

Article

Solving the Energy Efficient Coverage Problem in Wireless Sensor Networks: A Distributed Genetic Algorithm Approach with Hierarchical Fitness Evaluation

Zi-Jia Wang ¹, Zhi-Hui Zhan ^{2,*}  and Jun Zhang ²

¹ School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China; wangzjia@mail2.sysu.edu.cn

² Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China; junzhang@ieee.org

* Correspondence: zhanapollo@163.com; Tel.: +86-138-2608-9486

Received: 30 October 2018; Accepted: 13 December 2018; Published: 18 December 2018



Abstract: This paper proposed a distributed genetic algorithm (DGA) to solve the energy efficient coverage (EEC) problem in the wireless sensor networks (WSN). Due to the fact that the EEC problem is Non-deterministic Polynomial-Complete (NPC) and time-consuming, it is wise to use a nature-inspired meta-heuristic DGA approach to tackle this problem. The novelties and advantages in designing our approach and in modeling the EEC problems are as the following two aspects. Firstly, in the algorithm design, we realized DGA in the multi-processor distributed environment, where a set of processors run distributed to evaluate the fitness values in parallel to reduce the computational cost. Secondly, when we evaluate a chromosome, different from the traditional model of EEC problem in WSN that only calculates the number of disjoint sets, we proposed a hierarchical fitness evaluation and constructed a two-level fitness function to count the number of disjoint sets and the coverage performance of all the disjoint sets. Therefore, not only do we have the innovations in algorithm, but also have the contributions on the model of EEC problem in WSN. The experimental results show that our proposed DGA performs better than other state-of-the-art approaches in maximizing the number of disjoint sets.

Keywords: wireless sensor networks; energy efficient coverage; distributed genetic algorithm

1. Introduction

Wireless sensor networks (WSN) have become a hot research topic and have been widely used in numerous real-world applications, such as traffic monitoring [1], mobile computing [2], environmental observation [3], and many others [4–6]. In these sophisticated environments, in order to make full coverage and get more accurate results, many nodes should be randomly deployed in the area, causing a waste of resources. Since the sensor nodes are equipped with limited battery resources and the replacement of the battery is not feasible in many applications, low power consumption has become a critical factor to be considered when designing the WSN. Therefore, research into energy saving to prolong the network lifetime has become one of the most significant issues in WSN. Moreover, the energy saving in WSN is a significant research topic in smart and sustainable energy systems and applications [7–9].

Due to the significance of the energy efficient problem in the WSN [10–12], many efforts have been made to tackle this problem. These proposed techniques are generally divided into two categories,

where one is to design efficient protocols to reduce the energy consumption and the other one is to schedule the nodes to work efficiently. For the first category, protocols such as for the medium access control [13], for transmission [14], and for communication [15] have been proposed in the literature for reducing energy consumption. While the second categorized technique concentrates on scheduling the nodes, resulting in energy preservation and longer network lifetime. These two categories both have contributions to the energy efficiency for WSN from different approaches. The focus of this paper belongs to the second category, which is the energy efficient coverage (EEC) problem in the WSN [16,17].

In EEC, the sensor nodes are divided into different disjoint sets with the constraint that each set can guarantee the full coverage of the whole monitored area. In that way, in any time, only the necessary sensor nodes in one set are activated while the other sensor nodes that monitor the same regions can be turned off. These different disjoint sets work one by one and therefore the network lifetime can be prolonged. Moreover, with more disjoint sets that can be formed, longer network lifetime can be obtained because the nodes are scheduled to work energy efficiently [18–20]. Therefore, it is very interesting and promising for the approaches in [21–23] to divide the deployed sensor nodes into maximal disjoint sets and schedule the sets to work in turn. In this paper, we also focus on maximizing the number of disjoint sets.

Even though the above methods have been applied to the EEC problem [21–23], they can easily get trapped into local optima and cannot achieve or guarantee the full coverage because the EEC problem is NPC [21]. Therefore, in this paper, we proposed distributed genetic algorithm (DGA) to solve the EEC problems in WSN to further improve the performance due to the superior adaptation and strong global search ability of GA. Moreover, two major novel designs and advantages of DGA are described as follows:

(1) DGA is realized in the multi-processor distributed environment by the master-slave distributed model, where a set of processors run distributed to evaluate the fitness values in parallel to reduce the computational cost.

(2) When we evaluate a chromosome, different from the traditional model of EEC problem in WSN that only calculates the number of disjoint sets, a hierarchical fitness evaluation mechanism is proposed and a two-level fitness function with biased attention to the sets with larger coverage percentage is designed.

Therefore, not only do we have the innovations in algorithm, but also have the contributions on the model of EEC problem in WSN. The experimental results show that our proposed DGA performs better than other state-of-the-art approaches in maximizing the number of disjoint sets.

The rest of this paper is organized as follows. Section 2 gives the problem description of the EEC in the WSN and reviews some related work. Section 3 proposes our methodology for solving EEC in the WSN by using DGA. Section 4 presents the experimental results between our approach and other state-of-the-art approaches in the literature. Conclusions are given in Section 5.

2. Energy Efficient Coverage Problem in WSN

The EEC problem is a fundamental and significant research topic in the WSN. In this section, we present the formulation of the EEC and review the related work on this problem.

2.1. Problem Formulation

Given an $L \times W$ (Length $\times$ Width) rectangle area A and D randomly deployed sensor nodes. The EEC is to divide the nodes into several disjoint sets and then schedule these sets to work one by one. As the nodes do not have to work all the time, the energy can be significantly preserved. The aim of the EEC is to maximize the number of disjoint sets, with the constraint that each set can provide the full coverage for the monitored area.

In order to know whether the sensor network provides full coverage in the area A , we assume that the location of each sensor node is known in advance [19,24]. Moreover, the area is divided into

many grids and the coverage issue can be transformed to check whether each grid is covered by at least one active sensor node [19].

All the D sensors form the sensors set $S = \{s_1, s_2, \dots, s_D\}$, where each sensor node s_i is with the location (x_i, y_i) and the sensor radius R . For any grid $g(x, y) \in A$ in the monitored area, the relationship between the s_i and the g is defined as:

$$P(s_i, g) = \begin{cases} 1, & \text{if } (x - x_i)^2 + (y - y_i)^2 \leq R^2 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where 1 means that the grid g is covered by the sensor s_i while 0 means the sensor s_i does not cover the grid g . Therefore, for any grid g in the monitored area, if there exist at least one sensor $s_i (1 \leq i \leq D)$ that makes $P(s_i, g) = 1$, we say that A is fully covered by the sensor network.

In the EEC, the set S is divided into M subsets $S_j (1 \leq j \leq M)$, and with the objective of maximizing the value of M , as:

$$\begin{aligned} & f = \max K \\ \text{subject to} & \quad \text{(i) } \cup_{i=1}^K S_i \subseteq S \\ & \quad \text{(ii) } S_i \cap S_j = \Phi \\ & \quad \text{(iii) } \left(\oplus_{s_j \in S_i} P(s_j, g) \right) = 1, \quad \forall g \in A \end{aligned} \quad (2)$$

Here, the constraint (i) means that the unitization of all the subsets S_j must belong to the original set S . The (ii) indicates that there is no intersection between any two different subsets S_{j1} and S_{j2} . The (iii) represents that for any grid g in area A , there exists at least one sensor s_i in S_j which can cover the grid g . Obviously, these constraints can guarantee that each subset S_j can fully cover the monitored area.

2.2. Related Work

With the development of evolutionary computations (ECs) like GA [25,26], ant colony optimization (ACO) [27,28], particle swarm optimization (PSO) [29,30], and differential evolution (DE) [31,32], many researchers have applied ECs into solving the EEC problems in WSN, such as PSO-based [33] and ACO-based [34] approaches. Specifically, in [33], Zhan et al. extended the binary PSO (BPSO) to solve the EEC problem by finding a minimal set of nodes again and again to maximize the number of disjoint sets. Lin et al. [34] proposed the ACO-based approach to maximize the number of connected covers, called ACO-MNCC, to maximize the lifetime of heterogeneous WSNs by transforming the search space of the problem into a construction graph. Besides, in [35], Yang et al. proposed a probabilistic model to tackle the EEC problem in WSN, which transforms the area converge into point converge. It greatly reduces the dimension of problem. Lee et al. [36] also tried to solve the EEC problem in the WSN by using the ACO, which designs three pheromones to balance the local exploitation and global exploration. Meanwhile, they also introduced a probabilistic sensor detection model, which makes the algorithms more realistic [37].

Some researchers have also used GA for solving the EEC in the WSN recently. For example, Lai et al. [25] proposed a GA-based method to maximize the disjoint sensor covers called GAMDSC. However, there is still much room for improvement. First, their works only solved the small-scale problems (up to 140 sensors). When the scales of problems increase, their performance often face difficulties. Second, parallel or distributed methods can further improve the performance and save the computational time.

3. Methodology: DGA for the EEC Problem

In this section, the DGA for solving the EEC problem in WSN is implemented and described. The master-slave distributed framework is first given. Then the chromosome representation and the hierarchical fitness evaluation mechanism are described. After that, the genetic operators, including the selection, crossover, and mutation are proposed. At last, the overall algorithm is presented.

3.1. Distributed Framework of DGA

In DGA, the master-slave distributed model is used. The processor whose role is to control the whole evolutionary process including crossover, mutation, and selection operations, is called MASTER. The other processors, which are responsible for chromosomes' evaluation to reduce the computational costs, act as SLAVES. To give an overview of the DGA, the master-slave distributed structure is depicted in Figure 1. In each generation, the master sends chromosomes to slaves, while the slaves evaluate the chromosomes and return their corresponding fitness values back to the master [38].

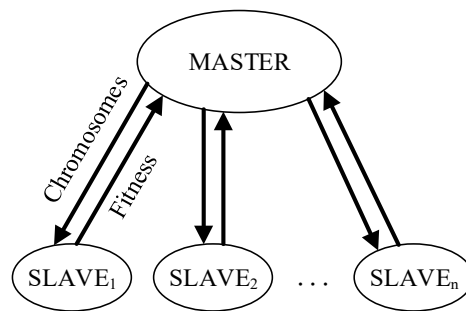


Figure 1. The distributed framework of the distributed genetic algorithm (DGA).

3.2. Chromosome Representation

The aim of the EEC problem is to maximize the number of disjoint sets. Therefore, the first thing is to determine how many sets the nodes can be divided into at most. This is also the upper bound of the disjoint sets number K [23]. Since the area has been divided into many grids, for each grid i , we find out the number of sensors k_i that cover the grid i . Then, the minimal k_i is the upper bound number of disjoint sets, i.e., $K = \min\{k_i\}$.

With the upper bound K determined, we can encode the chromosome in a very intuitive manner that each gene is an integer in the range of $[1, K]$. Therefore, the chromosome is a string of integers with the length of D , where D is the number of sensor nodes. The representation is

$$X = [x_1, x_2, \dots, x_D], \text{ where } x_i \in \mathbb{Z} \text{ and } 1 \leq x_i \leq K \quad (3)$$

In (3), the value of x_i denotes which set the i th sensor belongs to. Figure 2 gives two examples of the chromosome representation. In the examples, D is 4 and K is 2. In Figure 2a, the chromosome is $X^a = [1, 2, 1, 2]$, meaning that the first and the third sensors belong to the 1st set while the second and the fourth sensors belong to the 2nd set. In this case, it can be observed that both the 1st and the 2nd sets can provide full coverage for the monitored area. However, in Figure 2b, the chromosome is $X^b = [1, 2, 1, 1]$, meaning that the first, the third, and the fourth sensors belong to the 1st set while only the second sensor belongs to the 2nd set. In this case, only the 1st set can provide full coverage for the monitored area, but the 2nd set cannot.

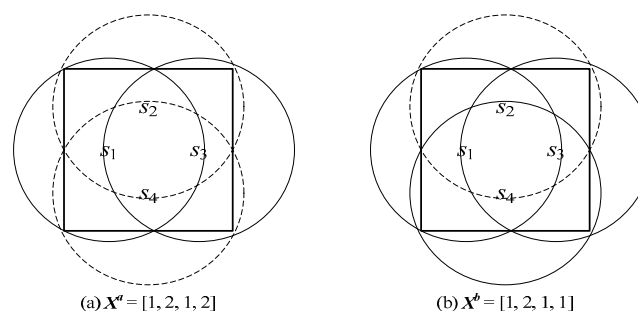


Figure 2. Illustration of the chromosome representation.

3.3. Hierarchical Fitness Evaluation

In order to evaluate the chromosome, herein, a hierarchical fitness evaluation mechanism and a two-level fitness function are proposed.

First, the genes are grouped into different sets according to their values. Then a check procedure is performed on each set to see whether the nodes in the set can provide full coverage. As we divide the monitored area into $L \times W$ grids, for each set S_i , we can count the grids number G that is covered by the sensor nodes in set S_i . Therefore, we can obtain the coverage percentage f_i as:

$$f_i = \frac{G}{L \times W} \quad (4)$$

Note that if f_i is equal to 1, it means that the set S_i can provide full coverage for the monitored area. In this way, we can count and obtain the number of disjoint sets (denoted as M) in the population which can achieve the full coverage. When two chromosomes A and B are compared, A wins B if A has a larger disjoint set number (larger M). The number of disjoint sets M is regarded as the first-level fitness evaluation.

However, to meet the case where the two chromosomes A and B have the same disjoint set number, we further designed a fitness function with biased attention to the sets with larger coverage percentage, so called the second-level fitness evaluation, which is calculated as:

$$F = \sum_{i=1}^K w_i \times f_i \quad (5)$$

where K is the upper bound of the disjoint sets and w_i is the weight for the set S_i . Herein, we give the biased attention to the sets with larger coverage percentage. That is because the set with a higher coverage percentage has a higher probability to achieve the full coverage. Therefore, we first sort the sets according to their coverage percentages in descending order, then the weight w_i for the set S_i is set as $w_i = 10,000.0/r_i$, where r_i is the rank of the set S_i in the sort of coverage percentage.

In that way, if two chromosomes A and B have the same disjoint set number, then the one with a larger fitness value is preferred (larger F).

3.4. Genetic Operators

The genetic operators in DGA consist of the selection, crossover, and mutation. In this subsection, we will briefly describe the implementations of these operators.

3.4.1. Selection

In the selection implementation, we use the tournament selection strategy. Specifically, in each selection, partial of the chromosomes are randomly selected to compete for survival. The winner enters into the next generation. Repeat the selection operator until a population size of chromosomes have been selected.

3.4.2. Crossover

After the selection, the survived chromosomes recombine to create offspring. First, for each chromosome, a random value in range $[0, 1]$ is generated. If the value is smaller than the crossover probability p_c , then the chromosome is used as one of the parents. After all the parents have been determined, every two parents are randomly mated to create two offspring. A crossover position k is randomly generated and the genes from the k th position of the two parents are exchanged, as shown in Figure 3a. In this way, two new chromosomes, so called offspring are generated and enter the population.

3.4.3. Mutation

The chromosomes will perform a mutation operation after the crossover. For each gene in a chromosome, a random value in range $[0, 1]$ is generated and compared with the mutation probability p_m . If the random value is smaller than p_m , then the mutation occurs. As shown in Figure 3b, when the mutation occurs, the gene is set as a random integer in the range $[1, K]$.

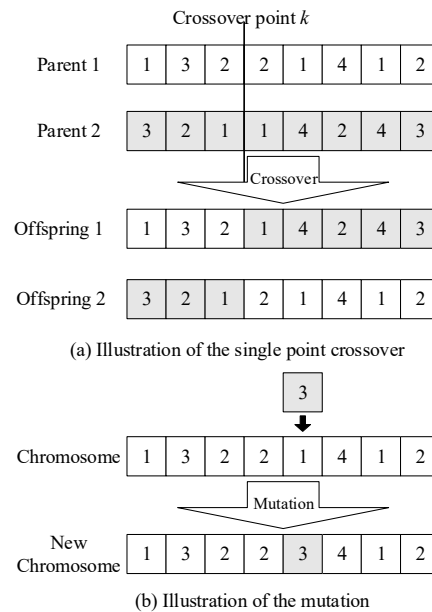


Figure 3. The illustration of the crossover and mutation operators.

3.5. Complete Algorithm

With the designs of the chromosome representation, fitness function, and the genetic operators, the DGA for solving the EEC problem is shown in Figure 4 and is described as the following seven steps.

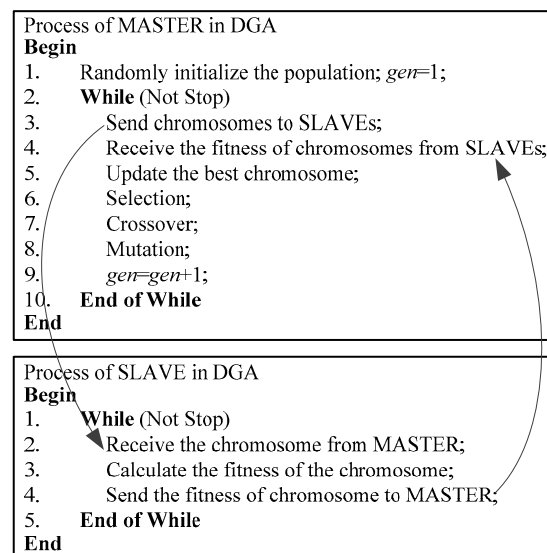


Figure 4. The flowchart of the DGA for solving the energy efficient coverage (EEC) problem in the wireless sensor networks (WSN).

Step 1: Initialization. The N chromosomes in the population are initialized at master. Each gene in the chromosome is set as a random integer value in the range $[1, K]$ as (3).

Step 2: Evaluation. Each chromosome is evaluated as (5) at the slave and returns its fitness to the master. Then the best chromosome with the highest fitness can be determined. This chromosome is compared with the historical best one and will replace the historical best one if it is better. Otherwise, the historical best chromosome will replace the worst chromosome in the current generation.

Step 3: Selection. The selection operation has been discussed in Section 3.4.1.

Step 4: Crossover. The crossover operation has been discussed in Section 3.4.2.

Step 5: Mutation. The mutation operation has been discussed in Section 3.4.3.

Step 6: Termination check. If the termination criteria is met, go to Step 7. Herein, the termination criteria include the obtaining of the upper bound K or reaching the maximal generation number. Otherwise, the algorithm goes to Step 2 and continues the next generation evolution.

Step 7: Terminate. Output the best chromosome as the final solution.

3.6. Computational Complexity

Herein, we denote the population size and the dimension of problem (the number of sensors) as N and D , respectively. First, in the initialization, the computational complexity of DGA is $O(N \times D)$, which is obtained by step 1 in MASTER process of Figure 4. Then, as for the chromosomes evaluation, the computational complexity is $O(N)$, as obtained by steps 3–4 in the MASTER process and steps 2–4 in the SLAVE process of Figure 4. After that, in the chromosome evolution, since we used the tournament selection, the computational complexity is $O(N^2 \times D)$, as obtained by steps 6–8 in the MASTER process of Figure 4. Therefore, the overall computational complexity of DGA is $O(N^2 \times D)$.

4. Experiments and Comparisons

4.1. Algorithms Configurations

The parameter configurations of the DGA are listed as Table 1 and are described as follows.

Table 1. The Parameters Configurations of DGA.

Parameter	Configuration
N	40
Generation	200
p_c	0.8
p_m	0.01

The population size is 40. Several processors are used here to evaluate the fitness of chromosomes. In each tournament selection round, 20% of the chromosomes are randomly chosen and compete for survival. The crossover probability p_c and mutation probability p_m are 0.8 and 0.01, respectively. The algorithm terminates at the maximal generations of 200 or it obtains the upper bound K of the disjoint sets number.

4.2. Experimental Results and Comparisons

In our experiments, we choose several EA-based algorithms, including a PSO-based method (BPSO [33]) and a GA-based method (GAMDSC [25]) to compare with our proposed DGA. To have a reliable and fair comparison, the parameter configurations of all the competitor algorithms are set the same as suggested in their original papers.

Herein, the network topology configurations used in [19] are adopted. That is, with the monitored area 50 m by 50 m, the sensing range R spans 8 m, 10 m, and 12 m, whilst the deployed nodes number N can be 100, 150, 200, 250, and 300. For each case with N original deployed nodes, we randomly deploy the D nodes in the network to form the topology. Note that the D nodes must provide the full coverage. Otherwise, the nodes are randomly deployed again until the network is fully covered. On each case, we generate three different random topologies. Therefore, there are totally $3 \times 3 \times 5 = 45$ cases in

this experiment. Due to the stochastic characteristic of the algorithms, we simulate each case on each topology 10 times and the mean results are recorded for statistics. All the approaches are dealing with the same test environments, resulting in a fair comparison.

The simulation results are presented and compared in Table 2. For clarity, the best results are highlighted in boldface. The results show that the DGA outperforms the other approaches on most of the cases. Also, DGA can maximize the number of disjoint sets and obtain the upper bound K in most of the cases, except cases 34, 42–44, while other algorithms can only find few disjoint sets. Moreover, with the increase in the number of nodes, especially in cases 37–45, the superiority of DGA is increasingly obvious, which further indicates the dominance of DGA when solving the complicated EEC problem in WSN.

Table 2. Experimental Results of Comparisons.

Case No.	Topology		Upper Bound K	All Combinations	Mean Disjoint Set Number M			Error		
	Node	Range			GAMDSC	BPSO	DGA	GAMDSC	BPSO	DGA
1	100	8	2	2*100	2.000	2.000	2.000	0.000	0.000	0.000
2			2	2*100	2.000	2.000	2.000	0.000	0.000	0.000
3			2	2*100	2.000	2.000	2.000	0.000	0.000	0.000
4	100	10	2	2*100	2.000	2.000	2.000	0.000	0.000	0.000
5			2	2*100	2.000	2.000	2.000	0.000	0.000	0.000
6			2	2*100	2.000	2.000	2.000	0.000	0.000	0.000
7	100	12	2	2*100	2.000	2.000	2.000	0.000	0.000	0.000
8			2	2*100	2.000	2.000	2.000	0.000	0.000	0.000
9			4	4*100	3.900	3.600	4.000	0.025	0.100	0.000
10	150	8	2	2*150	2.000	2.000	2.000	0.000	0.000	0.000
11			2	2*150	2.000	2.000	2.000	0.000	0.000	0.000
12			2	2*150	2.000	2.000	2.000	0.000	0.000	0.000
13	150	10	3	3*150	3.000	3.000	3.000	0.000	0.000	0.000
14			4	4*150	3.500	3.200	4.000	0.125	0.200	0.000
15			3	3*150	3.000	3.000	3.000	0.000	0.000	0.000
16	150	12	5	5*150	4.200	5.000	5.000	0.000	0.000	0.000
17			3	3*150	3.000	3.000	3.000	0.000	0.000	0.000
18			4	4*150	3.800	4.000	4.000	0.050	0.000	0.000
19	200	8	2	2*200	2.000	2.000	2.000	0.000	0.000	0.000
20			2	2*200	2.000	2.000	2.000	0.000	0.000	0.000
21			3	3*200	3.000	3.000	3.000	0.000	0.000	0.000
22	200	10	3	3*200	3.000	3.000	3.000	0.000	0.000	0.000
23			4	4*200	3.500	3.300	4.000	0.125	0.175	0.000
24			5	5*200	5.000	4.600	5.000	0.000	0.080	0.000
25	200	12	7	7*200	5.900	6.200	7.000	0.157	0.114	0.000
26			8	8*200	7.200	6.600	8.000	0.100	0.175	0.000
27			9	9*200	7.300	7.500	9.000	0.189	0.166	0.000
28	250	8	3	3*250	3.000	3.000	3.000	0.000	0.000	0.000
29			3	3*250	3.000	3.000	3.000	0.000	0.000	0.000
30			5	5*250	4.900	4.100	5.000	0.020	0.180	0.000
31	250	10	5	5*250	5.000	5.000	5.000	0.000	0.000	0.000
32			7	7*250	6.300	6.600	7.000	0.100	0.057	0.000
33			6	6*250	5.700	5.900	6.000	0.050	0.016	0.000
34	250	12	11	11*250	9.900	9.700	10.00	0.100	0.118	0.091
35			9	9*250	8.600	8.700	9.000	0.044	0.033	0.000
36			8	8*250	8.000	8.000	8.000	0.000	0.000	0.000
37	300	8	6	6*300	6.000	6.000	6.000	0.000	0.000	0.000
38			3	3*300	3.000	3.000	3.000	0.000	0.000	0.000
39			5	5*300	5.000	5.000	5.000	0.000	0.000	0.000
40	300	10	7	7*300	7.000	6.600	7.000	0.000	0.057	0.000
41			7	7*300	7.000	6.800	7.000	0.000	0.029	0.000
42			9	9*300	8.300	7.400	8.400	0.100	0.178	0.067
43	300	12	11	11*300	10.60	10.00	10.10	0.036	0.091	0.082
44			12	12*300	10.80	10.90	11.30	0.100	0.092	0.058
45			9	9*300	8.800	8.600	9.000	0.022	0.044	0.000

We also compare the *Error* values of these approaches when dealing with the EEC problem. The *Error* is calculated as

$$\text{Error} = (K - M)/K \quad (6)$$

where K is the upper bound in each case and M is the mean disjoint set number obtained by the approach. The *Error* values of DGA are smaller than 0.1 in all the cases and most of the *Error* values are 0, indicating the DGA is very promising in producing high quality solutions.

4.3. Effects of Parameters

The DGA involves two parameters, the crossover probability p_c and the mutation probability p_m . In this part, we investigate the two parameters based on the same topologies as in Section 4.2.

First, the p_c is set as 0.8 and the p_m varies from 0.01 to 0.09. The mean *Error* values of the 45 cases with different p_m are plotted in Figure 5. As we can see, the small value of p_m generally performs better than the large value of p_m . That may because the larger p_m destroys the convergence ability, which makes the algorithm act like a random search and cannot converge to a promising region. Therefore, the investigated results indicate that the DGA obtains the best performance when the p_m is set as 0.01.

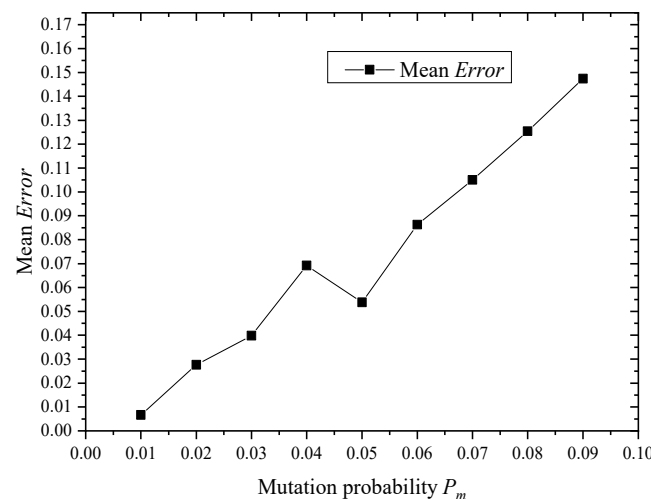


Figure 5. The mean *Error* values obtained by the DGA with different p_m .

Then, the p_c is investigated, with the p_m set as 0.01. The value of p_c spans from 0.1 to 0.9, with the step 0.1. The mean *Error* values of the 45 cases with different p_c are plotted in the Figure 6. It can be observed that the large value of p_c may be generally better than the small value of p_c . This may be due to large p_c can making the good information of the chromosome spread fast in the population. This is very helpful for the algorithm to use good information to search in the promising region, and therefore results in better performance. However, a too-large p_c value will sometimes make the algorithm converge too fast and be trapped into local optima. For example, when $p_c = 0.9$, the mean *Error* values are worse than that when $p_c = 0.8$. In general, $p_c = 0.8$ is promising for the DGA to obtain satisfying results.

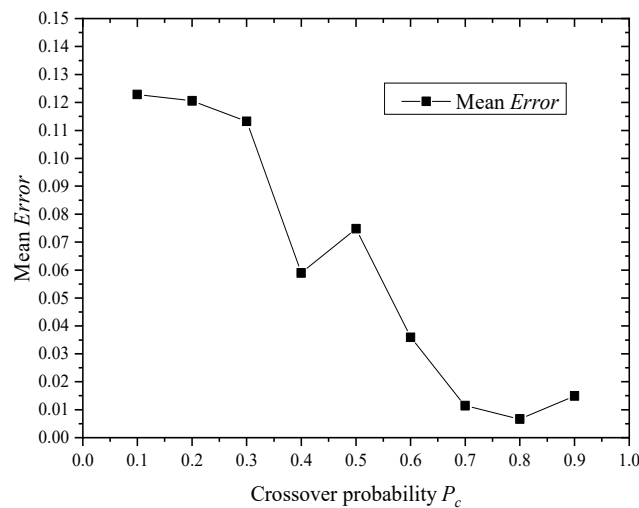


Figure 6. The mean *Error* values obtained by the DGA with different p_c .

4.4. Speedup Ratio

Another advantage of DGA is the parallel implementation, where several processors in DGA are used to evaluate the fitness of chromosomes independently. Table 3 shows the computational times of DGA on all the 45 cases with different processors.

Speedup ratio is the metric to measure the performance of a distributed or parallel algorithm, and it varies with the numbers of processors and the computation cost of the case. Herein, we use the small-scale case 1 and the large-scale case 45 as two representative instances. We test the speedup of DGA on these two cases and draw the speedup curve with the increasing processors, from 2 to 10, shown in Figure 7.

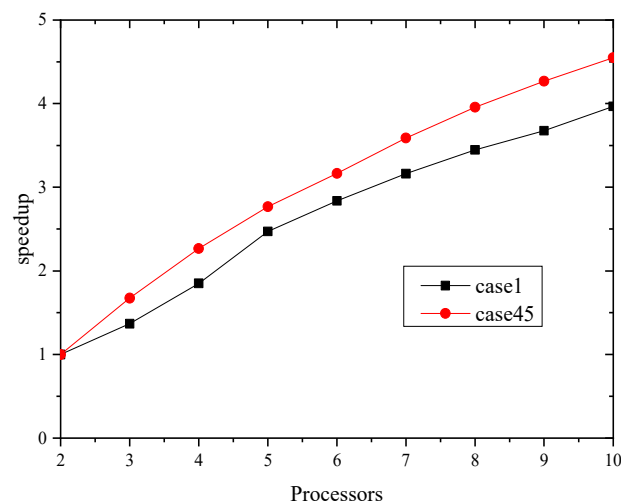


Figure 7. Speedup ratios achieved by DGA on two typical cases.

As we can see, the speedup ratio consistently grows with the increase of the number of processors on both of these two cases. However, the increase of speedup ratio slows down when the number of processors increases from 6 to 10. That may due to the huge communication cost among many processors. Even so, the speedup ratio still keeps growing with the increasing of the number of processors. Moreover, the speedup ratio of case 45 is larger the case 1. That may because case 45 is more complicated than case 1, which will cause more computation cost than case 1. Therefore, under a similar communication cost, the higher computation cost will directly lead to a higher speedup ratio in case 45.

Table 3. Computational time of DGA (s).

Processors Case No.	2	3	4	5	6	7	8	9	10
1	101.53	74.26	54.91	41.07	35.78	32.09	29.45	27.62	25.59
2	123.24	90.50	71.44	57.29	53.83	50.38	46.36	42.50	39.85
3	133.24	96.64	77.30	69.96	62.33	59.80	54.87	50.64	48.59
4	116.77	85.19	70.76	63.55	56.58	52.39	48.84	43.78	39.49
5	147.74	100.36	87.53	73.46	65.86	59.73	53.92	50.05	48.74
6	132.90	95.19	83.47	71.41	64.11	58.53	53.29	49.39	47.98
7	158.20	107.27	88.06	76.62	68.54	63.16	56.23	52.28	49.77
8	144.34	103.50	86.77	77.06	68.96	62.38	57.05	52.48	49.21
9	258.93	169.97	130.72	101.29	89.81	80.90	72.86	65.29	61.41
10	188.84	120.19	93.42	82.38	71.84	64.46	56.94	50.00	47.80
11	178.36	118.36	93.21	81.42	70.64	63.89	56.37	49.08	46.32
12	206.96	128.08	98.24	84.18	73.99	65.19	58.03	53.99	50.02
13	253.12	142.81	110.06	94.95	84.88	77.37	73.10	69.00	67.81
14	321.11	205.79	158.70	139.89	126.43	113.42	100.45	90.67	84.10
15	272.37	164.89	142.60	123.77	105.18	90.94	78.09	69.07	63.82
16	386.60	267.58	201.73	175.17	139.95	123.07	110.80	100.43	94.91
17	284.46	214.86	151.65	123.60	104.80	90.44	88.16	79.07	73.94
18	308.81	223.46	176.00	143.88	122.89	110.73	97.94	89.17	82.92
19	267.02	178.39	145.59	123.05	101.54	90.55	81.90	74.43	69.95
20	284.36	193.04	152.46	125.60	104.22	95.95	86.24	79.70	74.18
21	349.97	249.11	185.50	145.96	120.79	108.76	96.98	88.63	83.98
22	351.04	246.52	189.91	156.78	136.28	122.45	109.88	102.45	96.40
23	445.03	323.38	236.05	196.38	167.02	147.60	129.79	120.12	115.23
24	533.63	354.34	286.87	242.26	219.79	189.75	162.89	147.38	139.96
25	627.89	457.94	346.32	271.15	224.81	198.03	176.26	166.13	157.15
26	682.48	475.31	363.82	288.01	236.70	209.82	189.36	175.76	170.80
27	744.13	501.08	389.95	305.20	248.85	228.48	216.66	206.95	199.67
28	436.20	281.56	233.98	198.82	165.39	145.43	134.42	119.83	112.49
29	412.85	265.29	215.59	194.15	169.05	148.05	133.49	120.88	112.55
30	563.58	378.05	295.38	245.90	202.31	171.01	156.44	148.80	141.01
31	591.87	380.09	308.38	254.81	214.68	189.08	169.17	161.44	156.99
32	680.79	485.09	354.86	278.55	236.14	201.17	188.06	179.64	173.99
33	625.32	454.75	336.59	257.97	224.26	200.25	184.99	171.09	166.34
34	1687.20	1153.09	916.81	799.34	684.47	585.99	509.48	451.04	424.53
35	871.04	578.18	449.48	388.74	343.93	309.37	264.36	241.62	225.94
36	778.21	560.31	401.16	372.50	327.59	290.98	262.23	236.73	220.14
37	1014.52	697.61	580.35	502.69	439.96	384.46	321.52	288.86	272.87
38	836.59	580.41	446.17	385.78	332.98	294.35	251.36	222.99	210.33
39	958.63	635.55	528.04	418.04	350.24	303.71	264.91	244.66	233.68
40	1302.70	825.53	671.78	598.59	522.32	451.68	400.26	363.22	342.84
41	1412.63	915.99	732.56	603.58	508.87	449.00	414.48	382.68	367.28
42	2297.81	1542.50	1208.43	1048.97	915.04	801.93	713.63	638.74	586.86
43	2416.68	1651.56	1217.74	1082.13	951.69	837.25	732.56	651.62	606.25
44	2530.80	1754.12	1317.12	1181.00	1008.09	886.37	782.98	696.21	634.14
45	1916.47	1144.10	845.52	692.63	605.70	534.16	484.21	449.11	421.29

5. Conclusions

In this paper, DGA is proposed to solve the EEC problem in the WSN to achieve or guarantee the full converge. In the implementation, DGA is realized in the multi-processor distributed environment, where a set of processors run distributed to evaluate the fitness values in parallel to reduce the computational cost. Moreover, when we evaluate a chromosome, different from the traditional model of EEC problem in WSN that only calculates the number of disjoint sets, hierarchical fitness evaluation mechanism is proposed and a two-level fitness function with biased attention to the sets with larger coverage percentage is further designed. Therefore, not only do we have the innovations in the algorithm, but we also have the contributions to the model of the EEC problem in WSN. Simulations have been conducted and the experimental results confirm the effectiveness and efficiency of the proposed approach when compared with the state-of-the-art approaches.

For future work, we wish to further improve the performance of DGA by considering the potential uncertainties, so as to make DGA more applicable in the real applications. Moreover, we wish to apply the DGA into other more complicated optimization problems, like data routing [39], cloud resource scheduling [40], supply chain managing, even in a dynamic and multi-objective environment [41,42], not only in the WSN.

Author Contributions: Z.-J.W. conducted the experiments, performed the experiments, and wrote the draft of this paper. The idea of this paper was proposed by Z.-H.Z., who also made contributions to help organize and write the paper, together with J.Z.

Funding: This work was partially supported by the Outstanding Youth Science Foundation with No. 61822602, the National Natural Science Foundations of China (NSFC) with No. 61772207 and 61332002, the Natural Science Foundations of Guangdong Province for Distinguished Young Scholars with No. 2014A030306038, the Project for Pearl River New Star in Science and Technology with No. 201506010047, the GDUPS (2016), and the Fundamental Research Funds for the Central Universities.

Acknowledgments: The authors would like to thank the assistant editor and reviewers for their valuable comments and suggestions that improved the paper's quality.

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

Nomenclature

DGA	distributed genetic algorithm
EEC	energy efficient coverage
WSN	wireless sensor networks
D	the number of sensor nodes
M	the number of disjoint subsets
K	the upper bound of the disjoint sets number
N	the number of chromosomes
p_c	crossover probability
p_m	mutation probability

References

1. Guevara, J.; Barrero, F.; Vargas, E.; Becerra, J.; Toral, S. Environmental wireless sensor network for road traffic applications. *IET Intell. Transp. Syst.* **2012**, *6*, 177–186. [\[CrossRef\]](#)
2. Wang, C.; Guo, S.; Yang, Y. An optimization framework for mobile data collection in energy-harvesting wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *15*, 2969–2986. [\[CrossRef\]](#)
3. Martinez, K.; Hart, J.K.; Ong, R. Environmental sensor networks. *IEEE Comput. Soc.* **2004**, *37*, 50–56. [\[CrossRef\]](#)
4. Yahya, A.; Islam, S.; Akhunzada, A.; Ahmed, G.; Shamshirband, S.; Lloret, J. Towards efficient sink mobility in underwater wireless sensor networks. *Energies* **2018**, *11*, 1471. [\[CrossRef\]](#)
5. Al-Jaoufi, M.A.A.; Liu, Y.; Zhang, Z.J. An active defense model with low power consumption and deviation for wireless sensor networks utilizing evolutionary game theory. *Energies* **2018**, *11*, 1281. [\[CrossRef\]](#)
6. Shehadeh, H.A.; Idris, M.Y.I.; Ahmedy, I.; Ramli, R.; Noor, N.M. The multi-objective optimization algorithm based on sperm fertilization procedure (MOSFP) method for solving wireless sensor networks optimization problems in smart grid applications. *Energies* **2018**, *11*, 97. [\[CrossRef\]](#)
7. Marzband, M.; Azarinejadian, F.; Savaghebi, M.; Pouresmaeil, E.; Guerrero, J.M.; Lightbody, G. Smart transactive energy framework in grid-connected multiple home microgrids under independent and coalition operations. *Renew. Energy* **2018**, *126*, 95–106. [\[CrossRef\]](#)
8. Marzband, M.; Fouladfar, M.H.; Akorede, M.F.; Lightbody, G.; Pouresmaeil, E. Framework for smart transactive energy in home-microgrids considering coalition formation and demand side management. *Sustain. Cities Soc.* **2018**, *40*, 136–154. [\[CrossRef\]](#)
9. Tavakoli, M.; Shokridehaki, F.; Akorede, M.F.; Marzband, M.; Vechiu, I.; Pouresmaeil, E. CVaR-based energy management scheme for optimal resilience and operational cost in commercial building microgrids. *Int. J. Electr. Power Energy Syst.* **2018**, *100*, 1–9. [\[CrossRef\]](#)

10. Marzband, M.; Javadi, M.; Pourmousavi, S.A.; Lightbody, G. An advanced retail electricity market for active distribution systems and home microgrid interoperability based on game theory. *Electr. Power Syst. Res.* **2018**, *157*, 187–199. [[CrossRef](#)]
11. Tavakoli, M.; Shokridehaki, F.; Marzband, M.; Godina, R.; Pouresmaeil, E. A two stage hierarchical control approach for the optimal energy management in commercial building microgrids based on local wind power and PEVs. *Sustain. Cities Soc.* **2018**, *41*, 332–340. [[CrossRef](#)]
12. Marzband, M.; Sumper, A.; Domínguez-García, J.L.; Gumara-Ferreta, R. Experimental validation of a real time energy management system for microgrids in islanded mode using a local day-ahead electricity market and MINLP. *Energy Convers. Manag.* **2013**, *76*, 314–322. [[CrossRef](#)]
13. Rajendran, V.; Obraczka, K.; Garcia-Luna-Aceves, J.J. Energy-efficient collision-free medium access control for wireless sensor networks. *Wirel. Netw.* **2006**, *12*, 63–78. [[CrossRef](#)]
14. Chlamtac, I.; Petrioli, C.; Redi, J. Energy-conserving access protocols for identification networks. *IEEE/ACM Trans. Netw.* **1999**, *7*, 51–59. [[CrossRef](#)]
15. Stine, J.; Veciana, G. Improving energy efficiency of centrally controlled wireless data networks. *Wirel. Netw.* **2002**, *8*, 681–700. [[CrossRef](#)]
16. Huang, C.F.; Tseng, Y.C. A survey of solutions to the coverage problems in wireless sensor networks. *J. Internet Technol.* **2005**, *6*, 1–8.
17. Ammari, H.M. Investigating the energy sink-hole problem in connected-covered wireless sensor networks. *IEEE Trans. Comput.* **2014**, *63*, 2729–2742. [[CrossRef](#)]
18. Cardei, M.; Wu, J. Energy-efficient coverage problems in wireless ad-hoc sensor networks. *Comput. Commun.* **2006**, *29*, 413–420. [[CrossRef](#)]
19. Tian, D.; Georganas, N.D. A node scheduling scheme for energy conservation in large wireless sensor networks. *Wirel. Commun. Mob. Comput.* **2003**, *3*, 271–290. [[CrossRef](#)]
20. Zhan, Z.H.; Zhang, J.; Fan, Z. Solving the optimal coverage problem in wireless sensor networks using evolutionary computation algorithms. In Proceedings of the International Conference on Simulated Evolution and Learning, Kanpur, India, 1–4 December 2010; pp. 166–176.
21. Slijepcevic, S.; Potkonjak, M. Power efficient organization of wireless sensor networks. In Proceedings of the IEEE International Conference on Communications, Helsinki, Finland, 11–14 June 2001.
22. Cardei, M.; MacCallum, D.; Cheng, X.; Min, M.; Jia, X.; Li, D.; Du, D.Z. Wireless sensor networks with energy efficient organization. *J. Interconnect. Netw.* **2002**, *3*, 213–229. [[CrossRef](#)]
23. Zhang, X.Y.; Zhang, J.; Gong, Y.J.; Zhan, Z.H.; Chen, W.N.; Li, Y. Kuhn-munkres parallel genetic algorithm for the set cover problem and its application to large-scale wireless sensor networks. *IEEE Trans. Evol. Comput.* **2016**, *20*, 695–710. [[CrossRef](#)]
24. Li, X.Y.; Wan, P.J.; Frieder, O. Coverage in wireless and ad-hoc sensor networks. *IEEE Trans. Comput.* **2003**, *52*, 753–762. [[CrossRef](#)]
25. Lai, C.C.; Ting, C.K.; Ko, R.S. An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications. In Proceedings of the IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007.
26. Kampelis, N.; Tsekeri, E.; Kolokotsa, D.; Kalaitzakis, K.; Isidori, D.; Cristina, C. 3 Development of demand response energy management optimization at building and district levels using genetic algorithm and artificial neural network modelling power predictions. *Energies* **2018**, *11*, 3012. [[CrossRef](#)]
27. Liu, X.F.; Zhan, Z.H.; Deng, D.; Li, Y.; Gu, T.L.; Zhang, J. An energy efficient ant colony system for virtual machine placement in cloud computing. *IEEE Trans. Evol. Comput.* **2018**, *22*, 113–128. [[CrossRef](#)]
28. Chen, Z.G.; Zhan, Z.H.; Lin, Y.; Gong, Y.J.; Gu, T.L.; Zhao, F.; Yuan, H.Q.; Chen, X.; Li, Q.; Zhang, J. Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach. *IEEE Trans. Cybern.* **2018**. [[CrossRef](#)] [[PubMed](#)]
29. Li, Y.H.; Zhan, Z.H.; Lin, S.J.; Zhang, J.; Luo, X.N. Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems. *Inf. Sci.* **2015**, *293*, 370–382. [[CrossRef](#)]
30. Liu, X.F.; Zhan, Z.H.; Gao, Y.; Zhang, J.; Kwong, S.; Zhang, J. Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization. *IEEE Trans. Evol. Comput.* **2018**. [[CrossRef](#)]

31. Wang, Z.J.; Zhan, Z.H.; Lin, Y.; Yu, W.J.; Yuan, H.Q.; Gu, T.L.; Kwong, S.; Zhang, J. Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems. *IEEE Trans. Evol. Comput.* **2018**, *22*, 894–908. [[CrossRef](#)]
32. Liu, X.F.; Zhan, Z.H.; Lin, Y.; Chen, W.N.; Gong, Y.J.; Gu, T.L.; Yuan, H.Q.; Zhang, J. Historical and heuristic based adaptive differential evolution. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**. [[CrossRef](#)]
33. Zhan, Z.H.; Zhang, J.; Du, K.J.; Xiao, J. Extended binary particle swarm optimization approach for disjoint set covers problem in wireless sensor networks. In Proceedings of the IEEE Conference on Technologies and Applications of Artificial Intelligence, Tainan, Taiwan, 16–18 November 2012.
34. Lin, Y.; Zhang, J.; Chung, H.S.H.; Ip, W.H.; Li, Y.; Shi, Y.H. An ant colony optimization approach for maximizing the lifetime of heterogeneous wireless sensor networks. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2012**, *42*, 408–420. [[CrossRef](#)]
35. Yang, Q.Q.; He, S.B.; Li, J.K.; Chen, J.M.; Sun, Y.X. Energy-efficient probabilistic area coverage in wireless sensor networks. *IEEE Trans. Veh. Technol.* **2015**, *64*, 367–377. [[CrossRef](#)]
36. Lee, J.W.; Choi, B.S.; Lee, J.J. Energy-efficient coverage of wireless sensor networks using ant colony optimization with three types of pheromones. *IEEE Trans. Ind. Informat.* **2011**, *7*, 419–427. [[CrossRef](#)]
37. Lee, J.W.; Lee, J.Y.; Lee, J.J. Jenga-inspired optimization algorithm for energy-efficient coverage of unstructured WSNs. *IEEE Wirel. Commun. Lett.* **2013**, *2*, 34–37. [[CrossRef](#)]
38. Zhan, Z.H.; Liu, X.F.; Zhang, H.; Yu, Z.; Weng, J.; Li, Y.; Gu, T.; Zhang, J. Cloudde: A heterogeneous differential evolution algorithm and its distributed cloud version. *IEEE Trans. Parallel Distrib. Syst.* **2017**, *28*, 704–716. [[CrossRef](#)]
39. Shen, M.; Zhan, Z.H.; Chen, W.N.; Gong, Y.J.; Zhang, J.; Li, Y. Bi-velocity discrete particle swarm optimization and its application to multicast routing problem in communication networks. *IEEE Trans. Ind. Electron.* **2014**, *61*, 7141–7151. [[CrossRef](#)]
40. Zhan, Z.H.; Liu, X.F.; Gong, Y.J.; Zhang, J.; Chung, H.S.H.; Li, Y. Cloud computing resource scheduling and a survey of its evolutionary approaches. *ACM Comput. Surv.* **2015**, *47*, 1–33. [[CrossRef](#)]
41. Liu, X.F.; Zhan, Z.H.; Zhang, J. Neural network for change direction prediction in dynamic optimization. *IEEE Access* **2018**, *6*, 72649–72662. [[CrossRef](#)]
42. Zhan, Z.H.; Li, J.; Cao, J.; Zhang, J.; Chung, H.; Shi, Y.H. Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems. *IEEE Trans. Cybern.* **2013**, *43*, 445–463. [[CrossRef](#)] [[PubMed](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).