherwise, $K = 1$ in the non-hot area.

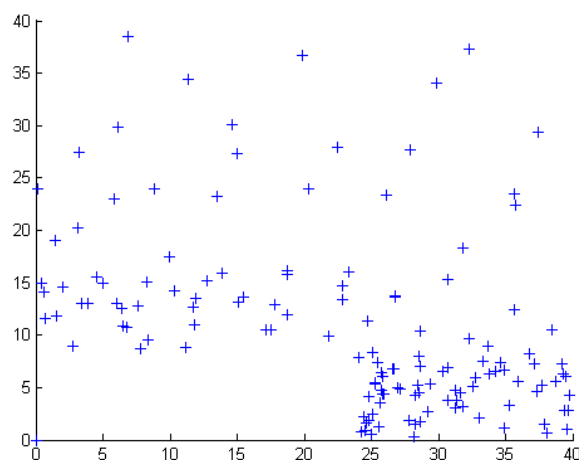


Figure 3. Clustered layout of targets (150 targets in a 40×40 area).

7.2.2. Experimental Results and Analysis

The results generated by LACRO with different generations are illustrated in Figure 4. As can be seen from the figure, with the increase in the iteration number, the solutions obtained by LACRO approximate optimal, which verify the effectiveness of the algorithm in solving multi-objective coverage optimization problems in HDSNs.

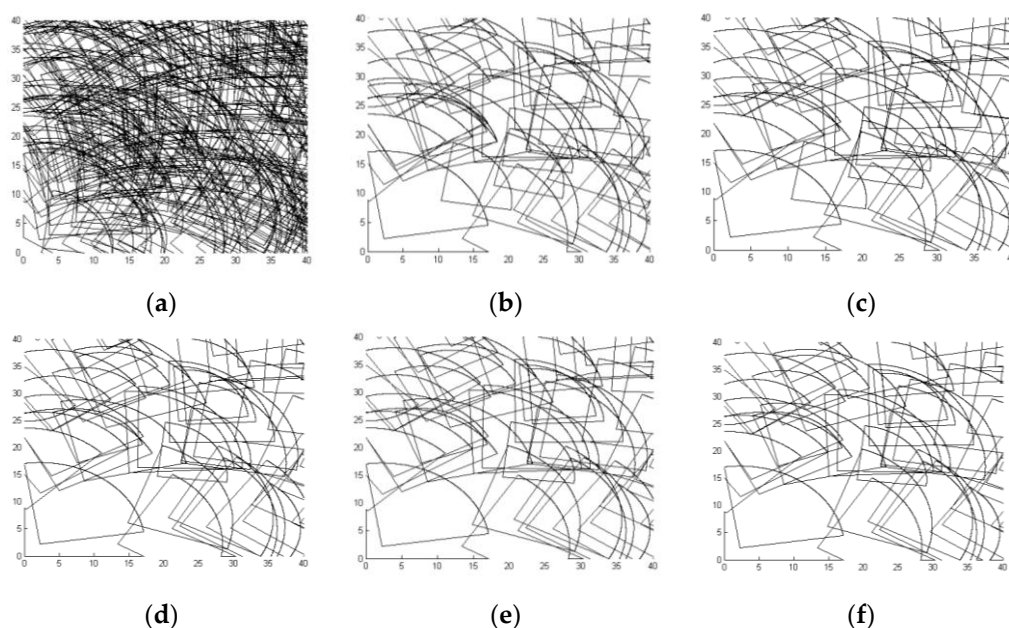


Figure 4. One of the non-dominated solutions. (a) 300 nodes, initial distribution; (b) the 100th generation, 66 nodes, 89.3% coverage rate, full connect value; (c) the 200th generation, 62 nodes, 90.9% coverage rate, full connect value; (d) the 400th generation, 58 nodes, 91.3% coverage rate, full connect value; (e) the 600th generation, 57 nodes, 92.9% coverage rate, full connect value; (f) the 800th generation, 53 nodes, 95.8% coverage rate, full connect value.

Experiment 1: Fitness comparison

This experiment studies the exploration/exploitation capabilities of the algorithms. Figure 5. shows the evolution of the average fitness and better fitness with the iteration number. As can be seen in the figure, the two curves are noticeably different and the LACRO curve converges more rapidly and effectively than the CRO curve, which is strong proof that LACRO has better performance than CRO.

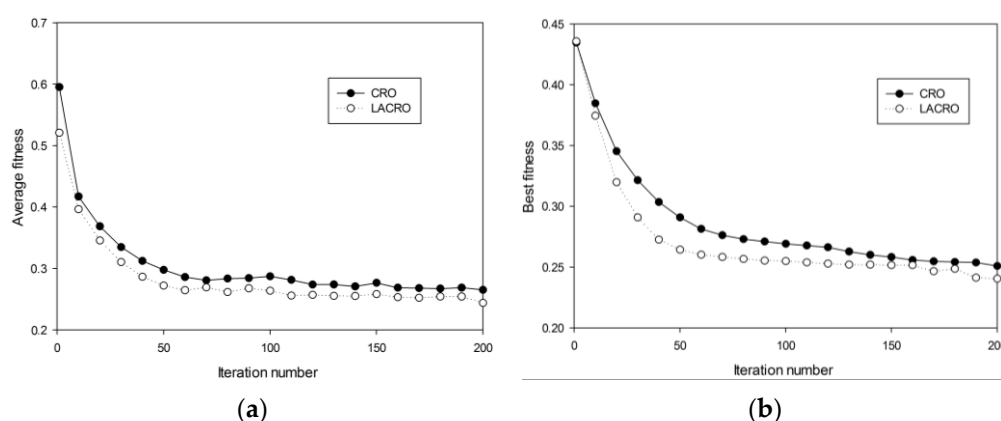


Figure 5. (a) Average fitness in various generations; (b) Best fitness in various generations.

Experiment 2: Coverage control performance comparison

In order to compare the performance of the two algorithms in the HDSN multi-objective optimization problem, several experiments are carried out and the results are presented in Figure 6. As can be seen from the figure, LACRO can converge more rapidly and obtains better solutions than the CRO algorithm in coverage rate, working node number and connect value. Due to the random nature of the CRO algorithm, it may “vibrate” in its optimization process as can be seen from

Figure 6, especially in Figure 6c. Overall, we can claim that LACRO outperforms CRO in these problems.

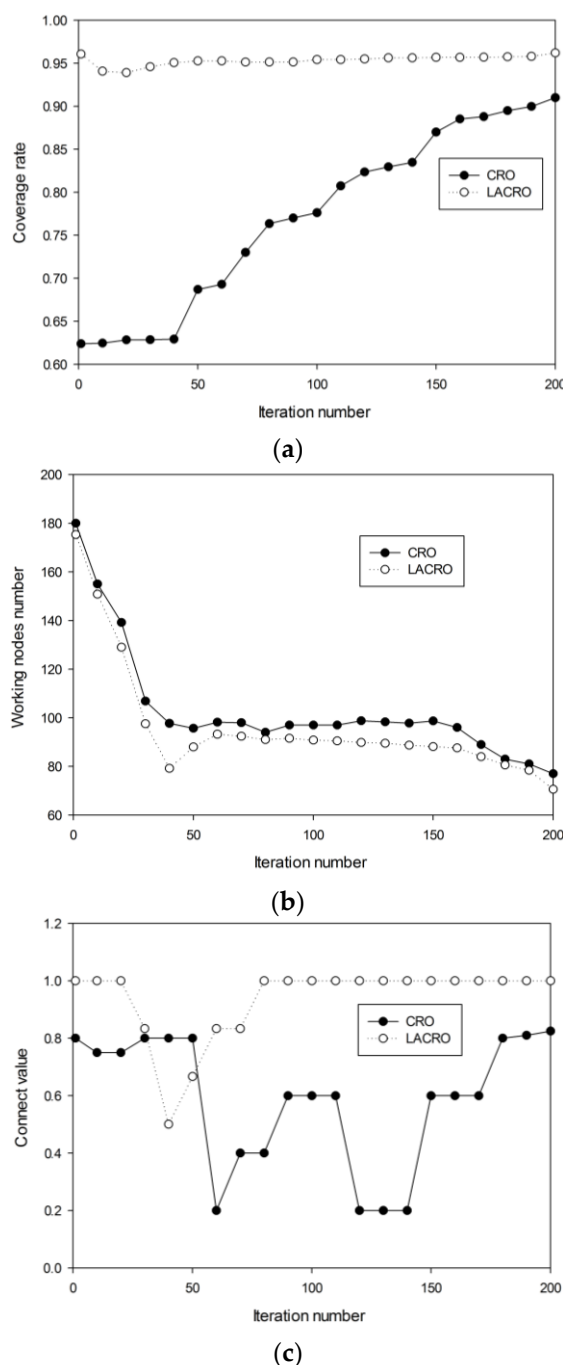


Figure 6. (a) Coverage rate f_1 in various generations; (b) Ratio of working nodes f_2 in various generations; (c) Connect value f_3 in various generations.

Experiment 3: Performance evaluation between LACRO and other algorithms

In order to compare the performance of different approaches, we run the pure random algorithm and k-random algorithm [41] where k is equal to 2 and 3, respectively. The results are illustrated in Figure 7. We can see clearly that LACRO needs the least amount of sensor nodes to ensure full connectivity value and the best coverage rate among the comparison algorithms. Thus it is fair to claim that LACRO has better optimization ability and scalability compared to other coverage control algorithms and it is much more effective for solving multi-objective coverage control problems with large dimension sizes.

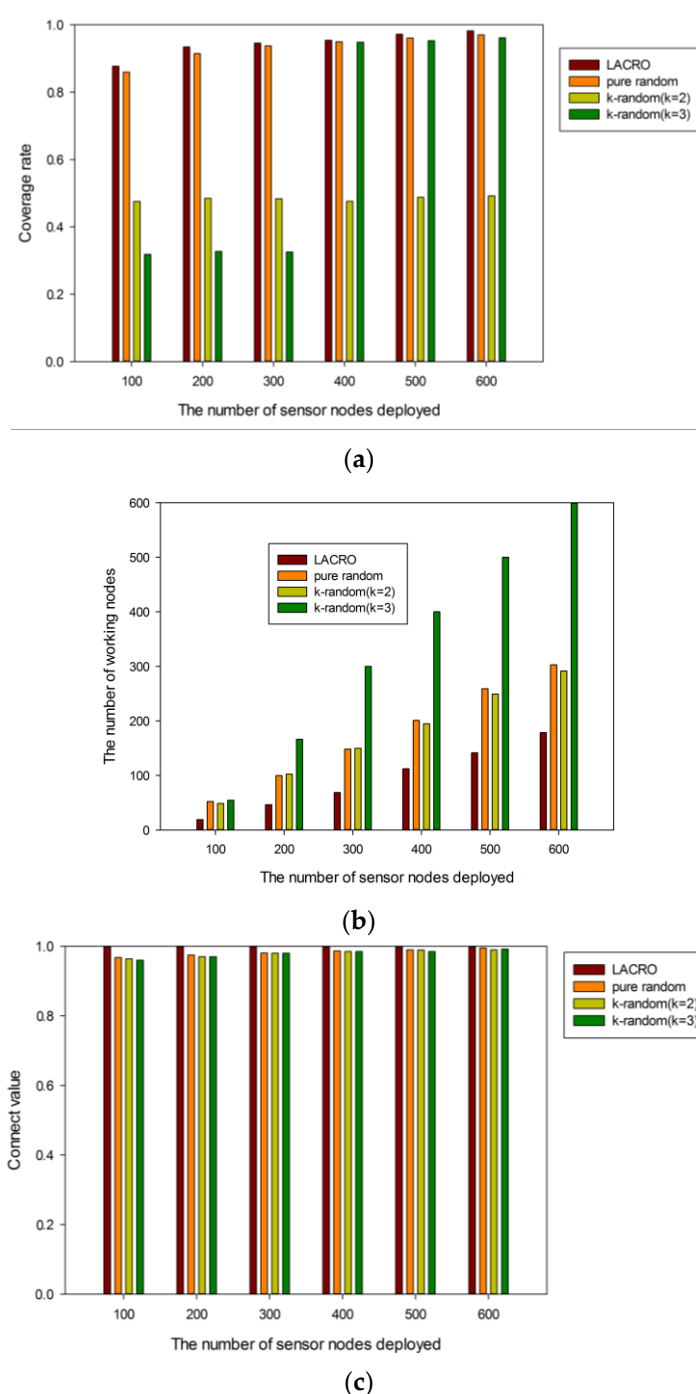


Figure 7. (a) Coverage rate f_1 vs. node density; (b) Ratio of working nodes f_2 vs. node density; (c) Connect value f_3 vs. node density.

8. Conclusions and Future Work

Featured by the geographical irregularity of the sensed events and heterogeneity of the sensor nodes, this paper presents a novel multi-objective optimization problem of HDSN which guarantees the coverage rate of the monitor area, the financial cost of deployment and the directional sensor network communication connectivity. Our proposal is a learning automata-based coral reef optimization algorithm (LACRO) which equips the CRO with the ability to create spur-in-time responses with better exploration/exploitation capabilities as a result. The CRO parameters are selected by the learning automata. The parameters of F_b and B_d are discretized within their permitted ranges, and a learning automaton with a finite action set is used for each parameter. An enhanced decomposition method based on the Tchebycheff Approach is proposed to decompose the

multi-objective problem into some single-objective problems. Experimental results indicate that the proposed algorithm and the improved Tchebycheff method are highly competitive with the CRO algorithm in test functions and multi-objective optimization problems in HDSNs. In the future, we will study energy efficient coverage optimization algorithms in mobile HDSN, which can adjust the working direction of the nodes and change their physical positions to improve the network coverage.

Acknowledgments: The authors wish to express their appreciation to Huang Minting for her language polishing work on the paper. The authors also appreciate the useful suggestions and opinions provided by three anonymous reviewers, which have helped to improve this paper. This research is funded partially by Chongqing Education Committee Cooperation Foundation, China (Grant No.KJ130716), Chongqing Science and Technology Commission, China (Grant No. CSTC2012jjA40037), the Scientific Research Foundation of Chongqing Technology and Business University (Grant No. 2012-56-03), China Scholarship Council (CSC).

Author Contributions: Ming Li conceived and designed the experiments; Ming Li. performed the experiments; Ming Li, Chunyan Miao and Cyril Leung. analyzed the data; Ming Li. wrote the paper.

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

References

1. Akyildiz, I.F.; Melodia, T.; Chowdhury, K.R. A survey on wireless multimedia sensor networks. *Comput. Netw.* **2007**, *51*, 921–960.
2. Aitsaadi, N.; Achir, A.; Boussetta, K.; Pujolle, G. Tabu Search WSN Deployment Method for Monitoring Geographically Irregular Distributed Events. *Sensors* **2009**, *9*, 1625–1643.
3. Iqbal, M.; Naeem, M.; Anpalagan, A.; Ahmed, A.; Azam, M. Wireless Sensor Network Optimization: Multi-Objective Paradigm. *Sensors* **2015**, *15*, 17572–17620.
4. Zhang, Q.; Li, H. MOEA/D: A multi-objective evolutionary algorithm based on decomposition. *IEEE Trans. Evolut. Comput.* **2007**, *6*, 712–731.
5. Konstantinidis, A.; Yang, K. Multi-objective energy-efficient dense deployment in Wireless Sensor Networks using a hybrid problem-specific MOEA/D. *Appl. Soft Comput.* **2012**, *12*, 1847–1864.
6. Lanza-Gutierrez, J.M.; Gomez-Pulido, J.A. Assuming multiobjective metaheuristics to solve a three-objective optimisation problem for Relay Node deployment in Wireless Sensor Networks. *Appl. Soft Comput.* **2015**, *30*, 675–687.
7. Rasul, E.; Yousefi, M.; Abdullah, A.H.; Darus, A.N. LAHS: A novel harmony search algorithm based on learning automata. *Commun. Nonlinear Sci. Numer. Simulat.* **2013**, *18*, 3481–3497.
8. Hashemi, A.B.; Meybod, M.R. A note on the learning automata based algorithms for adaptive parameter selection in PSO. *Appl. Soft Comput.* **2011**, *11*, 689–705.
9. Guvensan, M.A.; Yavu, A.G. On coverage issues in directional sensor networks: A survey. *Ad. Hoc. Netw.* **2011**, *9*, 1238–1255.
10. Mohamadia, H.; Salleh, S.; Razali, M.N. Heuristic methods to maximize network lifetime in directional sensor networks with adjustable sensing ranges. *J. Netw. Comput. Appl.* **2014**, *46*, 26–35.
11. Ai, J.; Abouzeid, A.A. Coverage by directional sensors in randomly deployed wireless sensor networks. *J. Comb. Optim.* **2006**, *11*, 21–41.
12. Cai, Y.; Lou, W.; Li, M.; Li, X.-Y. Energy efficient target-oriented scheduling in directional sensor networks. *IEEE Trans. Comput.* **2009**, *58*, 1259–1274.
13. Wen, J.; Fang, L.; Jiang, J.; Dou, W. Coverage Optimizing and Node Scheduling in Directional Wireless Sensor Networks. In Proceedings of the IEEE International Conferences on Wireless Communications, Networking and Mobile Computing (WiCom'08), Dalian, China, 12–14 October 2008; pp. 1–4.

14. Kim, Y.-H.; Han, Y.-H. Lifetime maximization considering target coverage and connectivity in directional image/video sensor networks. *J. Supercomput.* **2013**, *65*, 365–382.
15. Sung, T.-W.; Yang, C.-S. Distributed Voronoi-Based Self-Redeployment for Coverage Enhancement in a Mobile Directional Sensor Network. *Int. J. Distrib. Sens. Netw.* **2013**, 1–15.
16. Sung, T.-W.; Yang, C.-S. Voronoi-based coverage improvement approach for wireless directional sensor networks. *J. Netw. Comput. Appl.* **2014**, *39*, 202–213.
17. Mohamadi, H.; Ismail, A.; Salleh, S.I.; Nodehi, A. Learning automata-based algorithms for solving the target coverage problem in directional sensor networks. *Wirel. Pers. Commun.* **2013**, *73*, 1309–1330.
18. Ma, H.; Zhang, X.; Ming, A. A coverage-enhancing method for 3d directional sensor networks. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'09), Rio de Janeiro, Brazil, 19–25 April 2009; pp. 2791–2795.
19. Mohamadi, H.; Ismail, A.S.B.H.; Salleh, S. A learning automata-based algorithm for solving coverage problem in directional sensor networks. *Computing* **2013**, *95*, 1–24.
20. Wang, J.; Niu, C.; Shen, R. Priority-based target coverage in directional sensor networks using a genetic algorithm. *Comput. Math. Appl.* **2009**, *57*, 1915–1922.
21. Yang, H.; Li, D.; Chen, H. Coverage Quality Based Target-Oriented Scheduling in Directional Sensor Networks. In Proceedings of the International Conference on Communications, Beijing, China, 23–27 May 2010; pp. 1–5.
22. Mohamadi, H.; Ismail, A.; Salleh, A.S. A Learning Automata-Based Solution to the Priority-Based Target Coverage Problem in Directional Sensor Networks. *Wirel. Pers. Commun.* **2014**, *79*, 2323–2338.
23. Bian, F.; Kempe, D.; Govindan, R. Utility Based Sensor Selection. In Proceedings of the 5th International Conference on Information Processing in Sensor Networks, Nashville, TN, USA, 19–21 April 2006; pp. 11–18.
24. Salcedo-Sanz, S.; del Ser, J.; Landa-Torres, I.; Gil-López, S.; Portilla-Figueras, J.A. The Coral Reefs Optimization Algorithm: A Novel Metaheuristic for Efficiently Solving Optimization Problems. *Sci. World J.* **2014**, *2014*, 1–15.
25. Salcedo-Sanz, S.; García-Díaz, P.; Portilla-Figueras, J.A.; del Ser, J.; Gil-López, S. A Coral Reefs Optimization Algorithm for Optimal Mobile Network Deployment with Electromagnetic Pollution Control Criterion. *Appl. Soft Comput.* **2014**, *24*, 239–248.
26. Salcedo-Sanz, S.; Casanova-Mateo, C.; Pastor-Sánchez, A.; Sánchez-Girón, M. Daily Global Solar Radiation Prediction Based on a Hybrid Coral Reefs Optimization—Extreme Learning Machine Approach. *Solar Energy* **2014**, *105*, 91–98.
27. Salcedo-Sanz, S.; Pastor-Sánchez, A.; Prieto, L.; Blanco-Aguilera, A.; García-Herrera, R. Feature selection in wind speed prediction systems based on a hybrid coral reefs optimization—Extreme learning machine approach. *Energy Convers. Manag.* **2014**, *87*, 10–18.
28. Salcedo-Sanz, S.; Pastor-Sánchez, A.; del Ser, J.; Prieto, L.; Geem, Z.W. A Coral Reefs Optimization Algorithm with Harmony Search Operators for Accurate Wind Speed Prediction. *Renew. Energy* **2015**, *75*, 93–101.
29. Narendra, K.S.; Thathachar, M.A.L. Learning automata: a survey. *IEEE Trans. Syst. Man Cybern.* **1974**, *3*, 23–34.
30. Thathachar, M.A.L.; Sastry, P.S. A hierarchical system of learning automata that can learn the globally optimal path. *Inf. Sci.* **1997**, *42*, 743–766.
31. Beigy, H.; Meybodi, M.R. Cellular learning automata with multiple learning automata in each cell and its applications. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2010**, *40*, 54–66.

32. Konak, A.; Coit, D.W.; Smith, A.E. Multi-objective optimization using genetic algorithms. *Reliab. Eng. Syst. Saf.* **2006**, *91*, 992–1007.
33. Konstantinidis, A.; Yang, K. Multi-Objective *K*-Connected Deployment and Power Assignment in WSNs Using a Problem-Specific Constrained Evolutionary Algorithm Based on Decomposition. *Comput. Commun.* **2011**, *34*, 83–98.
34. Storn, R.; Price, K. Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359.
35. Salcedo-Sanz, S.; Gallo-Marazuela, D.; Pastor-Sánchez, A.; Carro-Calvo, L.; Portilla-Figueras, A.; Prieto, L. Offshore Wind Farm Design with the Coral Reefs Optimization Algorithm. *Renew. Energy* **2014**, *63*, 109–115.
36. Zitzler, E.; Thiele, L. Multiple objective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evolut. Comput.* **1999**, *3*, 257–271.
37. Schaffer, J.D. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In Proceedings of the 1st International Conference on Genetic Algorithms, Pittsburgh, PA, USA, 24–26 July 1985; pp. 93–100.
38. Joanna, L.; Eiben, A.E. A Multi-Sexual Genetic Algorithm for Multi-Objective Optimization. In Proceedings of the IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 13–16 April 1997; pp. 13–16.
39. Zitzler, E.; Deb, K.; Thiele, L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolut. Comput.* **2000**, *8*, 173–195.
40. Watts, D.J.; Strogatz, S.H. Collective dynamics of small-world networks. *Nature* **1998**, *393*, 440–442.
41. Liu, C.; Wu, K.; King, V. Randomized Coverage-Preserving Scheduling Schemes for Wireless Sensor Networks. *IEEE Trans. Parallel Distrib. Syst.* **2005**, *17*, 956–967.



© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons by Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).