

Research Article

A Biologically Inspired Energy-Efficient Duty Cycle Design Method for Wireless Sensor Networks

Jie Zhou

College of Information Science and Technology, Shihezi University, Shihezi, China

Correspondence should be addressed to Jie Zhou; jiezhou@shzu.edu.cn

Received 4 February 2017; Revised 30 March 2017; Accepted 26 April 2017; Published 28 June 2017

Academic Editor: Gabriele Cazzulani

Copyright © 2017 Jie Zhou. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The recent success of emerging wireless sensor networks technology has encouraged researchers to develop new energy-efficient duty cycle design algorithm in this field. The energy-efficient duty cycle design problem is a typical NP-hard combinatorial optimization problem. In this paper, we investigate an improved elite immune evolutionary algorithm (IEIEA) strategy to optimize energy-efficient duty cycle design scheme and monitored area jointly to enhance the network lifetimes. Simulation results show that the network lifetime of the proposed IEIEA method increased compared to the other two methods, which means that the proposed method improves the full coverage constraints.

1. Introduction

Recent technological advances in sensing, nanosystems technologies, and communication have made it possible to equip inexpensive small, low-cost, vulnerable, and fast response sensing units [1]. Wireless sensor networks (WSNs) are composed of some sensors having limited sensing, computing, communication, and self-organizing abilities [2]. Each sensor node consists of five modules: the computation module, the data acquisition module, the RF module, the data storage module, and the power module [3]. Wireless sensor networks have applications in many areas such as military surveillance, traffic avoidance, intelligent family, preventing forest fire loss, building monitoring and control, and advanced health care delivery [4].

The energy-efficient duty cycle design problem has recently attracted the attention of many researchers in the field of wireless sensor networks [5]. Limited by their size, small wireless sensors are equipped with restricted sensing capacity [6]. To have a long network lifetime, energy-efficient duty cycle scheme should be designed properly. However, finding the ideal energy-efficient duty cycle design is an NP-hard problem. For large-scale wireless sensor networks, the exhaustive search cannot be used to get the ideal duty cycle design in real time [7].

Due to its computational complexity, many heuristics are proposed to get near-optimal solutions in reasonable time. In [8], an energy-efficient duty cycle design technique that enables trade-offs between computation cost and network lifetime is investigated. It provides a wider search space by randomly selecting energy-efficient duty cycle design solutions when each one in the population gets updated. However, the reference does not take into consideration factors such as sensing radius. A research effort to the energy-efficient duty cycle design for the wireless sensor networks based on the genetic algorithm (GA) can be found in [9]. The work therein focused on an energy-efficient duty cycle design optimization with many constraints. But it suffers from premature convergence and low convergence rate when the number of nodes is high. A particle swarm optimization (PSO) is developed by employing a refined fitness evaluation technique [10]. In the work therein, the authors maximize the network lifetime without considering the fact that the energy of a battery is limited. PSO approach has a problem of algorithm convergence and complexity.

Elite computing and immune theory have always attracted attention of the scholars of artificial intelligence [11]. Besides that, the superior performance of evolutionary theory for combinatorial optimization problems is demonstrated in [12, 13]. In particular, in this paper, an improved elite immune

evolutionary algorithm (IEIEA) is investigated to effectively represent the individuals to explore the search space with a small group and to exploit the global optimal solution in the search space within a few iterations, respectively.

We first formulate an optimization problem as integer programming that is proven to be NP-complete. Then advanced evolutionary algorithm is used to solve the problem. Moreover, immune clone operator is adopted to enhance global search ability. The immune clone strategy helps to avoid local optima, and the improved stopping criterion automatically finds the best solution. Extensive simulations are conducted which compare IEIEA with SA and PSO. Simulation results show that the proposed IEIEA algorithm outperforms, regarding network lifetime, SA and PSO with the same computational complexity. Furthermore, IEIEA has better robustness.

The rest of this paper is briefed as follows. In Section 2, we describe the full mathematical model of energy-efficient duty cycle design. In Section 3, IEIEA for energy-efficient duty cycle design as well as its detailed implementation is presented. In Section 4, we evaluate the performance of the proposed algorithm and show the simulation results and comparisons. We also analyze the characteristics of IEIEA. Finally, in Section 5, concluding remarks are presented.

2. System Model

In this section, we build the system model to demonstrate the energy-efficient duty cycle design problem concerning the constraints of the energy of a battery and sensing radius. In [14], the authors propose a set of inequalities for modeling the energy efficiency. Later in [15], the authors showed a similar model by a fewer number of sensors and targets. The model proposed in [16] is more flexible than the model proposed in [14]. In this work, a similar set of inequalities for maximizing the network lifetime is employed to model the energy-efficient duty cycle design in WSNs. Consider a WSN system with F sensors and E targets in the monitoring area. The monitoring relationship can be shown as

$$Q = \begin{bmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,E-1} & q_{1,E} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,E-1} & q_{2,E} \\ \vdots & & & & \vdots \\ q_{F-1,1} & q_{F-1,2} & \cdots & q_{F-1,E-1} & q_{F-1,E} \\ q_{F,1} & q_{F,2} & \cdots & q_{F,E-1} & q_{F,E} \end{bmatrix} \quad (1)$$

$$(q_{f,e} \in \{0, 1\}),$$

where $q_{f,e}$ is the monitoring relationship between the e_{th} target and f_{th} sensor; $q_{f,e} = 1$ means that the e_{th} target is within the monitoring range of the f_{th} sensor and $q_{f,e} = 0$ otherwise. Due to the limited battery capacity of the sensor

nodes, the maximum length of work time is assumed to K rounds. The duty cycle of the WSN can be shown as

$$D = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,F-1} & d_{1,F} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,F-1} & d_{2,F} \\ \vdots & & & & \vdots \\ d_{FK-1,1} & d_{FK-1,2} & \cdots & d_{FK-1,F-1} & d_{FK-1,F} \\ d_{FK,1} & d_{FK,2} & \cdots & d_{FK,F-1} & d_{FK,F} \end{bmatrix} \quad (2)$$

$$(d_{i,f} \in \{0, 1\}),$$

where $d_{i,f} = 1$ means that the i_{th} sensor node is active in round i and $d_{i,f} = 0$ otherwise. The monitoring cycle of the whole network can be shown as

$$DQ = \begin{bmatrix} \sum_{f=1}^F d_{1,f} q_{f,1} & \cdots & \sum_{f=1}^F d_{1,f} q_{f,E-1} & \sum_{f=1}^F d_{1,f} q_{f,E} \\ \sum_{f=1}^F d_{2,f} q_{f,1} & \cdots & \sum_{f=1}^F d_{2,f} q_{f,E-1} & \sum_{f=1}^F d_{2,f} q_{f,E} \\ \vdots & \ddots & \vdots & \vdots \\ \sum_{f=1}^F d_{FK-1,f} q_{f,1} & \cdots & \sum_{f=1}^F d_{FK-1,f} q_{f,E-1} & \sum_{f=1}^F d_{FK-1,f} q_{f,E} \\ \sum_{f=1}^F d_{FK,f} q_{f,1} & \cdots & \sum_{f=1}^F d_{FK,f} q_{f,E-1} & \sum_{f=1}^F d_{FK,f} q_{f,E} \end{bmatrix}, \quad (3)$$

where the element $\sum_{f=1}^F d_{j,f} q_{f,e} = 1$ is positive and represents the fact that the e_{th} target is monitored by at least one sensor node in round j and $\sum_{f=1}^F d_{j,f} q_{f,e} = 0$ otherwise. When the WSN maintains a full coverage in the j_{th} round, all the elements in the j_{th} row must be positive. Thus, from top to bottom, the previous row number of the first zero element is the network lifetime. Thus, the mathematical model of the duty cycle design problem can be shown as follows:

Objective is

$$f(D) = \text{zero_row}(DQ) - 1 \quad (4)$$

subject to

$$\sum_{i=1}^{FK} d_{i,f} \leq K, \quad f = 1 \cdots F, \quad (5)$$

where zero_row is the row number of the first zero element which occurs from row 1 to row FK in matrix. The constraint $\sum_{i=1}^{FK} d_{i,f} \leq K$ means that the maximum lifetime of node is K .

3. Energy-Efficient Duty Cycle Design Based on IEIEA for Wireless Sensor Networks

An evolutionary algorithm (EA) is a search method that mimics the mechanism of Darwinian's evolutionary theory such as genetic recombination and survival of the fittest. EA

is used for a large number of multiobjective optimization problems which is confirmed to be suitable for locating the best and near-optimal answers. The EA has good searching ability. However, the algorithm is often not suitable for large-scale complex problems. In many cases, the algorithm falls into local optima.

In recent times, elite computing and immune theory impressed strategic methods are getting evolved. They are confirmed to be very successful while looking for better solutions, targeting not only more efficient convergence but also solving combinatorial optimization problems. In this paper, we propose an improved elite immune evolutionary algorithm (IEIEA), which is a modification of EA, for energy-efficient duty cycle design in wireless sensor networks. In IEIEA, based on the concept and principles of elite computing and immune theory, new evolutionary theory techniques are getting developed, which can achieve an increased balance in exploitation of the solution space and receive better results compared with the standard EA. In such a hybrid technique, search ability of elite evolutionary strategy and immune clone operator are involved to generate faster and efficient convergence. It operates parallel exploration in complicated search regions.

Throughout each generation, IEIEA gets a different population out of the current population by using genetic operators, such as selection, crossover, and mutation. Additionally, suggested elite evolutionary strategy is using the elite idea, and the provided immune clone operator is using the immune idea to improve the global search capability of IEIEA.

3.1. Initialization. In IEIEA, a solution is represented by a chromosome (individual). We select a string as a chromosome to symbolize a solution to the energy-efficient duty cycle design problem. Each chromosome within the population can signify a group of randomly selected duty cycles. The energy-efficient duty cycle design problem is acquiring a chromosome within prospective solution regions making use of IEIEA. Therefore, if matrix (2) is arranged in rows, a feasible solution could be presented as a binary sequence. So we use the way of binary coding for each chromosome. Each gene in the chromosome is mapped to a binary number, and the gene value means that the sensor is in active or sleep mode. In other words, the binary chromosome is made up of Boolean variables implying whether the sensor is active or not.

The number of binary solution genes matches the number of sensor nodes and the maximum rounds. As an example, consider a system with three sensor nodes and four rounds. Binary coding representation with 36 binary numbers is accurate and reliable as it covers up to the entire solution space, and, besides, the string size is the number of sensor nodes square times the maximum rounds of the duty cycle. For ten sensor nodes with the highest lifetime of 20 rounds, the total string length is 2000. Here, the search space is the space of all matrices that satisfy (5). The total amount of all possible solutions is very large.

IEIEA solves optimizing challenges by using a population of a preset number, named the population size, of solutions. IEIEA maintains a number of such solutions. The

initialization design applied by this algorithm is as follows. The algorithm starts out by constructing an initial population of random candidate solutions. The original population was built randomly expecting to search globally to find some solutions. In each solution, a previously chosen number of random bits in the chromosome are set to 1. The rest bits in the chromosome are set to 0. By doing this, the initial population contains randomly produced chromosomes. Each solution in the population will be initialized to a binary number randomly selected from the uniform distribution over the set $[0, 1]$. A large number of low fitness solutions are constructed via utilizing the discrete random values in the initialization step. If a solution owns a low fitness value, random bits are determined from the set $[0, 1]$ to renew the solution. After obtaining the genes for each chromosome, the initial population is generated. The process is applied over and over unless all the chromosomes in the population get feasible. There are P_size individuals in the population of IEIEA. The population in the initialization process carries on randomly to ensure that the diversity is well kept.

3.2. Selection and Crossover. For every generation, children are generated by mixing the characteristics of two parent individuals. The selection operator picks a solution coming from the most recent population for the next population with possibility proportional to its fitness value.

Solutions in the current population are evaluated for their merit to survive in the following population. Every solution inside population is associated with a figure of benefit as well as a fitness value. Every time a set of two parent solutions are to be selected from a current population for developing an offspring by a crossover operation. The fitness value of each solution within the current population is calculated as the objective function by (4).

Picking out parents is based on random and selective approaches. At each iteration, couples of parents are selected by fitness, whereby every parent is chosen as the best of those randomly selected among the best candidates. In the selection step, fitter individuals are more regularly used to generate offspring than less fit individuals, which tends to raise the average fitness as the algorithm continues. Because the objective is to maximize the network lifetime, a solution string with a practically significant number of cycles must have comparatively high fitness value. So the solutions that are better will have a greater chance to be chosen. The selection operation mimics relatively neutral selection in real life.

The RWPS (roulette wheel proportionate selection) is used for choosing the solutions from the latest population. Better solutions have a relatively higher possibility to be selected for reproduction making use of the RWPS. The idea of RWPS is simple. The selection possibility of each solution is defined by the roulette wheel selection utilizing the linear scaling. By using the scaling, the probability of choosing the relevant solution for reproduction is proportional to the amount of scaled fitness.

In each step, two parents are picked out first according to the roulette wheel rule. This process will likely be applied repeatedly except when the whole set of child individuals in the population turns into feasible. In this way, all child strings

in the population were split into several pairs, with every pair establishing two children in the crossover operation. When all the parents were selected, their properties will be passed to their offspring. The previously mentioned selection process permits efficient constraint raising the quality of solutions by selecting the child chromosomes from high fitness districts.

In a crossover, the genes of the parent are replicated to its offspring. It signifies the mixing up of information from both parents to construct the children. We use a crossover operator to each of the selected pairs of parent solutions. In crossover operation, a couple of offspring are produced by the selected pair of parent solutions. In this way, two new children are manufactured by the random mixture of the properties of their parents. The crossover procedure is done between two chromosomes in their binary form. In a binary crossover, two new child chromosomes are created while using the interval between the two parents.

Standard crossover operator switches the segments of selected strings through the crossover points with a probability. The crossover place is randomly selected. One-point crossover specifies the point of crossover of the two parent chromosomes. Therefore, in single-point crossover, one crossover suggests that a child gets the left side of the binary string coming from the first parent and the right side coming from the second parent. For the second child, the crossover is accomplished by combining other parts of child chromosomes of the parents. The second child from this union obtains the complementary substances not taken by the first child, that is, receives the right section from the first parent and the left side from the second parent, having the same crossover location. Two-point crossover indicates that the first child gets both right and left sections from the first parent. Meanwhile, the first child also gets the middle section from the second parent. The crossover point is randomly chosen.

It is often known that uniform crossover works more efficiently at solving combinatorial optimization issues than just one-point or two-point crossover. In this case, we present a binary execution of uniform crossover. The crossover operation is done in the mating pool to build children. In uniform crossover, every one of these couples produces two children utilizing an adaptive number of crossovers. Binary strings are changed between the two parents to generate two children. Despite establishing the crossover point in the conventional crossover, we use the random binary number as the mask to form the binary uniform crossover.

In binary uniform crossover operation, a probability term is set to manage the rate of crossover. We generate a random decimal between 0 and 1. Then choose each gene with an equal chance from two child strings and recombine the genes to create a new offspring. Exchange positions in the two chromosomes are selected arbitrarily. Once the decimal is over the probability term, no change takes place, and the corresponding bit in solutions one and two is passed on to child one and child two, respectively. Otherwise, the crossover operation is carried out. If the bit in the mask is equal to one, the related bit in solutions one and two is distributed to child two and child one, respectively, if the crossover opportunity term is below the probability term. For this reason, in

a binary uniform crossover, each pair of parents has a probability term chance of using uniform crossover operator. This process should be implemented continually except in cases where all the child strings become feasible. All elements of genes handed down from the parents ought to keep the order as they are available in their parents. So, there have been child individuals soon after the crossover has been carried out.

Observe that since binary encoding is applied, the tiniest unit that is exchanged in binary uniform crossover could be a bit. It allows developing the offspring positioned according to the parents. Crossover creates a designed yet randomized exchange of genetic elements between strings. Hence the offspring contains the properties received from each of its parents. Moreover, using uniform crossover can mix the genes from two solutions with a more consistent way.

3.3. Mutation and Recombination. The content of this portion is kept in the mutation procedure. In IEIEA, random mutation is conducted to the offspring to prevent the premature convergence. This is executed by using random adjustments to a ratio of the children generated. In this paper, mutation operation is established by randomly picking out any chromosomes utilizing prespecified possibilities. In this way, plenty of random mutations are exercised to add some diversity to the population. For the duty cycle design problem, this means replacing some genes with new random figures. Selected genes in a single individual are randomly chosen to carry out the mutation.

In IEIEA, thinking of a sufficiently large, diverse initial population where zeros and ones are available for every bit within the chromosomes, full convergence could conceivably be obtained without having mutations at all because any practical solutions can be manufactured from some blend of the initial population. Nevertheless, the initial population dimension is limited, and the tiniest gene that may be changed in a crossover is binary. Recognize that mutations are an essential method for IEIEA; we need to apply a mutation operator to develop new solutions.

Subsequent mutations of the offspring apply diversity to the population and investigate different locations of the searching space. Offspring experience a constraint amount of mutations completed, and so the offspring are then scored for fitness. Each and every gene in chromosomes is mutated as mentioned above applying a probability, labeled as mutation probability. The chances that a certain gene is mutated are operated by the mutation rate. In this paper, an initial mutation rate of 5% means that each gene has a 5% possibility of being replaced with a new value. We employ reverse mutation in IEIEA. The reverse mutation is an important operation for preserving diversity in the population to be able to stay away from local optima. It will be shown in the simulation section that this mutation assists in reaching more efficient solutions.

For stopping the premature convergence of the algorithm, we employed dynamic mutation operators to diversify the population to be able to result in making the strings in the population dissimilar to each other. Dynamic mutation function is an iteration based function that determines the mutation rate. Smaller mutation levels indicate a more

local search, while larger mutation levels are more efficient for introducing diversity. In case the constructed variables converge at the low objective value, the degree of mutation can be higher. If a solution has a superior fitness value, a lower mutation rate is picked. In this way, the convergence length of IEIEA is reduced.

As in natural reproduction, there exists a likelihood of illegal mutation during the mutation process. The solutions that comply with constraints (5) are classified as legal strings; otherwise, they are recognized as illegal chromosomes. In illegal strings, a fully new random legal individual will be created to replace the illegal one.

Following using the crossover operator and the mutation operator, there have already been P_size chromosomes in the population that consisted of P_size parent strings and P_size offspring solutions. More advanced than the execution in EA, the selection and recombination processes in IEIEA are blended to manufacture a much better population. The recombination technique is a sort of uniform crossover which selects one offspring from two strings. After recombination, the new offspring fitness will be estimated depending on the fitness function.

3.4. Evaluation and Elite Clone Operation. In the examination procedure, the values of fitness for each string are assessed. Each chromosome in the population is linked to a figure of merit (fitness value) following the value of the function to be optimized. These individuals are scored, with the best performers probably going to be parents for the next iteration. Individuals with the greatest scores are implemented to be parents. In this paper, the aim is to increase the network lifetime. Within the current context, the fitness value of a chromosome would depend upon the entire duty cycle. The function values (fitness values) linked to the duty cycle are then assessed. The duty cycle design problem can be adapted swiftly to suit the full coverage constraints by mapping the duty cycle to the chromosomes and defining a fitness function to evaluate the potential fitness computation function for solution strings by the chromosome.

Here, we intend to identify the formulation of examining the level of chromosomes. To be able to calculate the level of each solution string in the population, formula (4) associates a fitness function. We determine fitness as the amount of network lifetime that can be shown by a combination of duty cycles; thus, fitness is an integer value from 0 to maximum rounds. The function value denoted by network lifetime is calculated by delivering binary individuals into the objective function. The solutions are then decoded into their matrix equivalent and examined for constraints violation. The fitness function is going to be scaled to avoid premature convergence. Through this definition, the fitness value is more effective when it is larger.

This procedure will be used for Max_gen times except for the solution that becomes feasible before Max_gen times. In this way, the algorithm ends till the fitness of the highest quality string converges to the maximum rounds. If it does not reach this value, the final chromosome distribution is adopted as the fittest individual (demonstrating the optimized network lifetime) following the Max_gen generations.

In this way, the IEIEA is terminated, and the fittest string in the population is taken as the final solution.

The individuals in the elite collection demonstrate the promising zones to seek out the suitable solution. The string in the elite group was harvested over all the generations. The population of the next generation includes the certain percentage of solutions from the elite group and most from offspring. The percentage between 2 kinds of individuals inside the new generation is defined by the probabilities of the alternative. The activity repeats for $Elite_num$ times, and, as a consequence, a new population is constructed.

The IEIEA commences with the building of an initial population, normally by spreading random binary numbers within investigated space. First, all individuals in a population are rated in lowering the order of fitness. The first 10% points are selected for cloning. Individuals with higher fitness will have more clones. The clones, not the original individual, then undergo the clonal mutation progress. The clonal mutation only performs on the cloned chromosomes to conserve the information. By doing this, the benefits of using the clonal mutation provide a neighboring exploration throughout the original chromosome, while the utilization of the chaotic mutation provides a global survey surrounding the string. Following that, strings are evaluated on the fitness function, and merely the best of each clone can successfully pass to the population, preserving an equal scale of the population. With this type of alternative, the diversity is conserved, and new fields of looking region tend to be quite possibly explained.

4. Simulation and Discussion

In this section, we compare the proposed IEIEA method with the simulated annealing algorithm (SA) and PSO method for energy-efficient duty cycle design in WSN. Simulations are performed to validate and test the performance of the proposed IEIEA approach to the energy-efficient duty cycle design problem in WSN. To examine the applicability for practical implementation, we evaluate the performance of the schemes on a PC with Intel Core i5, 4 G RAM, WIN-10 OS, and MATLAB software. The fitness function in Section 2 is used to evaluate the IEIEA efficiency. Then we generate the position of sensors and targets randomly specified within a 500×500 m square area. We assume that each sensor node has the same energy consumption per unit time. We placed 70, 80, 90, and 100 sensors, respectively. The sensing range of all sensor nodes is 300 meters. IEIEA has been used to solve this problem, and the generations obtained have been compared with SA and PSO. The maximum number of generations of IEIEA, SA, and PSO is 100. In all three methods, a population size of 100 individuals (solutions) is used. In SA, the initial temperature is set to 500 degrees and the annealing temperature coefficient is set to 0.98. In IEIEA, crossover and mutation probabilities are 0.9 and 0.05, respectively. In PSO, the dynamic range of the particle has been set to 0.4, the maximum velocity is 4, and the cognitive and social parameters are $c1 = c2 = 0.5$.

Figure 1 presents the network lifetime obtained by IEIEA, SA, and PSO on the energy-efficient duty cycle design problem with 100 sensors and 10, 20, 30, and 40 targets,

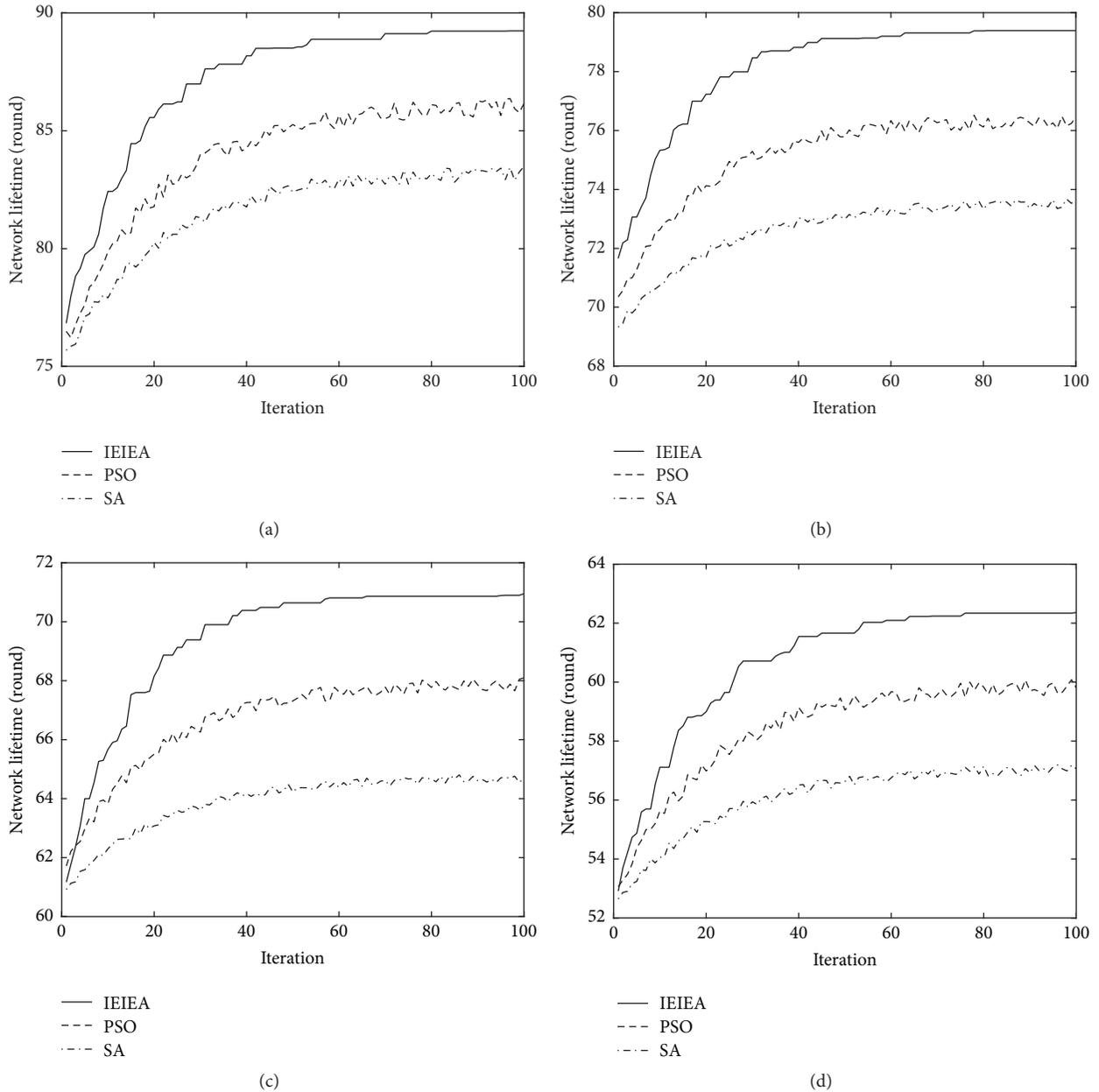


FIGURE 1: Network lifetime with 100 sensors and 10 targets (a), 20 targets (b), 30 targets (c), and 40 targets (d).

respectively. For each method, the fitness of the best individual at each generation is recorded. The results are the average of 30 time runs with three different methods to show the convergence difference. So there is a certain difference between the result of Figure 1 and the average value in Table 1.

From Figure 1, we can see that the network lifetime of IEIEA is better than those obtained by SA and PSO methods and it provides improvement of network lifetime after the 100 iterations. In the beginning, the value of network lifetime increased with the growth of the generations, as shown in the figure. After 50 iterations, IEIEA still has a fast convergence rate, but PSO and SA can only slowly converge to a lower value than IEIEA. As a result, diverse population leading to better solutions is maintained, so IEIEA is prevented from

TABLE 1: Network lifetime with different targets and 100 sensors.

Number of targets	IEIEA	PSO	SA
10	89.32	86.49	83.58
20	80.24	77.39	74.48
30	71.75	68.89	65.54
40	63.06	60.19	57.28

premature convergence. Over 100 iterations, IEIEA provides a longer network lifetime than PSO and SA, and IEIEA converges with a faster rate. Similar results can be obtained with 90 sensors in Figure 2.

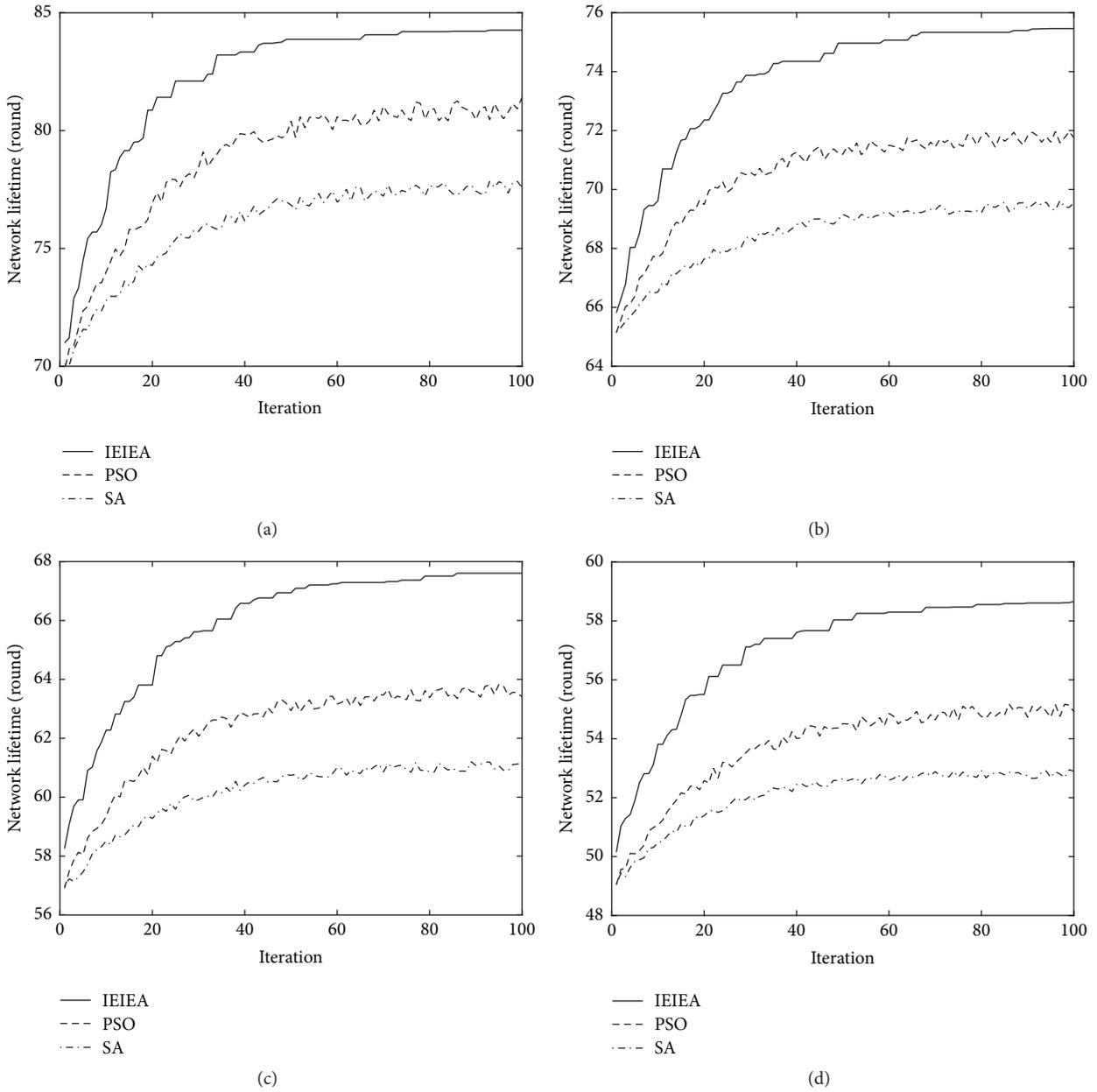


FIGURE 2: Network lifetime with 90 sensors and 10 targets (a), 20 targets (b), 30 targets (c), and 40 targets (d).

Figure 3 presents the obtained results of IEIEA, SA, and PSO approaches after 100 iterations with 100, 90, 80, and 70 sensors and 10, 20, 30, and 40 targets, respectively. All the results were averaged over 100 runs. Table 1 shows the results of Figure 3(a) with 100 sensor nodes and 10, 20, 30, and 40 targets, respectively. It can be seen that, after 100 iterations, the average network lifetime of IEIEA is 89.32, 80.24, 71.75, and 63.06 rounds, respectively. However, PSO cannot always obtain the optimal solution within the predefined iterations and the network lifetimes obtained by the PSO are 86.49, 77.39, 68.89, and 60.19, respectively. On the other hand, the SA shows quite slower convergence and the network lifetimes obtained by the SA are 83.58, 74.48, 65.54, and 57.28,

respectively, hence proving the superior reliability of IEIEA. Similar results can be obtained with 90, 80, and 70 sensors.

5. Conclusion

In this paper, we use an improved elite immune evolutionary algorithm (IEIEA) to solve the energy-efficient duty cycle design problem in WSN. In this work, we first propose a novel formulation of the objective function to maximize the network lifetime to satisfy the full coverage constraints. Simulations are conducted by using the IEIEA and the energy-efficient duty cycle design methods based on SA, PSO, and GA. Simulation results show that the proposed IEIEA

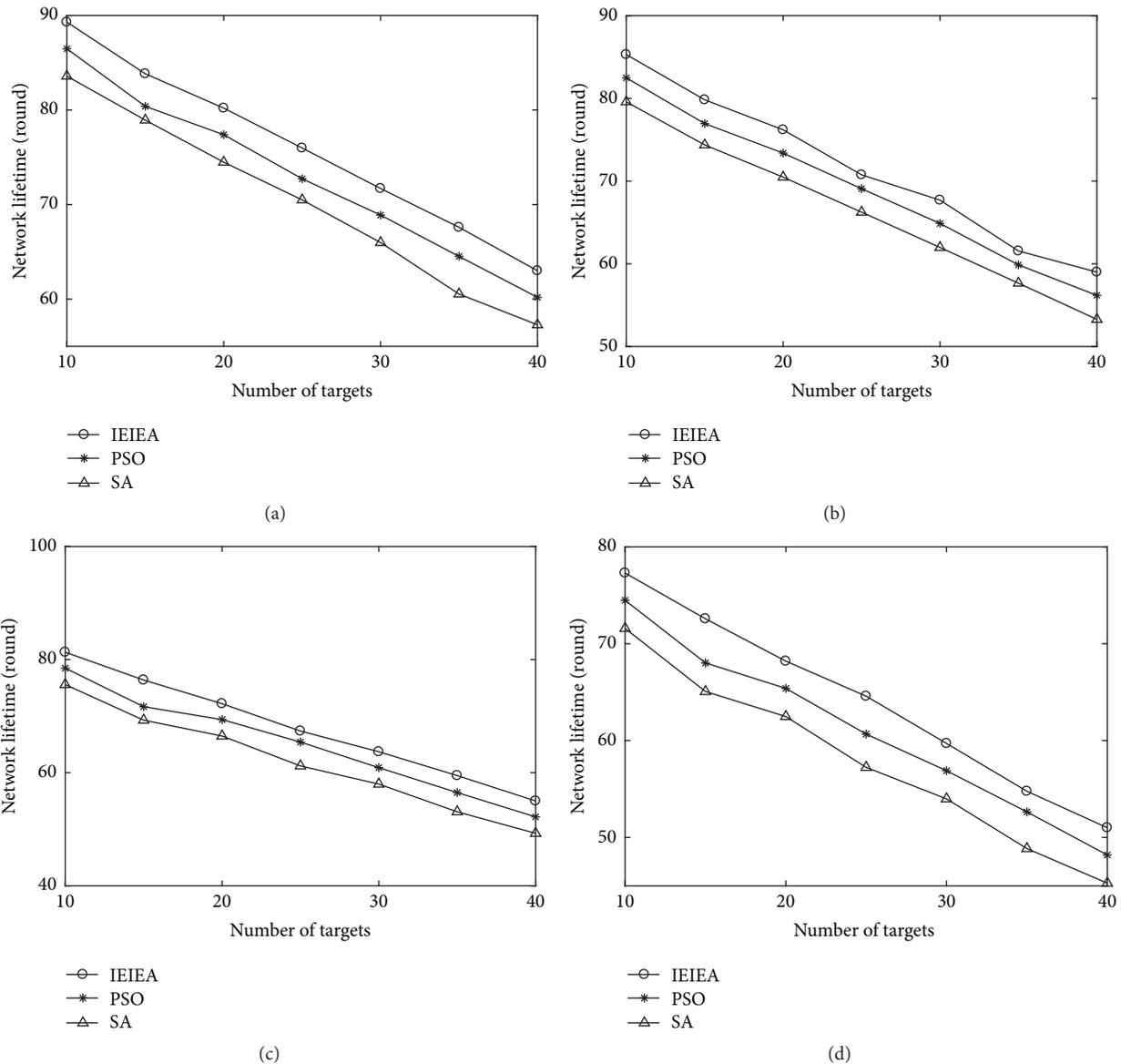


FIGURE 3: Network lifetime with different targets and 100 sensors (a), 90 sensors (b), 80 sensors (c), and 70 sensors (d).

provides longer network lifetime compared with previous SA, PSO, and GA. Besides that, the network lifetime of the proposed IEIEA is much higher than those of GA and SA. So the proposed method significantly enhanced the monitoring results.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China (no. 61662063) and High-Level Talent Research Project of Shihezi University (no. RCZX201530).

References

- [1] G. Y. Li, Z. Xu, C. Xiong et al., "Energy-efficient wireless communications: tutorial, survey, and open issues," *IEEE Wireless Communications*, vol. 18, no. 6, pp. 28–34, 2011.
- [2] X. Zhao, Y. Zhuang, and J. Wang, "Local adaptive transmit power assignment strategy for wireless sensor networks," *Journal of Central South University*, vol. 19, no. 7, pp. 1909–1920, 2012.
- [3] N. Sun, Y.-S. Jeong, and S.-H. Lee, "Energy efficient mechanism using flexible medium access control protocol for hybrid wireless sensor networks," *Journal of Central South University*, vol. 20, no. 8, pp. 2165–2174, 2013.
- [4] D. Feng, C. Jiang, G. Lim, L. J. Cimini Jr., G. Feng, and G. Y. Li, "A survey of energy-efficient wireless communications," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 167–178, 2013.

- [5] A. A. Kumar Somappa, K. Øvsthus, and L. M. Kristensen, "An industrial perspective on wireless sensor networks—a survey of requirements, protocols, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1391–1412, 2014.
- [6] D. Dong, X. Liao, K. Liu, Y. Liu, and W. Xu, "Distributed coverage in wireless ad hoc and sensor networks by topological graph approaches," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 61, no. 10, pp. 1417–1428, 2012.
- [7] W.-C. Ke, B.-H. Liu, and M.-J. Tsai, "Constructing a wireless sensor network to fully cover critical grids by deploying minimum sensors on grid points is NP-complete," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 56, no. 5, pp. 710–715, 2007.
- [8] K. Han, J. Luo, L. Xiang, M. Xiao, and L. Huang, "Achieving energy efficiency and reliability for data dissemination in duty-cycled WSNs," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1041–1052, 2015.
- [9] X. Hu, J. Zhang, Y. Yu et al., "Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 766–781, 2010.
- [10] E. R. Dosciatti, W. Godoy Junior, and A. Foronda, "TQ/PSO - A new scheduler to optimize the time frame with PSO in WiMAX networks," *IEEE Latin America Transactions*, vol. 13, no. 1, pp. 365–376, 2015.
- [11] T. Back, U. Hammel, and H.-P. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 3–17, 1997.
- [12] Z.-J. Zhang, J. Huang, and Y. Wei, "Frequent item sets mining from high-dimensional dataset based on a novel binary particle swarm optimization," *Journal of Central South University*, vol. 23, no. 7, pp. 1700–1708, 2016.
- [13] S. Sarafijanović and J.-Y. Le Boudec, "An artificial immune system approach with secondary response for misbehavior detection in mobile ad hoc networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 5, pp. 1076–1087, 2005.
- [14] Z. Lu, W. W. Li, and M. Pan, "Maximum lifetime scheduling for target coverage and data collection in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 2, pp. 714–727, 2015.
- [15] O. Demigha, W. Hidouci, and T. Ahmed, "On Energy efficiency in collaborative target tracking in wireless sensor network: a review," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1210–1222, 2013.
- [16] C.-L. Yang and K.-W. Chin, "Novel algorithms for complete targets coverage in energy harvesting wireless sensor networks," *IEEE Communications Letters*, vol. 18, no. 1, pp. 118–121, 2014.



Hindawi

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

