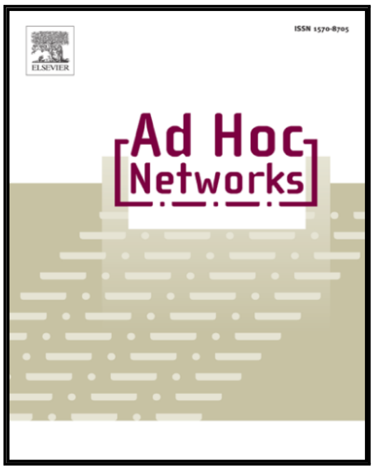


# Mobile wireless sensor network lifetime maximization by using evolutionary computing methods

Xiu Zhang , Xiaohui Lu , Xin Zhang

PII: S1570-8705(19)30906-0
DOI: https://doi.org/10.1016/j.adhoc.2020.102094
Reference: ADHOC 102094

To appear in: *Ad Hoc Networks*

Received date: 29 September 2019
Revised date: 8 January 2020
Accepted date: 8 February 2020

Please cite this article as: Xiu Zhang , Xiaohui Lu , Xin Zhang , Mobile wireless sensor network lifetime maximization by using evolutionary computing methods, *Ad Hoc Networks* (2020), doi: https://doi.org/10.1016/j.adhoc.2020.102094

# Mobile wireless sensor network lifetime maximization by using evolutionary computing methods

Xiu Zhang[a], Xiaohui Lu[a], Xin Zhang[a*1]

[a]Tianjin Key Laboratory of Wireless Mobile Communications and Power Transmission, Tianjin Normal University, Tianjin 300387, China

ARTICLE INFO

Article history:
Received
Revised
Keywords:
wireless sensor network
lifetime maximization
evolutionary computing
energy efficient
optimization

ABSTRACT

Due to the continuous development and progress of wireless communication technology and sensor network technology, wireless sensor networks (WSNs) have gradually become an attractive technology that facilitates people's lives. Due to the extensive use of WSNs, maximizing the lifetime of WSNs to obtain real-time and effective information has become a critical concern. This paper studies the life of mobile wireless sensor networks (MWSNs). MWSNs are a special type of WSN in that the sensor nodes are movable within a certain area. A system model is developed to prolong the network lifetime of MWSNs. This paper uses five evolutionary computing (EC) algorithms to develop the MWSN lifetime optimization model. Numerical simulations are performed to study the advantages and disadvantages of the five algorithms for solving the model. The comparison and discussion can provide advice for using EC algorithms to solve MWSN lifetime maximization problems.

## 1 Introduction

Due to the continuous development and progress of wireless communication technology, network technology, microprocessor technology and sensor network technology, WSNs have gradually become an attractive technology that facilitates people's lives [1]-[5]. Moreover, WSNs are a new way to acquire information through real-time monitoring of the environment. Because of their unique way of obtaining information, WSNs are widely used in various fields, such as military defense, biological medicine, smart home technology, industry and agriculture [2], [3]. As the capacity of battery of nodes is limited, the operational longevity of nodes is critical. The longevity of a WSN directly affects the overall performance of the network [4].

MWSNs are a special distributed network of many deployed sensor nodes that are movable within a monitoring area. MWSNs form a self-organizing network through wireless communication technology [5]. Unlike in static WSNs, the mobility of sensors or sink nodes in MWSNs causes network topology to change dynamically. Thus, compared to when designing static WSNs, more issues have to be addressed when designing mobile networks [4].

Recently, there have been studies on the lifetime of MWSNs. [6] studied maximizing the lifetime of MWSNs that contained mobile sink nodes. In [7], the exploration and exploitation trade-off was studied, and different methods were compared. According to this study, Thompson sampling and exponential weight algorithm action-section are robust methods for channel selection and transmission power control. In [8], the lifetime of WSNs was maximized by using a method for harvesting solar energy. In [9], a secure and efficient authentication protocol was proposed for multigateway wireless networks. In [10], scheduled transmissions from wireless networks were studied by using learning channel statistics. The proposed greedily constructed schedules show good scalability with the number of network links. In [11], prolonging the lifetime of WSNs was solved by proposed optimization formulations in the case of machine-to-machine communication. In [12], a distributed topology control game method was proposed for WSNs that have many sensor nodes but limited energy resources. In [13], a routing algorithm to reduce energy consumption and delay by MWSNs was proposed.

1 *Corresponding author: ecemark@tjnu.edu.cn (Xin Zhang).
*Email Addresses*: zhang210@126.com (Xiu Zhang[a]), lxh021895@163.com (Xiaohui Lu[a]), ecemark@tjnu.edu.cn (Xin Zhang[a])

The contributions of this paper are as follows:

(1) The objective of the MWSN model is to minimize the residual and consumed energies of all nodes. Minimizing residual energies is useful in preventing nodes from dying quickly. Minimizing consumed energies is useful in prolonging the lifetime of nodes. Both residual and consumed energies are combined to achieve the objective of the MWSN model.

(2) Unlike current studies, this paper uses EC algorithms to solve the MWSN model. Traditionally, MWSN models have to be relaxed to convex optimization problems [4], [14]-[19]. Then, problems can be solved by popular convex optimization solvers. This paper uses EC algorithms, which do not require the MWSN model to have a convex property.

(3) The properties of five EC algorithms are analyzed and discussed. The results provide useful hints for solving MWSN or WSN models.

This paper presents a lifetime model of an MWSN in Section 2. The paper introduces five EC methods in Section 3. Then, this paper analyzes the lifetime of wireless sensor networks under five EC algorithms in Section 4. According to the simulation experiment of the system model, the optimal EC algorithm is obtained under certain parameters so that the lifetime of the MWSN can be improved. Finally, Section 5 concludes the paper.

## 2 Lifetime model of the mobile wireless sensor network

It is assumed that the MWSN system consists of sensor nodes and sink nodes, as shown in Fig. 1. In the figure, the MWSN network consists of 2 sink nodes and 16 sensor nodes. Sink nodes are fixed and have a coverage area with a radius *Rc*. Sensor nodes can freely move with a certain degree of mobility. In cases where all sensor nodes are covered by sink nodes, the network becomes robust as all nodes are under the control of sink nodes. In cases where sink nodes may not cover all sensor nodes, ad hoc routing protocols are required. Sensor nodes not covered by sink nodes may construct a temporary multihop ad hoc network for communication.

To quantify the lifetime of the MWSN, it is assumed that the time interval consists of $K$ events. Each event consists of $F$ frames. Hence, there are $KF$ frames in the time interval. It is assumed that the network includes $Se$ sensor nodes and $Si$ sink nodes. The sensor nodes contain the same type of battery. The number of batteries in the sensor nodes is represented by $S(k)$. The amount of energy consumed by the nodes is represented by $B(k)$. A matrix of ratios $R(k)$ is used to denote the proportional energy available to the sensor nodes. Hence, the expected energy being consumed is $B(k)\circ R(k)$, where the symbol "$\circ$" represents the product of each element of matrix $B(k)$ and $R(k)$. Clearly, the level of consumption depends on the data rate, the distance, the channel, etc.

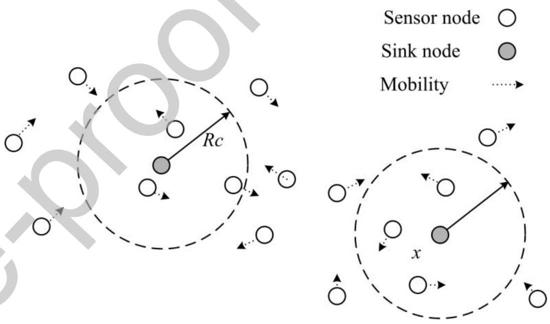


Fig 1. An example of a mobile wireless sensor network.

At event $k$, the residual energy of frame $(f+1)$ is a function of the residual energy of frame $f$:

$$S_{f+1}(k) = S_f(k) - B_f(k) \circ R_f(k) \quad . \qquad (1)$$

The lifetime model of the MWSN is:

$$\begin{aligned} \min \quad & \max\left\{\lambda_1 RE(k), \lambda_2 CE(k)\right\} \\ s.t. \quad & RE(k) = \left|S_f^*(k) - S_f(k)\right| \\ & CE(k) = \left|B_f^*(k) - B_f(k)\right| \\ & 0 \le R(k) \\ & \sum_{j=1}^{Se} \left(R(k)\right)_{ij} = 1, i = 1, 2, \cdots F \\ & S_{f+1}(k) = S_f(k) - B_f(k) \circ R_f(k) \end{aligned} \quad , (2)$$

where the objective is the weighted Tchebycheff metric of the residual energy $S(k)$ and consumed energy $B(k)$. In general, a weighted sum can be used to combine $S(k)$ and $B(k)$. However, a weighted sum technique may have difficulty solving nonconcave problems [20]. The advantage of the weighted Tchebycheff technique is that it is able to overcome the issue that cannot be solved by the weighted sum technique. Thus, the weighted

Tchebycheff technique is used in the lifetime model.

In (2), $S_f^*(k)$ and $B_f^*(k)$ are reference points of the weighted Tchebycheff metric. Moreover, $\lambda_1$ and $\lambda_2$ are the weights for $S(k)$ and $B(k)$, respectively. Different values of $\lambda_1$ and $\lambda_2$ result in different solutions of (2). In the model, $S(k)$ and $R(k)$ are parameters to be determined. $B(k)$ can be computed by $S(k)$ and $R(k)$ [3]:

$$B_f(k) = S_f(k) \circ \frac{1}{R_f(k)} \quad . \tag{3}$$

Based on model (2), the determination of $R(k)$ is the basis for the problem. Proper values of $R(k)$ indicate a good allocation of energies. Such an allocation, in turn, would help prolong the lifetime of the MWSN. In this paper, the $R(k)$ is determined by using EC algorithms.

## 3 Evolutionary computing algorithms

EC methods have been applied to many fields [21]-[25]. Five evolutionary computing methods are used to solve the lifetime improvement problem: genetic algorithms (GAs) [26], differential evolution (DE) [27], particle swarm optimization (PSO) [28], artificial bee colonies (ABCs) [29] and neighborhood field optimization (NFO) [30], [31]. These methods are explained in this section.

### 3.1 Genetic algorithm

GAs were first proposed by Holland of Michigan University; Holland was inspired by biological evolution. GAs are based on a population, and the process for dealing with the population mainly includes initialization and evolution. The population in GAs consists of individuals, and evolution includes three genetic operations: selection, hybridization or crossover, and mutation. GAs can perform global optimization in data space and highly converge; however, the disadvantage of GAs is that these algorithms cannot use local information effectively; thus, it takes a long time to converge to an optimal point [32].

By using a paradigm, a GA can be instantiated in different ways. For example, selection methods include roulette wheel selection, rank-based selection, and tournament-based selection. Crossover methods include uniform crossover and arithmetic crossover. An instance of a GA can be any combination of selection, crossover and mutation operations.

In this paper, the GA algorithm consists of a roulette wheel selection, a uniform crossover and uniform mutation operations. A uniform crossover is:

$$x_{i,j}^{t+1} = \begin{cases} x_{p1,j}^t, & if\ r_1 < p_c \\ x_{p2,j}^t, & otherwise \end{cases}, \ j = 1, 2, \cdots, D \ , \tag{4}$$

where $t$ is the generation counter, $r_1$ is a random number between 0 and 1, $x_{p1}^t$ and $x_{p2}^t$ are two parents of $x_i^t$, and $P_c$ is the crossover probability or crossover rate (which is an algorithmic parameter predefined by users).

A uniform mutation is:

$$x_{i,j}^{t+1} := \begin{cases} x_{i,j}^{t+1}, & if\ r_2 < p_m \\ x_{i,j}^{\min} + r_3\left(x_{i,j}^{\max} - x_{i,j}^{\min}\right), & otherwise \end{cases}, \tag{5}$$

where $P_m$ is the mutation probability (which is an algorithmic parameter predefined by users) and $x_i^{\max}$ and $x_i^{\min}$ are the upper and lower bound of variable $x_i$.

### 3.2 Differential evolution

A DE algorithm is an efficient, simple and fast global search evolutionary algorithm proposed by Storn and Price. The biggest difference between DE and other algorithms lies in the use of a differential mutation operator, which has the characteristics of search direction and search step-size adaptivity. DE has the advantages of having a simple structure, being easy to use and being strongly robust.

In this paper, the variant of the DE algorithm used is DE/rand/1/binomial, where “rand” means randomly choose mutation solutions, “1” means one differential vector in the mutation equation, and “binomial” means binomial crossover. In general, the survivor selection of the DE is a “(1+1)” selection. That is, a new solution competes with its corresponding parent solution, and the one that has the best fitness survives.

The mutation operation of the DE algorithm is:

$$x_i^{t+1} = x_{r1}^t + F \times \left(x_{r2}^t - x_{r3}^t\right) \ , \tag{6}$$

where $x_{r1}^t$, $x_{r2}^t$ and $x_{r3}^t$ are randomly chosen solutions from the population, and $F$ is the scale factor (which is an algorithmic parameter predefined by users) [27].

For $j = 1, 2, \cdots, D$, the binomial crossover of the DE algorithm is:

$$x_{i,j}^{t+1} := \begin{cases} x_{i,j}^{t+1}, & if\ r_2 < p_r\ or\ j = j_{r4} \\ x_{i,j}^{t}, & otherwise \end{cases}, \quad (7)$$

where integer $j_{r4}$ is randomly chosen to make sure that a variable is changed after the crossover operation.

### 3.3 Particle swarm optimization

A PSO algorithm is a population-based stochastic optimization algorithm proposed by Eberhart and Kennedy in 1995. PSO is inspired by the social behavior of birds or fish groups. The system is initialized by the population of random solutions, and the optimal solution is searched through the update generation. However, unlike GAs, PSO has no crossover, mutation or other evolutionary operators and seeks global optimization by following current optimal value. Moreover, PSO has become a popular modern optimization method because of its few parameters, simple operation and fast convergence.

In this paper, the PSO algorithm consists of velocity updating and particle position updating. Velocity updating is:

$$v_i^{t+1} = \varphi \times \left( v_i^t + c1 r_1 d_i^{t,p} + c2 r_2 d_i^{t,g} \right), \quad (8)$$

where $\varphi$ is inertia weight, $c1$ and $c2$ are accelerating factors (which need to be predefined by users [28]), $d_i^{t,p}$ is the best personal solution found by particle $I$, and $d_i^{t,g}$ is the best-so-far solution found by all particles.

Particle position updating is:

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \quad (9)$$

### 3.4 Artificial bee colony

An ABC algorithm is a kind of biological intelligent optimization algorithm that simulates the intelligent searching behavior of bee colonies. This algorithm is a new global optimization algorithm. ABC algorithms have few control parameters, are easy to implement and are simple to calculate. Hence, these algorithms are widely used in various fields. An ABC can be combined with other algorithms, such as PSO, DE, distributed thinking, a local search operator and a population diversity strategy, to improve the overall optimization performance of the algorithm. ABC algorithms and their application have important academic significance and practical value.

In this paper, the ABC algorithm consists of the employed bee stage, onlooker bee stage and scout stage. The scout stage randomly recreates new solutions by replacing old solutions. The employed bee and onlooker bee stage have a similar updating equation. For $j = 1, 2, \cdots, D$, the equation is:

$$v_{i,j}^{t+1} = x_{i,j}^t + \varphi_{i,j} \times \left( x_{i,j}^t - x_{k,j}^t \right), \quad (10)$$

where $\varphi_{i,j}$ is a random number (based on a uniform distribution) between -1 and 1 [29] and $x_{k,j}^t$ is a randomly chosen neighbor of $x_{i,j}^t$.

### 3.5 Neighborhood field optimization

NFO is a recently proposed EC method. By constructing a high-quality neighborhood structure, NFO algorithms can perform well. EC algorithms, such as GAs and PSO, have good global search ability. Although it is hard to trap these algorithms in local optima, many iterations are necessary to converge to desirable solutions. NFO algorithms differ from GAs and PSO in that NFO emphasizes the effect of neighborhood structure. Nearest neighbors are regarded as an approximate gradient to cooperatively guide the local search of the algorithm. Thus, NFO is more like a local search algorithm. Due to its population-based structure, NFO has good explorative ability in the initial evolutionary stage. Once the algorithm starts to converge to a small region, a differential of neighbors may approximate gradient information. The algorithm may have good exploitative ability.

In this paper, the NFO algorithm consists of neighborhood localization, mutation and a crossover. The crossover is a binomial crossover, as described in Section 3.2. Neighborhood localization involves choosing the neighbors for the solutions in the population. The mutation is based on the differentials of neighbors. The updating equation is:

$$x_i^{t+1} = x_i^t + \alpha r_1 ds_i^t + \alpha r_2 di_i^t \ , \quad (11)$$

$$ds_i^t = xc_i^t - x_i^t \ , \quad (12)$$

$$di_i^t = xc_i^t - xw_i^t \ , \quad (13)$$

where $\alpha$ is the learning factor; $r_1$ and $r_2$ are random numbers between 0 and 1; $xc_i^t$ and $xw_i^t$ are the superior and inferior neighbors, respectively, of $x_i^t$; and $ds_i^t$ and $di_i^t$ are the search directions toward superior and inferior neighbors, respectively, of $x_i^t$.

EC algorithms can be classified into two categories: evolutionary algorithms and swarm intelligence approaches. The two categories have different paradigm philosophies. GAs and DE algorithms can be classified as evolutionary algorithms. PSO and ABCs can be classified as swarm intelligence approaches. An intuitive method of classification is a crossover operation. NFO uses a binomial crossover; hence, NFO algorithms are evolutionary algorithms. Despite the differences between the two categories, methods wherein algorithms from both categories interact and cooperate have become popular. This study aims to apply EC algorithms to WSNs and promote the usage of EC algorithms in wireless networks.

The procedures of the optimization framework are shown in Algorithm 1. EC algorithms are used in line 4. As Algorithm 1 shows, various EC algorithms can be interfaced with the framework. EC algorithms are responsible for determining $R(k)$ such that the objective of (2) is minimized under the constraints.

| Line | Algorithm 1: procedures for lifetime optimization of MWSN |
|---|---|
| 1 | Input: MWSN parameters $S_e$, $S_i$, threshold for live or dead, $K$, $F$; |
| 2 | EC algorithm parameters $P_s$, $P_c$, $P_m$, termination condition. |
| 3 | Output: optimal solution |
| 4 | For each event $k$ |
| 5 | For each frame $f$ |
| 6 | Solve lifetime model by using an EC algorithm to obtain $S_{f+1}(k)$. |
| 7 | End For |
| 8 | Compute consumption energy $B(k)$. |
| 9 | Allocate time slots based on $R(k)$. |
| 10 | Update node energies and network status of MWSN. |
| 11 | End For |

## 4 Simulation experiment

The simulation experiment consists of three parts. First, the lifetime model at different mobilities is tested and analyzed. Second, the lifetime model with different weights in the objective function is tested and analyzed. Third, the impact of network density is tested and analyzed. Five EC algorithms are used to solve the problems for the three parts.

The settings of the MWSN are as follows: The number of sensor nodes *Se* is 10. The number of sink nodes *Si* is 1. The initial energy of the sensor nodes is 10. The threshold of the sensor nodes, live or dead, is 0.05. Sensor nodes whose battery level is lower than 0.05 are considered to be dead nodes.

The parameter settings of the EC algorithms are given in Table 1. The maximum number of function evaluations (MFE) is 1000. To have a fair comparison, no two population sizes of the algorithms are set to the same number. The settings for each algorithm follow the associated literature [26]-[30]. By using such settings, it is expected that each algorithm can function without harming its original design philosophy. To solve a problem, each EC algorithm is independently executed 30 times to obtain an average performance.

### 4.1 Optimizing lifetime at different mobilities

First, the lifetime of the MWSN at different mobilities is studied. MWSN sensor nodes are allowed to move with a certain velocity. If sensor nodes move fast, then frequent optimization is required to refresh the network topology and allocation strategy; otherwise, optimization can be performed less frequently to save computational time and resources. The effect of EC methods is studied when frame $F$=1, 5 and 10. $F$=1 means frequent optimization under the condition that sensors have a high moving speed. $F$=5 means a moderate moving speed. $F$=10 means less frequent optimization under the condition that sensors have a low moving speed.

The results are shown in Table 2. This table shows the mean and standard deviation (std) values of the objective function. The results of Table 2 are based on the settings $\lambda_1$=1 and $\lambda_2$=0. As Table 2 shows, when $K$=400 and $F$=1, the average life of the five EC algorithms is relatively high, and the life deviation is relatively low. When frame $F$ increases from 1 to 10, the average lifetime gradually decreases. However, the decreases in the lifetime of the five EC methods are less than 6.17%, which is very small. Therefore, EC algorithms show robust performance for the MWSN at different mobilities.

As Table 3 shows, when $K$=400 and $F$=1, the average lifetime of the five EC algorithms is relatively high, and the life deviation is relatively low. The results of Table 3 are based on the settings $\lambda_1=0$ and $\lambda_2=1$. Although the average lifetime decreases when $F$ increases from 1 to 10, the lifetime decreases for the five EC algorithms are less than 4.51%. Based on the results of Table 2 and Table 3, the std values are approximately the same. Therefore, the performance of the EC algorithms is stable for prolonging the lifetime of the MWSN. In a comparison of the results of Table 2 and Table 3, the average lifetime of the MWSN when $\lambda_1=0$ and $\lambda_2=1$ is longer than that of the MWSN when $\lambda_1=1$ and $\lambda_2=0$. Therefore, optimizing $S(k)$ or $B(k)$ causes a different effect.

Table 1
Summary of parameters of evolutionary computing methods.

| Method | Parameters |
|---|---|
| GA | $Ps$ = 50, $Pc$ = 0.8, $Pm$ = 0.05 |
| DE | $Ps$ = 50, $Sf$ = 0.5, $Pc$ = 0.9 |
| PSO | $Ps$ = 40, $w$ = 0.7298, $c1$=$c2$ = 2.05 |
| ABC | $Ps$ = 20, *limit* = 20 |
| NFO | $Ps$ = 20, $F$ = 1.3, $Pc$ = 0.1 |

Table 2
Summary of the results obtained by evolutionary computing methods after 30 repeats for the MWSN when $\lambda_1$=1 and $\lambda_2$=0

| *KF* | *K*=400, *F*=1 | | *K*=80, *F*=5 | | *K*=40, *F*=10 | |
|---|---|---|---|---|---|---|
| Methods | Mean | Std | Mean | Std | Mean | Std |
| GA | 232.97 | 20.33 | 231.77 | 20.29 | 225.03 | 19.94 |
| DE | 222.20 | 16.54 | 220.30 | 16.10 | 212.67 | 17.04 |
| PSO | 198.00 | 16.99 | 188.43 | 16.81 | 185.80 | 18.39 |
| ABC | 224.27 | 14.83 | 222.60 | 20.20 | 214.07 | 16.10 |
| NFO | 226.83 | 20.58 | 224.87 | 20.03 | 218.87 | 20.69 |

Table 3
Summary of the results obtained by evolutionary computing methods after 30 repeats for the MWSN when $\lambda_1$=0 and $\lambda_2$=1

| *KF* | *K*=400, *F*=1 | | *K*=80, *F*=5 | | *K*=40, *F*=10 | |
|---|---|---|---|---|---|---|
| Methods | Mean | Std | Mean | Std | Mean | Std |
| GA | 242.77 | 22.19 | 240.13 | 21.79 | 234.10 | 20.16 |
| DE | 226.47 | 15.51 | 222.73 | 15.41 | 216.27 | 17.94 |
| PSO | 181.73 | 18.88 | 175.40 | 17.96 | 174.37 | 17.79 |
| ABC | 232.70 | 17.07 | 226.10 | 22.60 | 224.00 | 19.95 |
| NFO | 234.97 | 21.61 | 233.17 | 22.19 | 227.13 | 22.00 |

Table 4
Combinations of weights $\lambda_1$ and $\lambda_2$ in the lifetime model

| Weight | $\lambda_1$ | $\lambda_2$ |
|---|---|---|
| Combination1 | 1 | 0 |
| Combination2 | 0.2 | 0.8 |
| Combination3 | 0.4 | 0.6 |
| Combination4 | 0.6 | 0.4 |
| Combination5 | 0.8 | 0.2 |
| Combination6 | 0 | 1 |

### 4.2 Optimizing lifetime with different weights in the objective function

As the objective function of (2) is a weighted Tchebycheff of the residual and consumed energies, weights may affect the performance of the MWSN. The effect of the EC algorithms in optimizing the lifetime model with different weights in the objective function is studied. The weight values are given in Table 4. In the table, the $\lambda_1$ value gradually decreases from 1 to 0, while the $\lambda_2$ value gradually increases from 0 to 1.

The results are shown in Table 5. The structure of Table 5 is similar to that of Table 2. The mean and std values of the results obtained after 30 executions of the algorithm are given.

As Table 5 shows, for combination3 ($\lambda_1$=0.4 and $\lambda_2$=0.6), the average life span of the five EC algorithms is relatively long, and the life deviation is relatively low. Among the six combinations, combination3 has the longest lifetime. After a change in the weight value of the same evolutionary algorithm , the lifetime of PSO increased by 8.28%; additionally, weight had the greatest influence on the lifetime of PSO. Similarly, the lifetime of DE increased by 2.19%, and the weight of DE had the least influence on the lifetime of DE. Therefore, different combination of weights can affect lifetime. Therefore, it is necessary to choose the proper combination based on the target of practical applications.

Box plots of the lifetimes obtained by the algorithms after 30 executions are shown in Fig. 2 and Fig. 3. In Fig. 2, the results are for the MWSN when using combination1, while the results in Fig. 3 are for the MWSN when using combination2. The other combinations are not given to save space. Both Fig. 2 and Fig. 3 show that the GA attains the longest lifetime, NFO attains the second longest, and PSO attains the shortest lifetime. Moreover, the ABC, DE, the GA and NFO have similar median lifetime values.

The Mann-Whitney U test (U test) is used to test the significance of the five EC algorithms. As the GA achieves the longest lifetime in optimizing the MWSN by using combination1, the GA is used as the reference algorithm in the U test. A significance level of 0.05 is set in the U test. According to the U test, the PSO and GA algorithms are significantly different at a given significance level. The *p*-value is 2.47e-8. The *p*-values for the ABC, DE and NFO are 0.11, 0.06, and 0.25, respectively. Based on the U test, the GA, ABC, DE and NFO algorithms are not significantly different with respect to solving the MWSN lifetime problem. The results of the U test are similar to the results of the five EC algorithms for optimizing the MWSN by using combination2.

Table 5
Summary of the results obtained by the five EC methods for the MWSN model at different weights

| Weight | Combination1 | | Combination2 | | Combination3 | |
|---|---|---|---|---|---|---|
| Method | Mean | Std | Mean | Std | Mean | Std |
| GA | 232.97 | 20.33 | 242.97 | 22.07 | 245.63 | 22.19 |
| DE | 222.20 | 16.54 | 226.83 | 15.50 | 227.17 | 15.78 |
| PSO | 198.00 | 16.99 | 181.73 | 18.88 | 181.60 | 18.87 |
| ABC | 224.27 | 14.83 | 231.03 | 18.75 | 231.23 | 19.87 |
| NFO | 226.83 | 20.58 | 235.07 | 21.52 | 237.83 | 21.26 |
| Weight | Combination4 | | Combination5 | | Combination6 | |
| GA | 232.97 | 20.33 | 232.97 | 20.33 | 242.77 | 22.19 |
| DE | 222.20 | 16.54 | 222.20 | 16.54 | 226.47 | 15.51 |
| PSO | 198.00 | 17.00 | 198.00 | 17.00 | 181.73 | 18.88 |
| ABC | 226.47 | 21.50 | 217.50 | 18.90 | 232.70 | 17.07 |
| NFO | 226.83 | 20.58 | 226.83 | 20.58 | 234.97 | 21.61 |

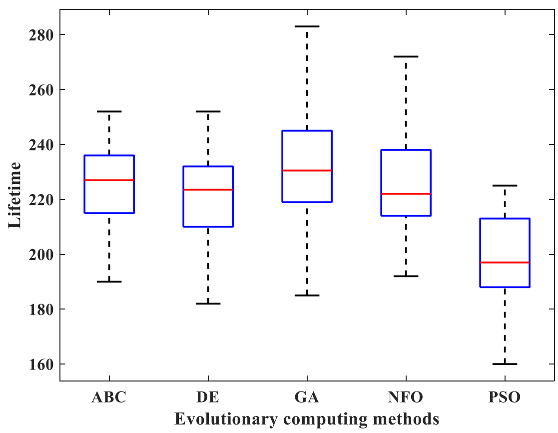


Fig 2. Box plot of lifetimes obtained by the five EC algorithms for the MWSN when $\lambda_1$=1 and $\lambda_2$=0.

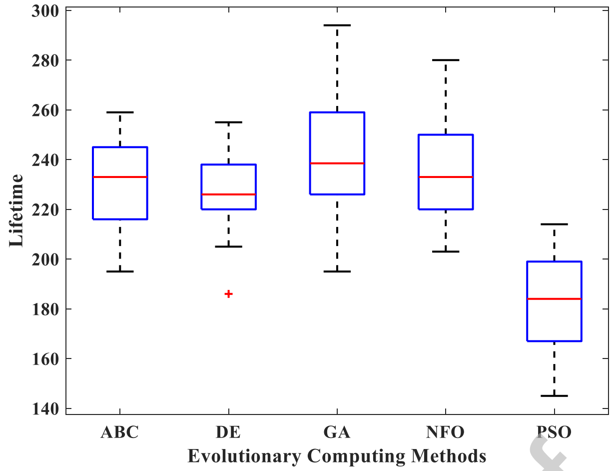


Fig 3. Box plot of lifetimes obtained by the five EC algorithms for the MWSN when $\lambda_1$=0.2 and $\lambda_2$=0.8.

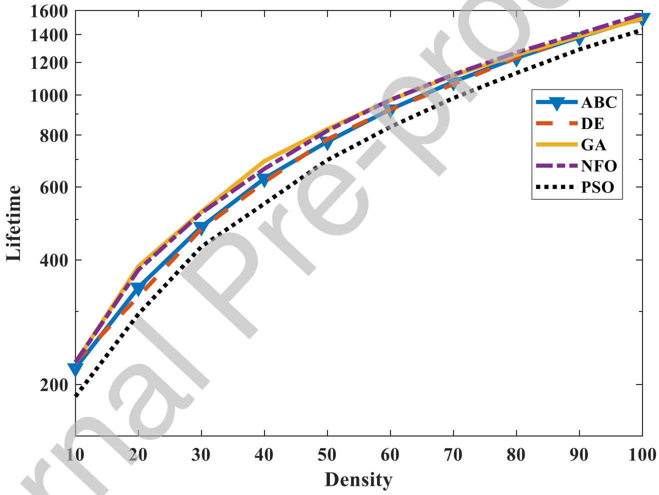


Fig 4. Lifetime obtained by the five EC algorithms for the MWSN at different densities.

4.3 Optimizing lifetime at different densities

The effect of the EC methods on the MWSN at different densities is studied. Network density can be changed by the number of sensor nodes. From *Se*=10 to *Se* =100, the step length of the algorithm is 10; the different densities are used to test network performance. The results are shown in Fig. 4.

When the density increases from 10 to 100, the lifetime achieved by the five EC algorithms for the MWSN increases quickly, as shown in Fig. 4. Among them, the lifetime achieved by the NFO algorithm is increased by 1342.70; additionally, the change of density has the greatest impact on the NFO algorithm. The PSO algorithm, which increased the lifetime of the MWSN by 1250.80, has the least lifetime gain. Therefore, the change in density is very important to improving the lifetime of the MWSN. When the density *Se=100*, the lifetime achieved by the NFO algorithm is the maximum, and the lifetime achieved by the PSO algorithm is the minimum. Moreover, Fig. 4 shows that the curve of the PSO algorithm lies below that of the other algorithms. Therefore, the PSO algorithm at different network densities performs worse than the other algorithms. The curves of the ABC and DE algorithms are very close to each other with an *Se* from 10 to100. Therefore, both algorithms have similar performance with respect to solving the lifetime problem. The curves of the GA and NFO algorithms are very close to each other with an *Se* from 10 to 100. Therefore, both the GA and NFO algorithms have similar performance with respect to solving the lifetime problem. Furthermore, when the number of nodes is fewer than 60 (i.e., $Se \leq 60$), the GA algorithm (compared with the other algorithms) finds the best solution for MWSN lifetime. When the number of nodes is more than 60 and fewer than 100 (i.e., $60 \leq Se \leq 100$), the NFO algorithm (compared with the other four algorithms) finds the best solution for MWSN lifetime. Thus, concerning the lifetime optimization of the MWSN, the GA algorithm is suitable for low-density networks, while the NFO algorithm is suitable for high-density networks.

## 5 Conclusion

MWSNs are special distributed networks consisting of many sensor nodes. In MWSNs, unlike in static WSNs, sensors or sink nodes have mobility, thus causing the network topology to change dynamically. In the design of MWSNs, more issues have to be addressed than in the design of static WSNs. Thus, lifetime maximization is an important problem in MWSNs.

This paper presents a new mathematical model for MWSNs. The objective of the MWSN model is to minimize the residual and consumed energies of all nodes. Minimizing residual energies is useful in preventing nodes from dying quickly. Minimizing consumed energies is useful in prolonging the lifetime of nodes. Both residual and consumed energies are combined to achieve the objective of the MWSN model by using a weighted Tchebycheff technique.

Unlike previous works in the literature, this paper uses EC algorithms to solve the MWSN model. Traditionally, MWSN models have to be relaxed to convex optimization problems. Then, problems could be solved by popular convex optimization solvers. This paper uses EC algorithms, which do not require the MWSN model to be convex. Five algorithms are used in the study.

The properties of the five EC algorithms for solving the MWSN model are analyzed and discussed. The GA and NFO perform the best and second best, respectively, for solving the network at different mobilities. The GA and NFO also perform the best and the second best, respectively, for solving the network with different weights of the objective function. Moreover, the PSO algorithm is significantly worse than the GA, the ABC, DE and NFO based on the U test, while the GA, ABC, DE and NFO algorithms are not significantly different with respect to solving the MWSN lifetime problem. Furthermore, for the lifetime optimization of the MWSN, the GA algorithm is suitable for low-density networks, while the NFO algorithm is suitable for high-density networks. The reported results provide useful hints for solving MWSN or WSN models.

## Declaration of Conflicting interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This research was supported in part by the National Natural Science Foundation of China (Project No. 61601329, 61603275, 61704122, 61701345, 61801327), the Natural Science Foundation of Tianjin (Project No. 18JCZDJC31900), and the Tianjin Higher Education Creative Team Funds Program.

## References

[1] Xiuwen Fu, Haiqing Yao, Yongsheng Yang, Cascading failures in wireless sensor networks with load redistribution of links and nodes, Ad Hoc Networks 93(2019) Article No 101900.
[2] Xin Liu, Min Jia, Xueyan Zhang, A Novel Multichannel Internet of Things Based on Dynamic Spectrum Sharing in 5G Communication, IEEE Internet of Things Journal 6 (2019) 5962-5970.
[3] X. Lu, H. Wang, X. Wang, On kalman smoothing for wireless sensor networks systems with multiplicative noises, Journal of Applied Mathematics 2012 (2012) Article No 717504.
[4] D. A. Guimaraes, L. J. Sakai, A. M. Alberti, R. A. A. de Souza, Increasing the Lifetime of Mobile WSNs via Dynamic Optimization of Sensor Node Communication Activity, Sensors 16 (2016) Article No 1536.
[5] Xin Liu, Xueyan Zhang, Rate and Energy Efficiency Improvements for 5G-Based IoT With Simultaneous Transfer, IEEE Internet of Things Journal 6 (2019) 5971-5980.
[6] F. Tashtarian, et al., On maximizing the lifetime of wireless sensor networks in event-driven applications with mobile sinks, IEEE Trans. Veh. Technol., 64 (2015) 3177-3189.
[7] Francesc Wilhelmi, Cristina Cano, Gergely Neu, Boris Bellalta, Anders Jonsson, Sergio Barrachina-Muñoz, Collaborative Spatial Reuse in wireless networks via selfish Multi-Armed Bandits, Ad Hoc Networks 88(2019) 129-141.
[8] Himanshu Sharma, Ahteshamul Haque, Zainul Abdin Jaffery, Maximization of wireless sensor network lifetime using solar energy harvesting for smart agriculture monitoring, Ad Hoc Networks 94(2019) Article No 101966.
[9] Hua Guo, Ya Gao, Tongge Xu, Xiyong Zhang, Jianfeng Ye, A secure and efficient three-factor multi-gateway

authentication protocol for wireless sensor networks, Ad Hoc Networks 95(2019) Article No 101965.

[10] Thomas Stahlbuhk, Brooke Shrader, Eytan Modiano, Learning algorithms for scheduling in wireless networks with unknown channel statistics, Ad Hoc Networks 85(2019) 131-144.

[11] Emma Fitzgerald, Michał Pióro, Artur Tomaszewski, Network lifetime maximization in wireless mesh networks for machine-to-machine communication, Ad Hoc Networks 95(2019) Article No 101987.

[12] Xiaochen Hao, Liyuan Wang, Ning Yao, Dehua Geng, Bai Chen, Topology control game algorithm based on Markov lifetime prediction model for wireless sensor network, Ad Hoc Networks 78(2018) 13-23.

[13] Ramin Yarinezhad, Reducing delay and prolonging the lifetime of wireless sensor network using efficient routing protocol based on mobile sink and virtual infrastructure, Ad Hoc Networks 84(2019) 42-55.

[14] J. Tang, G. He, and L. Fang, A new non-interior continuation method for second-order cone programming, J. Numer. Math., 21(2013) 301-323.

[15] Changhe Liu, Youlin Shang, Ping Han, A new infeasible-interior-point algorithm for linear programming over symmetric cones, Acta Math. Appl. Sinica, English Ser. 33(2017) 771-788.

[16] J. Zhu, K. Li, and B. Hao, Restoration of remote sensing images based on nonconvex constrained high-order total variation regularization, Journal of Applied Remote Sensing 13(2019) Article No 022006.

[17] Cong-Ying Han, Fang-Ying Zheng, Tian-De Guo, Guoping He, Parallel algorithms for large-scale linearly constrained minimization problem, Acta Mathematicae Applicatae Sinica-English Series 30(2014) 707–720.

[18] Xikui Liu, Yan Li, Weihai Zhang, Stochastic linear quadratic optimal control with constraint for discrete-time systems, Applied Mathematics and Computation 228(2014) 264-270.

[19] Congying Han, Tingting Feng, Guoping He, Tiande Guo, Parallel variable distribution algorithm for constrained optimization with nonmonotone technique, Journal of Applied Mathematics 2013(2013) 401–420.

[20] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11(2007) 712-731.

[21] Rongwu Lu, Xinhua Wang, Hao Yu, Dan Li, Multiparty evolutionary game model in coal mine safety management and its application, Complexity 2018 (2018) Article No 9620142.

[22] Xin Zhang, Xiu Zhang, Zhou Wu, Dual problem of sorptive barrier design with a multiobjective approach, Neural Computing and Applications 30(2018) 2895-2905.

[23] Xin Zhang, Xiu Zhang, Lei Wang, Antenna design by an adaptive variable differential artificial bee colony algorithm, IEEE Transactions on Magnetics 54 (2018) Article No 7201704.

[24] Z. S. Zhao, X. Feng, Y. Y. Lin, F. Wei, S. K. Wang, T. L. Xiao, M. Y. Cao, Z. G. Hou, M. Tan, Improved rao-blackwellized particle filter by particle swarm optimization, Journal of Applied Mathematics 2013 (2013) Article No. 302170.

[25] S. Pang, T. Li, F. Dai, M. Yu, Particle swarm optimization algorithm for multisalesman problem with time and capacity constraints, Applied Mathematics & Information Sciences 7(2013) 2439–2444.

[26] S. Y. Yuen, C. K. Chow, A Genetic Algorithm That Adaptively Mutates and Never Revisits, IEEE Transactions on Evolutionary Computation 13(2009) 454-472.

[27] R. Storn, K. Price, Differential Evolution-A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, Journal of Global Optimization 11(1997) 341-359.

[28] Xin Jin, Yongquan Liang, Dongping Tian, Fuzhen Zhuang, Particle swarm optimization using dimension selection methods, Applied Mathematics and Computation 219(2013) 5185-5197.

[29] Xin Zhang, Xiu Zhang, Lei Wang, Antenna design by an adaptive variable differential artificial bee colony algorithm, IEEE Transactions on Magnetics 54(2018) Article No 7201704.

[30] Z. Wu, T. W. S. Chow, Neighborhood field for cooperative optimization, Soft Computing 17(2012) 819-834.

[31] Z. Wu, H. Tazvinga, X. Xia, Demand side management of photovoltaic-battery hybrid system, Applied Energy 148(2015) 294-304.

[32] Y. Lou, S. Y. Yuen, Non-revisiting genetic algorithm with adaptive mutation using constant memory, Memetic Computing 8(2016) 189-210.

# Author-Biography

**Xiu Zhang** received the B.Eng. and M.Eng. degrees in biomedical engineering from Hebei University of Technology, Tianjin, China, in 2006 and 2009, respectively, and the Ph.D. degree in electrical engineering from The Hong Kong Polytechnic University in 2012. From 2013 to 2015, she was a Post-Doctoral Fellow with The Hong Kong Polytechnic University. She is currently an associate professor with Tianjin Normal University. Her research interests are mainly focused on numerical methods of electromagnetic field computation, novel wireless energy transfer systems, and wireless network optimization.

**Xiaohui** Lu received the B.Eng. degree from Weifang University in 2018. Since 2018, she has been a student at Tianjin Normal University and studying for a Master's degree. Her main research interests are wireless networks, wireless power transmission and evolutionary computation.

**Xin Zhang** received the B.Sc. degree from Ludong University in 2006, the M.Sc. degree from the Shandong University of Science and Technology in 2009, and the Ph.D. degree from the City University of Hong Kong in 2013. Since 2015, he has been a lecturer at Tianjin Normal University. He has published more than 50 technical papers, including over 30 papers in international journals. His main research interests are resource allocation, evolutionary computation, and machine intelligence.

**Declaration of interests**

☑ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: