Sensors **2014**, *14*, 20500-20518; doi:10.3390/s141120500

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Memetic Algorithm-Based Multi-Objective Coverage Optimization for Wireless Sensor Networks

Zhi Chen [1,2,3,*], Shuai Li [1] and Wenjing Yue [4]

[1] College of Computer, Nanjing University of Posts and Telecommunications, No.9, Wenyuan Road, Yadong new District, Nanjing 210023, China; E-Mail: acm@njupt.edu.cn

[2] Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, No.66, New Mofan Road, Gulou District, Nanjing 210003, China

[3] State Key Laboratory for Novel Software Technology, Nanjing University, No.163, Xianlin Road, Qixia District, Nanjing 210023, China

[4] Key Lab of Broadband Wireless Communication and Sensor Network Technology, Ministry of Education, No.66, New Mofan Road, Gulou District, Nanjing 210003, China; E-Mail: yuewj@njupt.edu.cn

* Author to whom correspondence should be addressed; E-Mail: chenz@njupt.edu.cn; Tel.: +86-25-8586-6422; Fax: +86-25-8586-6433.

External Editor: Leonhard M. Reindl

Received: 9 August 2014; in revised form: 8 October 2014 / Accepted: 18 October 2014 / Published: 30 October 2014

Abstract: Maintaining effective coverage and extending the network lifetime as much as possible has become one of the most critical issues in the coverage of WSNs. In this paper, we propose a multi-objective coverage optimization algorithm for WSNs, namely MOCADMA, which models the coverage control of WSNs as the multi-objective optimization problem. MOCADMA uses a memetic algorithm with a dynamic local search strategy to optimize the coverage of WSNs and achieve the objectives such as high network coverage, effective node utilization and more residual energy. In MOCADMA, the alternative solutions are represented as the chromosomes in matrix form, and the optimal solutions are selected through numerous iterations of the evolution process, including selection, crossover, mutation, local enhancement, and fitness evaluation. The experiment and evaluation results show MOCADMA can have good capabilities in maintaining the sensing coverage, achieve higher network coverage while improving the energy efficiency and effectively prolonging the network lifetime, and have a significant improvement over some existing algorithms.

Keywords: sensor networks; coverage algorithm; memetic algorithm; multi-objective optimization

1. Introduction

Wireless Sensor Networks (WSNs) are self-organized networks consisting of sensor nodes with the ability of sensing, processing and wireless communicating [1]. Coverage control is one of the most fundamental research issues in sensor networks, and studies how well a sensor network will monitor a field of interest with the proper node deployment [2,3]. Sensor nodes often have constrained resources and it is sometimes difficult to recharge their energy, and thus coverage sustainability in such sensor networks cannot be guaranteed. How to balance the network energy consumption in coverage control is an important issue, which can be modeled as a multi-objective optimization problem of prolonging the network lifetime and improving network coverage according to the characteristics of WSNs [4].

Multi-objective optimization problems involve two or more conflicting objectives and have not one optimal solution but many solutions which form the Pareto front representing a tradeoff of one objective against the other. In most applications, the goal of solving the multi-objective optimization problems is to compute an approximation of the Pareto front. Computational Intelligence (CI) and evolutionary algorithms provide adaptive, flexible and robust mechanisms that exhibit intelligent behavior to solve the multi-objective optimization problems of coverage control in complex and dynamic environments like WSNs [5]. Habib modeled the coverage problem with two sub-problems: floorplan and placement, and used the genetic algorithm (GA) to search the optimal coverage in WSNs [6]. Ozturk *et al.* applied artificial bee colony algorithm to the dynamic deployment of the sensor networks and obtained better deployments for WSNs than the particle swarm optimization algorithm [7,8]. The aforementioned articles emphasized to improve the coverage rate and did not specifically take into account network lifetime and energy balance in WSNs. Memetic algorithms are computational intelligence structures and a class of stochastic global search heuristics where evolutionary algorithms are combined with multiple and various local search heuristics in order to address such optimization problems as those in WSNs [9–11].

This paper presents a multi-objective coverage optimization algorithm based on memetic algorithm for WSNs, namely MOCOAMA, which considers the coverage degree, node utilization, node residual energy, and solves the 1-coverage multi-objective problem for WSNs [12–15], and finally gets the optimal deployment of network coverage.

The rest of the paper is organized into five sections: Section 2 briefly introduces related work. Section 3 discusses the multi-objective optimization coverage problem of WSNs. Section 4 presents the key schemes for the proposed coverage algorithm for WSNs. Section 5 describes the multi-objective optimization coverage algorithm based on memetic algorithms. The simulation experiments and evaluation are given in Section 6. Finally, the conclusions are offered in Section 7.

2. Related Work

Konstantinidis *et al.* proposed a memetic algorithm (MA)-based solution of energy-aware topology control (ToCMA) for WSNs using a combination of problem-specific light-weighted local searches and genetic algorithms [11], which can provide a better performance than the classical minimum spanning tree solutions. In ToCMA, the composing entities of a chromosome or genes are the power of sensor nodes; the fitness function of a solution is the sum of the power assigned to each gene, and repair and improvement methods are employed to refine solutions.

Ferentinos *et al.* used a memetic algorithm to dynamically optimize the design of WSNs and considered different constraints such as application-specific requirements, communication constraints and energy consumption [14]. Their work showed that the hybridization of the original GA with the local search operations presented in memetic algorithms brought some improvement on the performance of the design process.

Jiang *et al.* used memetic algorithms to implement energy-efficient coverage control in cluster-based WSNs (CoCMA) [15], which contains a memetic algorithm-based schedule for sensor nodes and a wake-up scheme. In CoCMA, the coverage solutions are represented by binary strings, and the status of a node is represented by an allele of a chromosome. CoCMA can prolong the network lifetime while maintaining coverage preservation of WSNs with no sensing error, and has a significantly improved performance compared with the LEACH [16], LEACH-Coverage-U [17], *etc.*

Ting *et al.* proposed a memetic algorithm to solve the set k-cover problem of WSNs, which has the effectiveness and efficiency of extending the network lifetime [18]. In MA, a chromosome represents the sequence in which all sensor nodes are collected to form covers, and the fitness value of a chromosome is the sum of coverage contributions of all sensor nodes and can enhance the differentiation of promising chromosomes.

Arivudainambi *et al.* proposed a knowledge added improved memetic algorithm (iMA) for target coverage in WSNs, which is concerned with exploiting all available knowledge and demonstrated the effectiveness in extending the lifetime of WSNs [19]. iMA encodes the candidate solution as the chromosome represented by a matrix in which row represents the sensor nodes, and column represents the targets.

In this paper, our proposed MOCOAMA uses the coverage optimization framework of CoCMA [15], MA [18] and, iMA [19], but it deals with the coverage problem of WSNs with the sensing error, divides the target area with the virtual cells as the basis of chromosome representation, and randomly and alternately selects some local search algorithms to achieve the dynamic local search.

3. Multi-Objective Coverage Optimization Problem

In this paper, we assume that sensor nodes are randomly and uniformly distributed over the target area, with the same physical structure, communication capacity, sensing range, initial limited energy and computing capacity. The sink node or the base station (sink) is assumed to have unlimited energy with plenty of power supplies. The sink or every sensor node has a unique identifier, and can get its own location information and communicate with its neighbor nodes usually using normal power or with each other through power control.

A target field D is a two-dimensional plane divided into $M \times N$ virtual cells. The points of interest are distributed in D and there exists at most one point of interest in one virtual cell. A set of sensor nodes T are deployed over D where $T = \{t_1, t_2, ..., t_n\}$, $t_i = (x_i, \mathrm{y}_i, \mathrm{r}_i)$, n is the number of sensor nodes, $i \in [1, n]$, (x_i, y_i) is the coordinate of the sensor node t_i, and r_i is the maximum ideal sensing radius of t_i. The coverage field of t_i is a circle area with (x_i, y_i) as the center and r_i as the radius. The Euclidean distance between t_i and the point of interest c whose coordinate is (x, y) in D is:

$$d(t_i, c) = \sqrt{(x_i - x)^2 + (y_i - y)^2} \tag{1}$$

If the set of virtual cells sensed by the sensor node t_i is s_i, the coverage area of all sensor nodes is:

$$S(\mathrm{T}) = |\bigcup_{i=1}^{n} \gamma_i s_i| \tag{2}$$

where γ_i is the decision variable to be determined by the coverage algorithm, $\gamma_i = 1$ if t_i is in the state of activation, and $\gamma_i = 0$ if t_i is in the state of inactivation.

The probability of sensing the point of interest c by the sensor node t_i in D is:

$$p(t_i, \mathrm{c}) = \begin{cases} 0 & r_i \le d(t_i, c) \\ e^{-\lambda \times \frac{d(\mathrm{t}_i, d) - \ r_i - r_e}{r_i - d(\mathrm{t}_i, d)}} & r_i - r_e < d(t_i, c) < r_i \\ 1 & r_i - r_e \ge d(t_i, c) \end{cases} \tag{3}$$

where r_e is the sensing error value of the sensor nodes, r_i has the same value for all sensor networks assumed in this paper and λ is the sensing attenuation coefficient.

As long as the point of interest c is effectively sensed by a sensor node, it can be said that c is covered by the network. If a set of points of interest (*i.e.*, $c_1, c_2, ..., c_k$, $k \le M \times N$) at the same time is sensed by t_i, we get:

$$|s_i| = k(1 - \prod_{j=1}^{k} (1 - p(t_i, c_j))) \tag{4}$$

We define sensing coverage degree or coverage degree as the percentage of the field area covered by sensor nodes in the monitored field, which reflects the actual network sensing ability of monitoring a given field of interest or the points of interest. According to Equation (2), the coverage degree of sensor networks deployed in D is:

$$\varpi(\mathrm{T}) = \frac{S(T)}{M \times N} = \frac{|\bigcup_{i=1}^{n} \gamma_i s_i|}{M \times N} \tag{5}$$

According to Equation (4), we can get:

$$\varpi(\mathrm{T}) \le \frac{\sum_{i=1}^{n} \gamma_i |\mathrm{s}_i|}{M \times N} \le \frac{\sum_{i=1}^{n} \gamma_i k(1 - \prod_{j=1}^{k} (1 - p(t_i, c_j)))}{M \times N} \tag{6}$$

If $k = M \times N$, we can get:

$$\varpi(\mathrm{T}) \le \sum_{i=1}^{n} \gamma_i (1 - \prod_{j=1}^{M \times N} (1 - p(t_i, c_j))) \tag{7}$$

We define the objective optimization model of the network coverage as:

$$\begin{cases} \min \sum_{i=1}^{n} \gamma_i \\ \max \varpi(\mathrm{T}) \end{cases} \tag{8}$$

In WSNs, due to the limited energy of sensor nodes, reducing the energy consumption of some nodes and balancing the energy consumption of the entire network can effectively save the limited energy of the network and extend the network lifetime. Under the premise of guaranteeing some certain coverage degree, we may reduce the number of sensor nodes in working and deactivate the redundant nodes to reduce the energy consumption as much as possible to increase the network lifetime.

We define node utilization as the ratio of the number of cells having active sensor nodes to the total cell number in the target area given by:

$$U(T) = \frac{\| T \|}{M \times N} \tag{9}$$

where $\| \mathrm{T} \|$ is the number of cells having active sensor nodes and sensor nodes are deployed in the target field D which is divided into $M \times N$ virtual cells.

To extend the lifetime of sensor networks, we also consider the residual energy of WSNs in the target field D. If E_{i} is the residual energy of the sensor node t_i, the energy distribution of the network is expressed as:

$$E(T) = \frac{\frac{1}{n}\sum_{i=1}^{n}(\mathrm{E}_i - \frac{1}{n}\sum_{i=1}^{n}\mathrm{E}_i)^2}{(\frac{1}{n}\sum_{i=1}^{n}\mathrm{E}_i)^2} = \frac{n\sum_{i=1}^{n}(\mathrm{E}_i - \frac{1}{n}\sum_{i=1}^{n}\mathrm{E}_i)^2}{(\sum_{i=1}^{n}\mathrm{E}_i)^2} \tag{10}$$

where $E(T)$ reflects the degree of energy difference in the network; if its value is smaller, the energy difference between sensor nodes may be smaller, and the residual energy of the sensor nodes may be higher.

To increase the network lifetime, we may not only use as few nodes as possible to reduce energy consumption, but also make sure to select the nodes with balanced energy consumption and a higher residual energy. Therefore, we define the objective optimization model of the energy consumption as:

$$\begin{cases} \min \alpha \times U(T) + \beta \times E(T) \\ \max \varpi(\mathrm{T}) \end{cases} \tag{11}$$

where α is the node utilization weighting coefficient, β is the energy balance weighting coefficient and $\alpha + \beta = 1$.

According to Equations (8) and (11), in MOCOAMA, we pay attention to Pareto optimal solutions of the multi-objective coverage optimization problem of WSNs formulated as:

$$\begin{cases} \min \sum_{i=1}^{n} \gamma_i \\ \max \varpi(\mathrm{T}) \\ \min \alpha \times U(T) + \beta \times E(T) \end{cases} \tag{12}$$
$$s.t.\ \gamma_i = 0 \ or \ 1, \mathrm{i} \in [1, \mathrm{n}]$$

4. Multi-Objective Coverage Optimization Designs

A memetic algorithm uses the notion of meme(s) as units of information encoded in computational representations [20], which is a combination of global search and local search, and the objects of genetic manipulation are not any individuals in the population space, but some locally-optimal representatives elected by local search and from the local area. Using the inherent parallelism of memetic algorithm, we can greatly accelerate the rate of convergence of the multi-objective coverage optimization algorithm for WSNs.

4.1. Encoding Rule

Encoding is a mapping from the problem space to the solution space, but memetic algorithm cannot directly process the solution data of the solution space, so before searching, the variables of the solution space must be mapped into the data structures of evolutionary space, namely chromosomes. In MOCOAMA, when the sink divides the target area into $M \times N$ cells, each chromosome represents a deployment solution of WSNs with the form of an $M \times N$ matrix given in Equation (13):

$$\aleph = \begin{bmatrix} \mathrm{X}_{1,1} & \mathrm{X}_{1,2} & \cdots & \mathrm{X}_{1,N} \\ \mathrm{X}_{2,1} & \mathrm{X}_{2,2} & \cdots & \mathrm{X}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{X}_{M,1} & \mathrm{X}_{M,2} & \cdots & \mathrm{X}_{M,N} \end{bmatrix} \tag{13}$$

In each chromosome $\aleph$, allele $X_{i,j}$ denotes one of the n sensor nodes covering the virtual cell of (i,j) where $X_{i,j} \in [0,n]$; if $X_{i,j} = 0$, the virtual cell of (i,j) is not covered by any sensor node. The sensor nodes in one cell have one of three states: "sleeping" state, "detecting" state and "working" state; the "working" state is the state of activation, and other two states are the states of inactivation. Sensor nodes covering one cell update their remaining energy in real time, and the sink chooses one sensor node as the working node which has shorter distance from the center of the virtual cell or has more residual energy and has a priority right to being the working node. After collecting the information on locations and residual energy of all nodes covering each cell, the sink computes the probability of being a working node for each sensor node. The node with greater probability will be elected as the working node covering one cell, or a node will be randomly selected as the working node from those nodes with the same probability, and other nodes will turn into being the sleeping state from the detecting state. In order to balance the energy consumption of each node, re-election of the working node covering one cell will be requested under the control of the sink when the remaining energy of the working node is less than the average energy of sensor nodes covering one cell.

In MOCOAMA, only one of the sensor nodes is elected in its virtual cell; one virtual cell is covered at least one sensor node at the different time unless it cannot be covered by any sensor node. Therefore, from each chromosome, given the WSNs of $T = \{t_1, t_2, ..., t_n\}$, we can get some sets of sensor nodes in working state and one set of sensor nodes can be described as:

$$\Psi = \bigcup_{i=1,j=1}^{i=M,j=N} \{X_{i,j}\} = \bigcup_{i=1}^{n} \gamma_i t_i \tag{14}$$

where $\gamma_i = 1$ if the sensor node i is in working state and $\gamma_i = 0$ if otherwise.

According to Equations (13) and (14), we can get one of the decision solutions of deploying WSNs given by:

$$\Gamma = \bigcup_{i=1}^{n} \gamma_i \tag{15}$$

4.2. Fitness Function

In MOCOAMA, the fitness of each chromosome may be evaluated according to the values of the multiple optimization sub-goals. We apply Pareto ranking [21] as the fitness value evaluation scheme in searching for a set of Pareto-optimal solutions of multi-objective coverage optimization for WSNs. The allocation order of each non-dominated individual in current population is 1; the allocation order of any other individual is the number of dominant individuals plus 1; the formula is as follows:

$$R(\aleph_i) = |\{\aleph_j \mid \aleph_j \in P, \aleph_j \succ \aleph_i, \forall \aleph_i \in P\}| \tag{16}$$

where $\aleph_i$ is any individual of the population P, $\succ$ means the dominance relationship between two individuals, $R(\aleph_i)$ means the number of individuals which dominate $\aleph_i$ in population P. The allocation order of $\aleph_i$ is $R(\aleph_i)+1$. If the individual $\aleph_i$ is closer to the optimal solution, the order number of $\aleph_i$ is smaller. When $\aleph_i$ is a non-dominated solution, the order number of $\aleph_i$ is 1. According to Equations (12) and (16), the fitness function in MOCOAMA is:

$$f(\aleph_i) = (\delta_1(\alpha \times U(T) + \beta \times E(T)) - \delta_2 \varpi(\mathrm{T}))^{-(R(\aleph_i)+1) \times \sum_{i=1}^{n} \gamma_i} \tag{17}$$

4.3. Local Search Strategy

The local search strategy of memetic algorithms is a process of screening the excellent individuals in the local area. The combination of global search algorithms and local search algorithms often shows good convergence and strong global search capability in solving the multi-objective optimization problem, but there are no uniform standards and guides in the choice of local search strategies, as well as the position, mild and frequency, so we need to discover the local search strategy which is suitable for the unresolved problem. Synthesizing the advantages of a variety of local search algorithms can make the process of the local search converge faster and have higher solution quality by designing specific local search access policies.

In order to achieve dynamic local search, we propose some self-adaption scheduling rules of the local algorithm. In every iteration of MOCOAMA, some algorithms from the pool of local search algorithms are randomly and alternately selected to make local search and get diverse solutions. Then the sink searches m searched nearest points from the initial point of the local search and the situation of solution fitness improvement corresponding to the local search algorithms used in the m points is evaluated. The one of the local search algorithms having the most obvious improvement on fitness is eventually used in the multi-objective optimization. The proposed rules learn from the search history information to regulate the local search strategy with adaptive, self-learning features.

The pool of local search algorithms is a collection of many local search algorithms, including the tabu search algorithm [22], hill climbing search algorithm [23] and adaptive directional local search strategy [24]. Tabu search is an optimization algorithm for simulating human intelligence, which mimics the human memory function, saves the searched optimal solution to the tabu table in the process of solving, and marks the solutions to avoid repeating the same search in order to gain a broad search range [22]. The hill-climbing algorithm is a heuristic search algorithm based on greedy search strategy, which selects a random solution as the current solution in the solution space and compares with the solutions in the neighborhood scope one by one until you find a local optimal solution [23]. The adaptive directional local search strategy dynamically adjusts the neighborhood radius and/or local search probability, depending on the relative local and global effectiveness of evolutionary operators and the local search operator [24].

5. Memetic Algorithm Based Multi-Objective Coverage Optimization

In this paper, MOCADMA uses the dynamic local search strategy and a local algorithm scheduling mechanism to adapt to the selection of local search algorithms; and increases the optimal storage strategy to accelerate the convergence speed of the algorithm. The pseudo code of MOCADMA is described as follows:

Pseudo code of MOCADMA

```
Given information: the target field D, the points of interest $\{c_1, c_2, ..., c_k\}$ ,
                   the sink node (Sink), the set of sensor nodes $T = \{t_1, t_2, ..., t_n\}$,
                   the candidate population (CP), the optimal population (P)
                   the candidate population size (pts), the optimal population size (pos),
                   the crossover probability (pcross), the mutation probability (pmutation)
                   maximum iterations (mi), parameters used in the fitness function
step(A): Deploying T over D
    for $i \leftarrow 1$ to n do
        $Sink \leftarrow (location_i, r_i, re_i, residual\_energy_i)$
    endfor
step(B): $D \rightarrow M \times N$ virtual cells
step(C): for $i \leftarrow 1$ to n do
        for $j \leftarrow 1$ to k do
            Calculate $p(t_i, c_j)$
        endfor
    endfor
    for $i \leftarrow 1$ to n do
        calculate $|s_i|$
    endfor
    repeat
        for $i \leftarrow 1$ to M do
            for $j \leftarrow 1$ to N do
                $X_{i,j} \leftarrow t = RouletteWheelSelect(T, i, j)$
            endfor
        endfor
        $\aleph \leftarrow EncodeChromosome(Matrix_{i=1,j=1}^{i=M,j=N}(X_{i,j}))$
        if $(\aleph \notin CP)$ then
            $CP \leftarrow \aleph$
        endif
    until the number of individuals in CP reaches pts
```

```
        for $i \leftarrow 1$ to pts do
            Calculate $f(\aleph_i)$
        endfor
        repeat
            Select the Pareto-optimal individual in CP and add it into P
        until the number of individuals in P reaches pos
repeat
step(D): for $i \leftarrow 1$ to NumofCrossover do
            $ChromoA \leftarrow Select(P)$
            $ChromoB \leftarrow Select(P)$
            $OP \leftarrow OP + Crossover(A, B, pcross)$
        endfor
step(E): for $i \leftarrow 1$ to NumofMutation do
            $ChromoA \leftarrow Select(OP)$
            $OP \leftarrow OP + Mutate(A, pmutation)$
        endfor
step(F): //randomly and alternately select some local search algorithms to make local search
        $P_1 \leftarrow SelectTabuSearch(P \cup OP)$              // example1
        $P_2 \leftarrow HillClimbingSearch(P \cup OP)$            // example2
        $P_3 \leftarrow AdaptiveDirectionalSearch(P \cup OP)$     // example3
        $CP \leftarrow Evaluate(P_1, P_2, P_3)$                   // select the optimal candidate solution
step(G): repeat
            Select the Pareto-optimal individual in CP and add it into P
        until the number of individuals in P reaches pos
until the number of iterations reaches mi
step(H): Get current optimal deployment solution from P
step(I): Execute current optimal deployment solution over T
```

MOCADMA includes iterated evolutionary operations of memetic algorithm and dynamic local search strategy, which is described in detail as follows:

(A). The Sink Collects the Information of Sensor Nodes

After initial uniform deployment, every sensor node is in the "detecting" state and sends its location information (*#location*), maximum ideal sensing radius (*#r*), sensing error value (*#re*) and energy information (*#residual_Energy*) to the sink by flooding communication.

(B). The Sink Divides the Network

The sink divides the monitored target field into $M \times N$ virtual cells (or units) according to the locations of the sensor nodes; each sensor node belongs to only one cell but can cover numerous cells; one cell may have many sensor nodes.

(C). The Sink Generates the Initial Population, Calculates the Fitness Values and Saves the Optimal Solutions

The sink calculates the probability of sensing points of interest by sensor nodes according to Equation (3) and the numbers of virtual cells sensed by sensor nodes according to Equation (4) using the information (*#location*, *#r*, *#re*) of every sensor node, repeatedly selects one node in every cell as the working node by roulette wheel selection that takes the number of the sensed virtual cells as the weight of every node, and generates the deployment solutions which form the initial candidate population CP of chromosomes. Then the sink gets the decision variables of sensor nodes from the chromosomes and calculates the fitness value of individual chromosome $\aleph_i$ in the initial population according to Equation (17) using the information (*#residual_energy*) of sensor nodes, selects the chromosomes with higher fitness value as the local Pareto-optimal solutions, and then saves these solutions in the current optimal population P. For the local optimal solution, if the population P does not contain $\aleph_i$ and $\aleph_i$ is not dominated by the existed solutions in P, the sink adds $\aleph_i$ to P. At the same time, the Euclidean distance of the local best individual $\aleph_i$ to other individuals in the external population space is calculated, and the closer individual is substituted for $\aleph_i$.

(D). Crossover

The sink takes multi-point crossover through exchanging the statuses of nodes in different local optimal chromosomes represented as the matrices. Two individual chromosomes in the current optimal population P are randomly and continuously selected and exchange one random row with the given crossover probability; if this recombination generates two new individuals, they will be added into the offspring population OP.

(E). Mutation

To maintain the diversity of solutions, the sink generates a variety of solutions through randomly changing the status of the nodes in the virtual cells. The mutation operator is applied to all individual chromosomes in OP, and every allele in one chromosome is modified with the given mutation probability; if this recombination generates a new individual, it will be added into OP.

(F). Dynamic Local Search

Some algorithm from the pool of local search algorithms, including the tabu search algorithm [22], hill climbing search algorithm [23] and adaptive directional local search strategy [24] is randomly and alternately selected to make local search in $P \cup OP$, and every local search algorithm generates its local population; then the sink compares the average fitness values of the same number of sample individuals around the initial point of the local search in those local populations, gets the local search algorithm that has the highest average fitness value, and selects its local population as the current candidate population CP.

(G). The Sink Calculates the Fitness Values and Saves the Optimal Solutions

The sink calculates the fitness value of individual chromosome $\aleph_i$ in the current candidate population CP, selects the local Pareto-optimal solutions, and then saves these solutions in P.

(H). The Sink Determines the Current Optimal Deployment Solution

When the number of iterations has reached the predetermined maximum threshold, the sink finishes determining the current optimal deployment solution having the highest fitness value; otherwise, MOCADMA continues to run from *(D)*.

(I).The Sink Broadcasts the Deployment Solution

The sink broadcasts the optimal network deployment solution to each sensor node, and informs each node in its cell responsible as the working node or the non-working node. The working nodes are in the "working" state covering the target area with the optimal deployment solution, and the non-working nodes are alternate in the "sleeping" or "detecting" state in a period of time. The sink monitors the network and evaluates the energy consuming of every sensor node. When the next round threshold time that may be decided by the routing protocol has arrived, or the remaining energy of one working node is less than the average energy in one cell, another new round comes and MOCADMA continues to run from *(D)*.

6. Experiments and Evaluation

We use Matlab to perform the simulation experiments of MOCADMA in which the sink is placed at the center of the target area. In fact, the sink may be deployed at any place in the area, but the energy consumption will be distributed over the field evenly if the position of the sink approximates the geometric center of the target area [15]. An energy consumption model [15,16,25] is used in the simulation experiments. In the energy consumption mode, the transmission energy E_{TX} for transmitting K bits of information between two sensor nodes is calculated by Equation (18); the consumed energy E_{RX} for receiving K bits of information by one sensor node is calculated by Equation (19); when the clustered routing algorithm is applied to the WSNs, the energy consumption of one cluster head [16] is calculated by Equation (20):

$$E_{TX} = K \times E_{elec} + \varepsilon_{fs} \times K \times \mathrm{d}^2 \tag{18}$$

$$E_{RX} = K \times E_{elec} \tag{19}$$

$$E_{CH} = (\frac{n}{C} - 1) \times K \times E_{elec} + K \times E_{elec} + \frac{n}{C} \times K \times E_{DA} + \varepsilon_{amp} \times K \times \mathrm{d}_{toSink}^4 \tag{20}$$

where n is the number of sensor nodes, C is the number of clusters, d_{toSink} is the distance between the cluster head and the sink, E_{elec} is the energy consumed by the electronics in the transmitter or receiver, ε_{fs} is the energy consumption of the signal power amplifier per square meter, E_{DA} is the energy consumption of processing one-unit bit data, and ε_{amp} is the energy consumption of transmitting one-unit bit data to the sink node.

To evaluate the performances of MOCADMA, simulation results are compared to those of CoCMA [15], MA [18] and iMA [19] using the same parameters shown in Table 1, and all the results are from the experiments repeated 30 times.

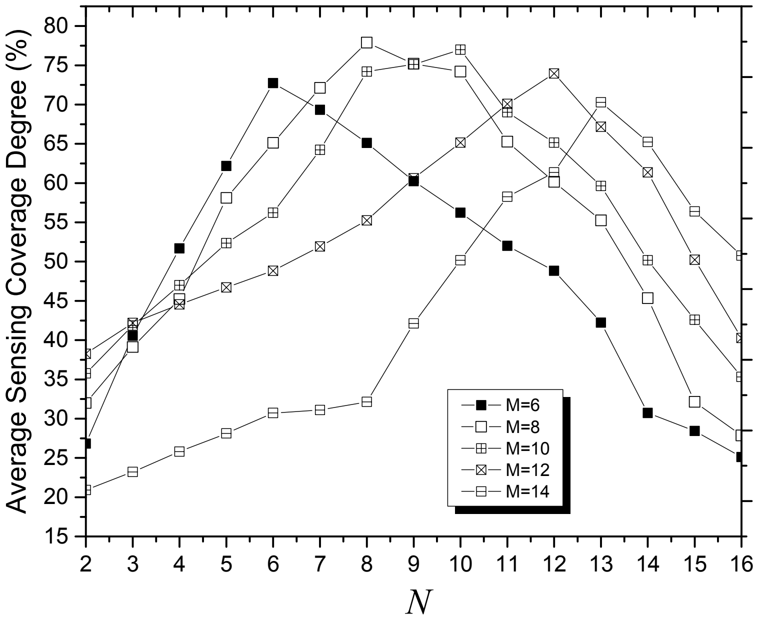
For MOCADMA, the $M \times N$ virtual cells dividing the target area at the initial stage bridge the problem space of WSNs and the solution space of chromosomes. In the simulation experiments, we consider 400 sensor nodes with no sensing error ($\mathrm{r}_e = 0$) and 64 points of interest randomly and uniformly deployed in a 100 m × 100 m target area. Figure 1 shows the average sensing coverage degree (SCD) *versus* different dividing of virtual cells and indicates that the average SCD has the maximum value at $N = 6$ when $M = 6$, $N = 8$ when $M = 8$, $N = 12$ when $M = 12$, and $N = 13$ when

$M = 14$. As Figure 2 shows, we can get the best dividing of virtual cells having maximum average SCD if $M \approx N$.

Table 1. Parameters used in simulations

Parameter	Value
Network size: n	≤ 400
Number of points of interest	64
Crossover probability: pcross	0.6
Mutation probability: pmutation	0.1
Maximum iterations	≤ 8000
Communication radius	20 m~80 m
Packet length: K	300 bit
Number of clusters: C	≤ 40
Node initial energy: E_0	10 J
Maximum ideal sensing radius: r_i	15 m
Sensing error value: r_e	≤1 m
λ	1
E_{elec}	50 nJ/bit
ε_{fs}	100 pJ/bit/m^2
E_{DA}	5 nJ/bit
ε_{amp}	0.0013 pJ/bit/m^4
δ_1	1
δ_2	1

Figure 1. Average SCD *versus* different dividing of virtual cells.



In the fitness function of MOCADMA, α and β ($\beta = 1 - \alpha$) are the node utilization weighting coefficient and the energy balance weighting coefficient, respectively. In the next simulation experiments, we assume that $N = 8$, $M = 8$, or $N = 10$, $M = 10$, or $N = 13$, $M = 14$. Figures 3 and 4 depict the network lifetime (rounds) *versus* different values of α when $r_e = 0$ and $r_e = 1$, respectively, and we can see that the network has a maximum lifetime (rounds) when $\alpha \approx 0.4$ ($\beta \approx 0.6$).

Figure 2. The best dividing of virtual cells having maximum average SCD.

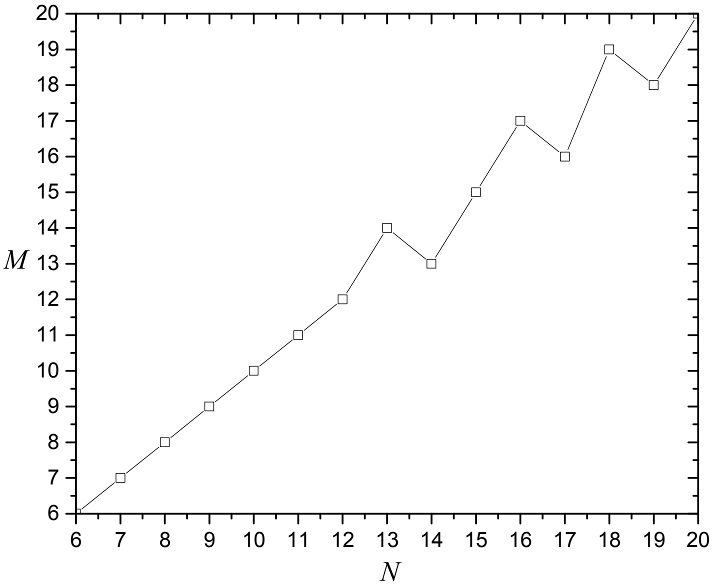
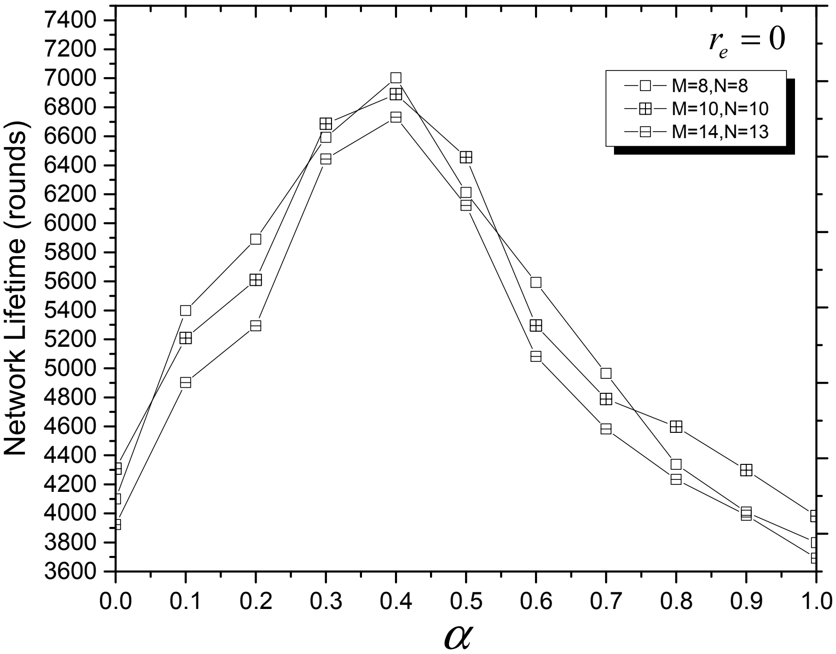


Figure 3. Network lifetime *versus* different values of α with $r_e = 0$.



In a hostile environment, the sensing error often exists in the applications of WSNs, so we consider it (r_e) in computing the probability of sensing the point of interest by one sensor node eventually used in the fitness function of MOCADMA. Figure 5 shows the average SCD *versus* different values of r_e and indicates that the average SCD decreases with the increase of r_e. It is clearly seen that the more uncertainty from the sensing error causes more difficulties for network coverage.

Figure 4. Network lifetime *versus* different values of α with $r_e = 1$.

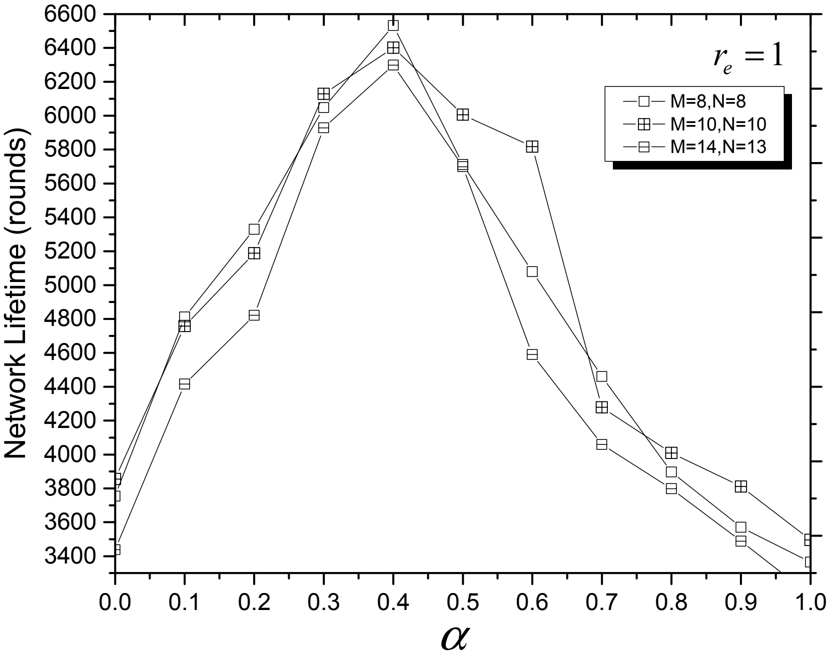


Figure 5. Average SCD *versus* different values of r_e.

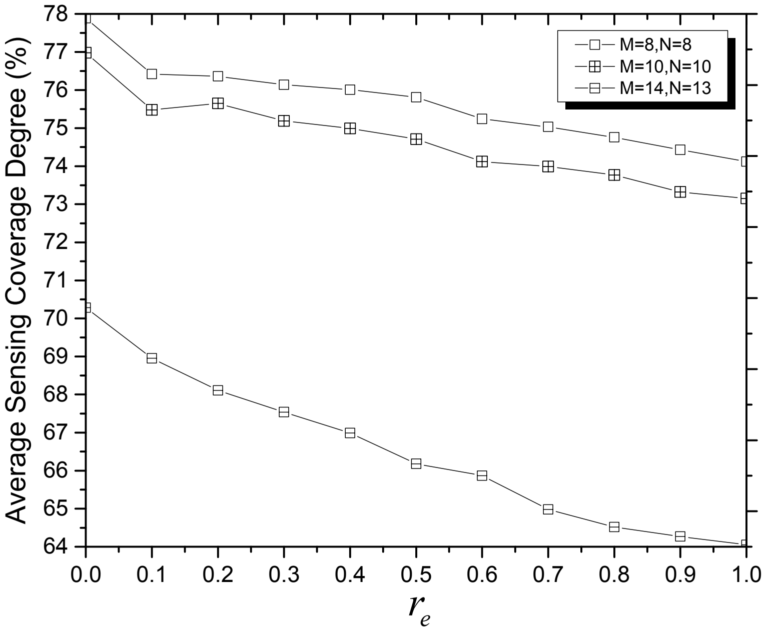
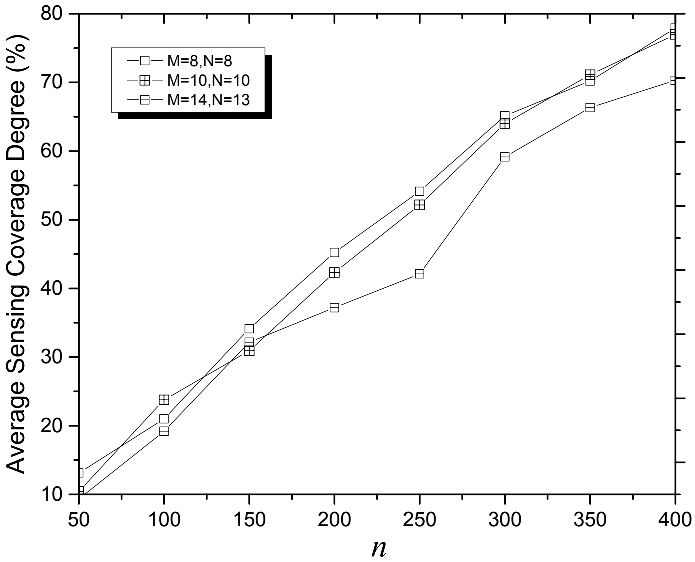


Figure 6 presents the average SCD *versus* network size with the fixed number of points of interest where $\alpha = 0.4$, $r_e = 0$. As Figure 6 shows, when 64 points of interest randomly and uniformly are deployed in the 100 m × 100 m target area, and the number of deployed sensor nodes varies from 50 to 400, the average SCD increases with the increase of network size (n).

Figure 6. Average SCD *versus* network size with the fixed number of points of interest.



In the next simulations, we verify the feasibility of MOCADMA in WSNs and consider 400 sensor nodes and 64 points of interest randomly and uniformly deployed in the 100 m × 100 m target area. To evaluate the performance of the MOCADMA, an LEACH-based clustered routing protocol [26] is applied to the WSNs, and CoCMA [15], MA [18] and iMA [19] are re-implemented. In addition, the design of CoCMA, MA and iMA didn't allow the sensing error, so we only consider the WSNs with no sensing error ($\mathrm{r}_e = 0$) in the simulation experiments. Figure 7 depicts the SCD *versus* network lifetime in rounds ($\alpha = 0.4$ in MOCADMA). We note that all the algorithms have good capabilities in maintaining the sensing coverage, but MOCADMA has maintained the sensing coverage degree at 100% until the 3800th additional round which is a significant improvement over other algorithms. As Figure 7 shows, the network lifetime is prolonged to 7000 more rounds by the MOCADMA, which lasts about 800 rounds longer compared to CoCMA and iMA.

Figure 7. SCD *versus* network lifetime in rounds.

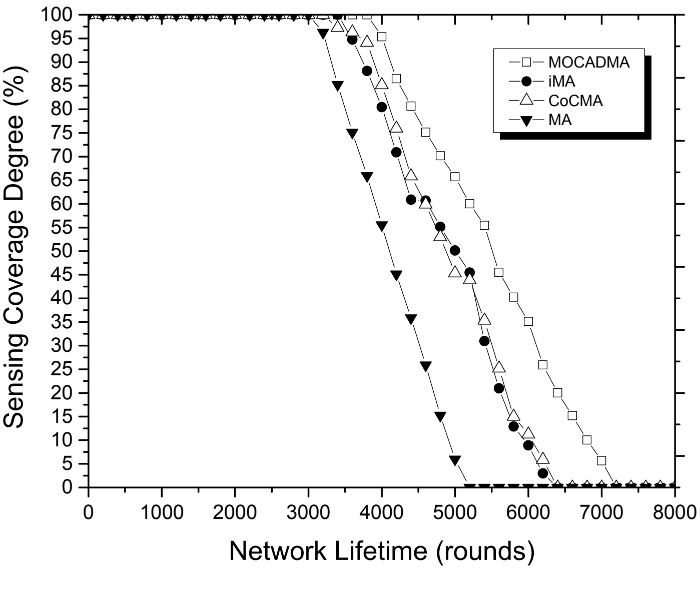


Figure 8 displays the percentage of node death *versus* network lifetime in rounds ($\alpha = 0.4$ in MOCADMA). The MOCADMA, CoCMA, iMA, and MA lose their 50% of nodes at the 5516th, 5142nd, 4889th, and 4103rd round, respectively. The simulation result demonstrates that the proposed MOCADMA significantly prolongs the lifetime of the network compared to other three methods. The proposed MOCADMA divides the WSNs at the early stage and helps form the clusters earlier than other three methods that save much energy, get the multi-objective Pareto-optimal solutions of network coverage by the fitness function presented in Equation (17) that has a quick-convergence characteristic, so the longer network lifetime can be obtained. The simulation results show that MOCADMA can achieve higher network coverage while improving the energy efficiency and effectively prolonging the network lifetime.

Figure 8. Percentage of node death *versus* network lifetime in rounds.

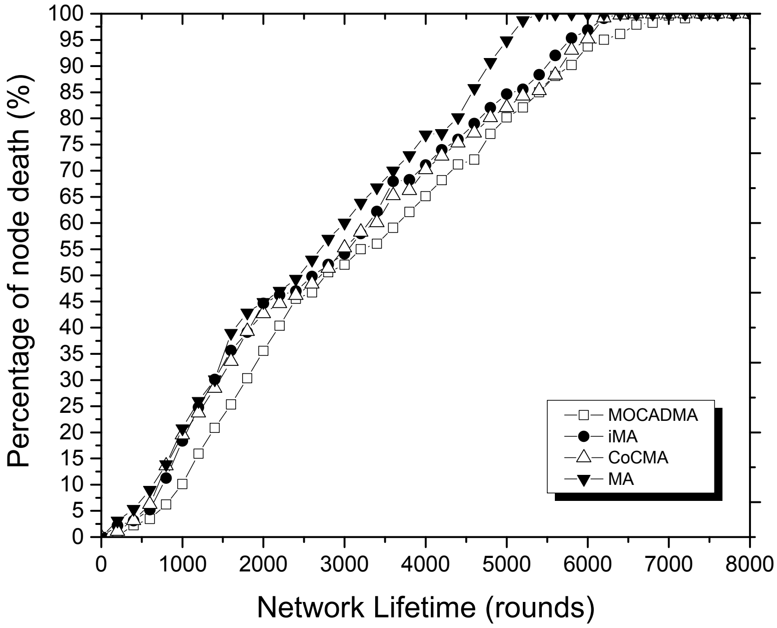
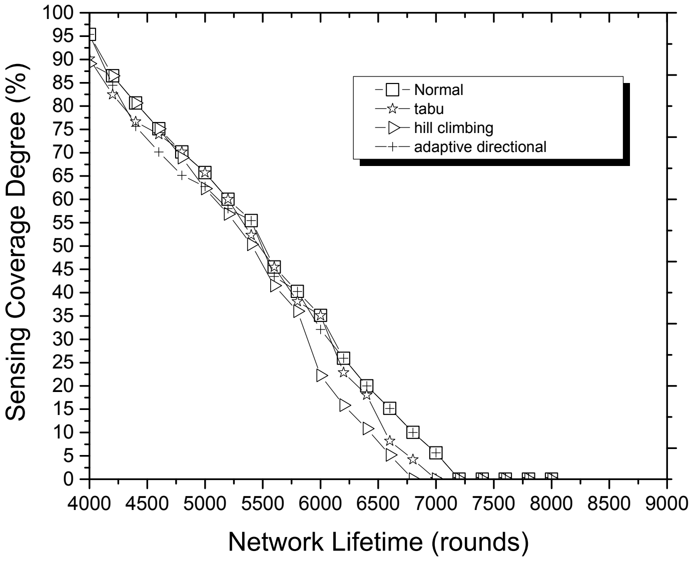


Figure 9. Performances of the dynamic local search strategy in the different cases.



Finally, we analyze the dynamic local search strategy of MOCOAMA through the simulation experiments with the same conditions as the previous experiments. In the normal case, there are tabu search algorithm, hill climbing search algorithm and adaptive directional local search strategy in the pool of local search algorithms, but when only one of those algorithms is in the pool, MOCOAMA would have a different performance. Figure 9 shows the performances of the dynamic local search strategy in the different pools, and the normal case of MOCOAMA has a better SCD than other cases. The better performance in the normal case may be due to the fact that MOCOAMA can dynamically use the advantages of the multiple local search algorithms.

7. Conclusions

This paper presents a multi-objective coverage optimization algorithm for WSNs, called MOCOAMA, which uses a memetic algorithm and gets the Pareto optimal solutions of the coverage problem. MOCOAMA maps the alternative solutions into the chromosomes represented as the matrices and uses a fitness function over sensing coverage degree, network coverage, energy consumption and Pareto ranking. We also introduce the dynamic local search strategy and a local algorithm scheduling mechanism to improving some or all chromosomes in one population. The simulation results show that MOCOAMA can solve the 1-coverage multi-objective problems for WSNs, get optimal network coverage, effectively extending the network lifetime, and have a significant improvement over some related algorithms even when the sensing error exists in the network.

In real applications, MOCOAMA depends on the sink which is assumed to have unlimited energy and finishes most of the problem-solving work according to the data sent from the sensor nodes, so how to reduce the dependence on the sink in designing memetic algorithm-based coverage algorithm shall be a challenging research task in the future.

Acknowledgments

The authors would like to thank the reviewers for their very valuable comments and suggestions, which have helped to improve the manuscript. This work was supported by the National Natural Science Foundation of China (Grant No.60905040), the Basic Research Program of Jiangsu Province (Natural Science Foundation) (Grant No.BK2011756, BK20131382), the 11th Six Talent Peaks Program of Jiangsu Province (XXRJ-009), China Postdoctoral Science Foundation (GrantNo.2012M511251, 2013M531393), Jiangsu Planned Projects for Postdoctoral Research Funds (Grant No.1101006B, 1102102C), Scientific Research Foundation of Nanjing University of Posts and Telecommunications (Grant No.NY213135, NY213152) and the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (Grant No.yx002001).

Author Contributions

Zhi Chen mainly designed the proposed algorithm and revised all the manuscript; Shuai Li initially analyzed the multi-objective coverage optimization problem, co-designed the framework of the proposed algorithm and wrote the initial manuscript under the supervision of Zhi Chen; Wenjing Yue carried out all the experiments and performance evaluation on the proposed algorithm.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless sensor networks: A survey. *Comput. Netw.* **2002**, *4*, 393–422.
2. Wang, B. *Coverage Control in Sensor Networks*; Springer: London, UK, 2008.
3. Mulligan, R.; Ammari, H.M. Coverage in wireless sensor networks: A survey. *Netw. Protoc. Algorithms* **2010**, *2*, 27–53.
4. Jia, J.; Chen, J.; Chang, G.; Wen, Y.; Song, J. Multi-Objective optimization for coverage control in wireless sensor network with adjustable sensing radius. *Comput. Math. Appl.* **2009**, *11*, 1767–1775.
5. Kulkarni, R.V.; Forster, A.; Venayagamoorthy, G.K. Computational intelligence in wireless sensor networks: A survey. *IEEE Commun. Surv. Tutor.* **2011**, *1*, 68–96.
6. Habib, S.J. Modeling and simulating coverage in sensor networks. *Comput. Commun.* **2007**, *5*, 1029–1035.
7. Ozturk, C.; Karaboga, D.; Gorkemli, B. Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm. *Sensors* **2011**, *11*, 6056–6065.
8. Kulkarni, R.V.; Venayagamoorthy, G.K. Particle swarm optimization in wireless-sensor networks: A brief survey. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2011**, *2*, 262–267.
9. Moscato, P.; Carlos, C.; Alexandre, M. Memetic algorithms. In *New Optimization Techniques in Engineering*; *Studies in Fuzziness and Soft Computing*; Godfrey, C.O., Babu, B.V., Eds.; Springer: Berlin, Germany, 2004; Volume 141, pp. 53–85.
10. Neri, F.; Carlos, C.; Pablo, M. *Handbook of Memetic Algorithms*; Springer: Berlin/Heidelberg, Germany, 2008.
11. Konstantinidis, A.; Yang, K.; Chen, H.H.; Zhang, Q. Energy-Aware topology control for wireless sensor networks using memetic algorithms. *Comput. Commun.* **2007**, *14*, 2753–2764.
12. Fleming, P.J.; Purshouse, R.C.; Lygoe, R.J. Many-Objective optimization: An engineering design perspective. In *Evolutionary Multi-Criterion Optimization*; *Lecture Notes in Computer Science*; Carlos, A.C.C., Arturo, H.A., Eckart, Z., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3410, pp. 14–32.
13. Ngatchou, P.; Zarei, A.; El-Sharkawi, M.A. Pareto multi objective optimization. In Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems (ISAP 2005), Arlington, VA, USA, 6–10 November 2005; pp. 84–91.
14. Ferentinos, K.P.; Tsiligiridis, T.A. A memetic algorithm for optimal dynamic design of wireless sensor networks. *Comput. Commun.* **2010**, *2*, 250–258.
15. Jiang, J.A.; Chen, C.P.; Chuang, C.L.; Lin, T.S.; Tseng, C.L.; Yang, E.C.; Wang, Y.C. CoCMA: Energy-Efficient coverage control in cluster-based wireless sensor networks using a memetic algorithm. *Sensors* **2009**, *9*, 4918–4940.

16. Heinzelman, W.; Chandrakasan, A.; Balakrishnan, H. An Application-Specific Protocol Architecture for Wireless Microsensor Networks. *IEEE Trans. Wirel. Commun.* **2002**, *4*, 660–670.
17. Tsai, Y.R. Coverage-preserving routing protocols for randomly distributed wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2007**, *4*, 1240–1245.
18. Ting, C.K.; Liao, C.C. A memetic algorithm for extending wireless sensor network lifetime. *Inf. Sci.* **2010**, *24*, 4818–4833.
19. Arivudainambi, D.; Balaji, S.; Rekha, D. Improved memetic algorithm for energy efficient target coverage in WSNs. In Proceedings of the IEEE 11th International Conference on Networking, Sensing and Control (ICNSC 14), Miami, FL, USA, 7–9 April 2014; pp. 261–266.
20. Chen, X.; Ong, Y.S.; Lim, M.H.; Tan, K.C. A multi-facet survey on memetic computation. *IEEE Trans. Evol. Comput.* **2011**, *5*, 591–607.
21. Zheng, D.X.; Ng, S.T.; Kumaraswamy, M.M. Applying Pareto ranking and niche formation to genetic algorithm-based multiobjective time-cost optimization. *J. Constr. Eng. Manag.* **2005**, *1*, 81–91.
22. Gendreau, M.; Potvin, J.Y. Tabu search. In *Handbook of Metaheuristics, International Series in Operations Research & Management Science*; Gendreau, M., Potvin, J.Y., Eds.; Springer US: New York, NY, USA, 2010; Volume 146, pp. 41–59.
23. Lozano, M.; Herrera, F.; Krasnogor, N.; Molina, D. Real-coded memetic algorithms with crossover hill-climbing. *Evol. Comput.* **2004**, *3*, 273–302.
24. Kim, H.; Liou, M.S. Adaptive directional local search strategy for hybrid evolutionary multiobjective optimization. *Appl. Soft Comput.* **2014**, *19*, 290–311.
25. Deshpande, A.; Montiel, C.; McLauchlan, L. Wireless Sensor Networks—A Comparative Study for Energy Minimization Using Topology Control. In Proceedings of the 2014 Sixth Annual IEEE Green Technologies Conference (GreenTech), Corpus Christi, TX, USA, 3–4 April 2014; pp. 44–48.
26. Katiyar, V.; Chand N.; Gautam, G.C.; Kumar, A. Improvement in LEACH protocol for large-scale wireless sensor networks. In Proceedings of the 2011 International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT 2011), Nagercoil, India, 23–24 March 2011; pp. 1070–1075.