Wireless Pers Commun
DOI 10.1007/s11277-017-4746-6

An Estimation of Distribution Algorithm Based Dynamic Clustering Approach for Wireless Sensor Networks

Dongbin Jiao[1] · Liangjun Ke[1] · Weibo Yang[1] · Jing Li[2]

Abstract The design of energy efficiency is a very challenging issue for wireless sensor networks (WSNs). Clustering provides an effective means of tackling the issue. It could reduce energy consumption of the nodes and prolong the network lifetime. However, cluster heads deplete more energy since they bear great load of receiving, aggregation and transmission data than sensor nodes in WSNs. Therefore, the load-balanced clustering is a most significant problem for WSNs with unequal load of the sensor nodes but it is known to be an NP-hard problem. In this paper, we introduce a new model for this problem in which the objective function is to maximize the overall minimum lifetime of the cluster heads. To solve this model, we propose a novel estimation of distribution algorithm based dynamic clustering approach (EDA-MADCA). In EDA-MADCA, a new vector encoding is introduced for representing a complete clustering solution and a probability matrix model is constructed to guide the individual search. In addition, EDA-MADCA merges the EDA based exploration and the local search based exploitation within the memetic algorithm framework. A minimum-lifetime-based local search strategy is presented to avoid invalid search and enhance the local exploitation of the EDA. Experiment results demonstrate that EDA-MADCA can prolong network lifetime, it outperforms the existing DECA algorithm in terms of various performance metrics.

This paper was presented in part at the 20th IEEE International Conference on Computational Science and Engineering (CSE), Guangzhou, China, July 2017 [65].

✉ Liangjun Ke
keljxjtu@xjtu.edu.cn

Dongbin Jiao
dbjiao@stu.xjtu.edu.cn

Weibo Yang
yang.wei.bo.09@stu.xjtu.edu.cn

1 State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an 710049, China

2 State Key Laboratory of Astronautic Dynamics, Xi'an Satellite Control Center, Xi'an 710043, China

Published online: 24 August 2017

Keywords Energy efficiency · Load-balanced clustering · Estimation of distribution algorithm · Memetic algorithm · Minimum lifetime · Wireless sensor networks

1 Introduction

Wireless sensor networks (WSNs) is an advanced information acquisition technology. It has got a considerable development in recent years. A WSN typically is equipped with a large number of small, low power and inexpensive communication devices called sensor nodes, which are randomly or manually densely deployed in an unattended target area and harsh environments. A significant class of WSNs applications is the monitoring area, such as continuous sensing, event identification (ID), event detection, local control of actuators and location sensing [2]. The application fields include health care, agriculture and environment monitoring, public safety and military systems, transportation systems, industry and so on [44]. These applications need to deploy plenty of sensor nodes for continuous sensing, data aggregating and communicating. These sensor nodes periodically collect and process local sensing information, and finally send it to the remote processing center which is called base station (BS) or sink.

Generally, the sensor nodes are deployed in desolate area or vile environment, transmitting or receiving data is over the wireless medium, the replacement or recharging of the embedded batteries is a very difficult and impractical process once these nodes have been deployed. Therefore, energy is a very precious resource for WSNs and has to be managed wisely so as to prolong the network lifetime for the duration of a specific mission [20]. For the sake of tackling the challenging issue, various technologies have been studied such as low-power radio communication hardware [13], energy-aware medium access control (MAC) layer protocols [61]. However, hierarchical clustering [1, 8, 10, 43, 51] has been testified to be an effective technique to conserve sensor energy [34, 62] and also be a promising solution to schedulable tasks .

In hierarchical clustering architecture, the sensor nodes are divided into different clusters, each cluster consists of cluster head (CH) and cluster membership. The set of cluster heads forms backbone of a WSN, providing a scalable solution to organizational networking tasks, and acting as local controllers of network workings [55]. In each cluster, various missions are managed by cluster head, such as data receiving, data aggregation, transmission, authentication, task assignment. The advantages of cluster techniques in WSNs can be summarized as follows [1]:

- Routing can be easily managed, because it can localize the route build within the cluster which requires small routing tables to store at the sensor node and thus reduce long-haul transmission.
- It can save communication bandwidth as it limits each sensor node interactions to only one cluster head and averts redundant exchange of messages among sensor nodes.
- It can enhance the scalability of the network topology, cut down the entire network overhead and the end-to-end transmission delay between the sensor node and the base station, thus extend the battery life of each sensor node and the network lifetime.

However, as the leader of a cluster, cluster head has more energy consumption than cluster membership, it signifies that cluster head bears some extra work load due to various activities compared to their member sensor nodes. These activities include receiving sensed data, sending control message, data aggregation and transmitting data to the base station. In addition, in hierarchical clustering of WSNs, cluster heads are needed to cover a large area of interest without reducing the service quality of the system [35]. But the sensor nodes and cluster heads may be not "well distributed", some cluster heads may be overloaded owing to connecting too many sensor nodes and long-haul communication with the base station. Such overload not only increases communication delay and decreases performance of overall network, but also shortens the network lifetime.

Therefore, load-balanced distribution of cluster heads is one of the critical issues of WSNs, it can efficiently make use of scarce energy resources in battery operated sensor nodes. This problem is also known Load-Balanced Clustering Problem (LBCP), it has been proved to be NP-hard [35], and hence too computationally expensive to find out proper size of cluster for a large-scale WSN by exact algorithms.

In this paper, the problem of load-balanced clustering is briefly expressed as a WSN with n sensor nodes $S = (s_1, s_2, \ldots, s_n)$ and m cluster heads $CH = (ch_1, ch_2, \ldots, ch_m)$ (The notation is summarized in Table 1.) form the appropriate size of cluster of the sensor nodes around the cluster heads in order to reduce the overall energy utilization and improve the entire network lifetime, this process is not static but rather dynamically adjusted to the cluster sizes according to the remaining energy of cluster heads.

However, since the number of all possible clusterings is very huge, it is impracticable to enumerate. Sampling is a reasonable way to deal with such a kind of problem. Estimation of distribution algorithm (EDA) [32, 40] is a sampling based optimization tool which has been used for various academic and engineering problems [54]. Here, we propose an estimation of distribution algorithm (EDA) based dynamic clustering approach for WSNs called EDA-MADCA. To the best of our knowledge, we are the first to employ EDA to

Table 1 Meanings of the notations

Notation	Meaning
n	Number of sensor nodes
m	Number of cluster heads
S	Set of sensor nodes
CH	Set of cluster heads
BS	Base station
s_i	ith node in S
ch_i	ith node in CH
d	Euclidian distance between s_i and ch_i
R	Maximum communication range
$round_{max}$	Maximum number of rounds
$l(ch_i)$	$ch_i's$ lifetime
c_i	Number of sensor nodes which can be assigned to ch_i
$E_{s_i}(0)$	Initial energy of s_i
$E_{ch_i}(0)$	Initial energy of ch_i
$E_{residual}(ch_i, r)$	Residual energy of ch_i in rth round
$E_{req}(ch_i, r)$	Requested energy of ch_i in rth round
$E_{DA}(ch_i, r)$	Aggregation energy of ch_i in rth round
$E_R(ch_i, r)$	Energy depleted by ch_i during receiving packet in rth round
$E_T(ch_i, r)$	Energy consumed during ch_i transmission to BS in rth round

address the load-balanced clustering problem in WSNs. The contributions of our paper are enumerated as follows:

- A new model is introduced for LBCP in which the objective function is to maximize the overall minimum lifetime of the cluster heads.
- An EDA-MA based dynamic clustering approach is proposed for WSNs to extend network lifetime.
- A vector encoding based on EDA is adopted for representing clustering solution.
- A minimum-lifetime-based local search (MLLS) strategy is presented to improve the local exploitation of the EDA and enhance the optimization performance.
- Compared to DECA [29], which is a state of the art metaheuristic for LBCP, EDA-MADCA performs better in terms of network lifetime, half dead cluster head (HDCH), last dead cluster head (LDCH), stability and instability period, number of alive sensor nodes, balanced degree of energy consumption (BDEC), energy consumption and search process of optimal solution.

The remainder of this paper is structured as follows. Section 2 reviews the related work. Section 3 describes a new model for LBCP. Section 4 presents EDA-MADCA that is tailored for WSNs. Section 5 demonstrates experiments and simulation results. Finally, Section 6 concludes the paper.

2 Related Work

Energy-efficient communication is one of the critical issues in WSNs, and clustering provides an effective means for conserving energy and extending network lifetime, thus it has been extensively investigated in the literature. The following summarizes some of the most relevant research papers. We first review the general methods related to clustering and lifetime maximization, then focus on metaheuristic approaches.

2.1 Clustering and Lifetime Maximization

LEACH [24] is a famous clustering method and has been widely regarded as a benchmark. In LEACH, each sensor node is selected as a cluster head by a certain probability per round, and the role of being a cluster head is rotated among the sensor nodes to fairly distribute the task and balance the load. However, in the worst case, the sensor node with very low energy gets selected as a cluster head which may die quickly. Furthermore, the cluster head transmits directly the data to base station by single-hop within their communication range and this is unrealistic. Therefore, it is not applied to WSNs deployed in large area [1].

PEGASIS [33] improves the performance of LEACH and extends the network lifetime with a linear chain topology. It reduces the sensor nodes overhead owing to dynamic cluster formation, but it requires global information of the locations of all sensor nodes at the base station, so the delay is significant and it is not well suitable to large-scale networks.

Unlike LEACH, HEED [63] considers the intra-cluster communication dissipation and overcomes the drawback of nonuniform distributed cluster heads. Only sensor nodes with high remaining energy can become cluster head. Thus, it balances energy consumption and prolongs the network lifetime. One disadvantage of HEED is the iterative process and takes considerably more time.

EECS [60] proposes a distance-based cluster formation approach which generates cluster of uneven size in single-hop networks. In order to balance energy consumption, a weight function is used to allow clusters closer to the base station to deplete less energy during data transmission. One drawback of EECS needs to knowledge of the distance between each sensor node and the base station.

As a classic clustering method, WCA [15] takes into account connectivity balance, the voting of cluster heads depends on the sensor node degree. But it is the necessity of knowledge of the weights of sensor nodes, hence the communications result extra dissipation.

For balancing load among the cluster heads, Gupta et al. [20] define cardinality of the cluster head and try to minimize the variance of each cluster head in WSNs, which improves the network lifetime.

Low et al. [35] take into account the issue of assigning sensor nodes to cluster head so as to distribute the traffic load among the cluster heads to ensure the load balances. As the time complexity of the method is $O(mn^2)$ for n sensor nodes and m cluster heads, thus it is unsuitable to deal with the large-scale networks.

2.2 Metaheuristic Approaches

Many metaheuristic algorithms have been proposed in WSNs and here we survey some promising approaches related to clustering and lifetime maximization.

Hussain et al. [27] propose a genetic algorithms (GA) based energy efficiency hierarchical clustering method which increases the network lifetime and it adopts four fitness parameters to define the fitness function. But the method ignores load balancing between sensor nodes and cluster heads. Bari et al. [11] present a GA-based approach for data routing by relay nodes in two-tiered WSNs. Selection of solution is performed by the Roulette-Wheel selection method and the fitness function is defined as it takes into account the currently available nodes and their remaining energy in terms of rounds. For mutation operation, a critical node which consumes the maximum energy is selected. However, they did not consider data communication between the sensor nodes and the cluster heads within each cluster. Kuila et al. [30] propose a GA based load-balanced clustering method for WSNs. The method forms the proper size of cluster which the maximum load of each cluster head is minimized. It can operate between unbalance load of sensor nodes. The main drawback is that the cluster heads directly communicate with the base station which may be impractical for large-scale network. Besides, the remaining energy of sensor nodes and cluster heads were not considered per round so that their energy consumption are imbalance.

Chakraborty et al. [14] present a differential evolution (DE) based memetic algorithm (MA) which addresses routing problems with more than a thousand relay nodes. Its goal is to accomplish the cluster heads that minimizes the maximum energy expenditure. Whereas, the method does not considers the cluster formation and inappropriate clustering may cause energy dissipation. Kuila et al. [29] propose a novel DE based clustering algorithm for WSNs. A new fitness function which takes into consideration energy expenditure of both the cluster heads and sensor nodes is derived for prolonging the network lifetime. Meanwhile, the method embeds a local improvement so as to accelerate convergence rate and get better performance. However, it ignores the distance between the cluster heads and the base station and the cluster heads are chose randomly which may lead to energy inefficiency.

Fuzzy method is used in some of research works for load-balanced clustering of the WSNs. A fuzzy logic based cluster-head election is proposed in [21] to reduce the energy consumption and prolong the network lifetime. In this approach, three parameters, i.e., energy, concentration and centrality, are used as input variables. Compared to LEACH, the approach achieved better results. However, it requires centralized means for electing cluster head, hence the algorithm is difficult to scalability. In DUCF [9], a fuzzy inference system based distributed cluster head elect method is devised, it forms unequal clusters to balance the energy cost among the cluster heads, node residual energy, node degree and distance between node and BS as input variables, node size and chance as output fuzzy parameters for cluster head election. This method guarantees load balancing among the clusters by changing the cluster size of its cluster head and extends the network lifetime. Some other fuzzy-based methods for clustering are introduced in [4, 5, 28, 48, 52].

Singh et al. [50] propose a novel energy-aware cluster head selection in WSNs using particle swarm optimization (PSO), whereas it does not take into account the cluster formation. Azharuddin et al. [3] present PSO-based routing and clustering approaches for WSNs. Wang et al. [53] propose an ant colony (ACO) based clustering routing algorithm in WSNs. For more surveys on evolutionary algorithms for WSNs, they can be found in [31, 38, 46, 47, 49, 64] and the references therein.

In load-balanced clustering problem (LBCP), the residual energy of cluster heads and sensor nodes is the utmost essential factors for balancing load and prolonging the network lifetime. In this paper, we propose a novel algorithm (i.e., EDA-MADCA) for solving the problem.

3 A New Model for LBCP

3.1 Notations

we summarize the notations in Table 1.

3.2 Energy Model

The radio energy dissipation model in [24] is used here. In this model, the energy consumption mainly occurs at the transmitter, the power amplifier, and the receiver to run the radio electronics. It adopts the free space and the multi-path fading channel, depending on the distance between the transmitter and receiver. The total energy expended to deliver an l-bit packet from the transmitter to its receiver over a link of distance d is shown below:

$$E(l,d) = E_T(l) + E_{T_{amp}}(l,d) + E_R(l) \tag{1}$$

$$E(l,d) = \begin{cases} lE_{elec} + l\epsilon_{fs}d^2 + lE_{elec} & \text{if} \quad d < d_{TH} \\ lE_{elec} + l\epsilon_{mp}d^4 + lE_{elec} & \text{if} \quad d \geq d_{TH} \end{cases} \tag{2}$$

In Eq. (2), E_{elec} is the electronics energy which depends on some factors [23], such as digital coding, modulation, filtering, and spreading of the signal, but the amplifier energy may be $\epsilon_{fs}d^2$ or $\epsilon_{mp}d^4$ depending on the distance between the transmitter and the receiver and the acceptable bit-error rate, d_{TH} is the threshold distance. The energy consumption which aggregates $\bar{n}$ message signals of length l-bit was calculated as:

$$E_{DA} = \bar{n} l E_{da} \tag{3}$$

where E_{da} is the energy required for data aggregation. The radio channel is assumed to be symmetric.

3.3 Network Model

As in [29], we assume the WSNs scenario as follows:

- All the sensor nodes are randomly deployed along with a few cluster heads within a square area.
- Cluster heads are chosen prior and the locations of sensor nodes are known.
- The positions of all nodes (i.e., sensor nodes and cluster heads) and base station are fixed once they are deployed.
- The message can be transmitted between a sensor node and cluster head by wireless link.
- Each sensor node has a list of cluster heads and it can be assigned to only one cluster head by single-hop within their communication range per round.
- All sensor nodes have the same initial energy, and all cluster heads have the same energy, but the energy of cluster heads is more than the energy of sensor nodes. The base station has no energy restriction.
- A sensor node is regarded as alive if only its energy is larger than zero and at least one alive cluster head is reachable within its communication range. In case a sensor node can not find any cluster head within its communication range, even though it may have some remaining energy, it is still regarded as dead in the network model.

Figure 1 shows a basic hierarchical clustering of WSN model. The operation of data gathering is divided into rounds as done in LEACH. Each round consists of two stages: set-up and clustering. In the set-up phase, all sensor nodes and cluster heads are assigned to a unique ID. The sensor nodes broadcast message containing their ID by CSMA/CA MAC protocol [12]. The cluster heads within their communication range of the sensor nodes can collect the message and send the local sensing information to the BS. During the clustering phase, the BS receives the data gathering from cluster heads by executing clustering

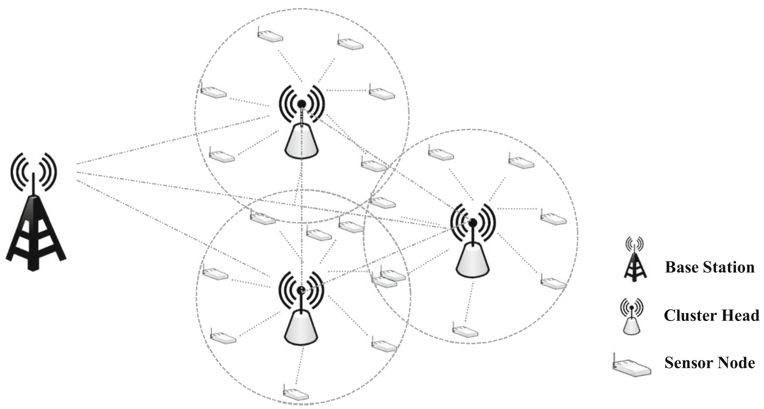

Fig. 1 A basic hierarchical clustering of WSN model

algorithm. Then all cluster heads send an announcement message to their member by single-hop. In each cluster, the communication uses TDMA protocol [12].

3.4 Problem Representation

3.4.1 Definition of Lifetime

The WSNs lifetime is defined by different ways in [18]. Among them, the popular definition of network lifetime is the time until the first cluster head depletes all its energy. The lifetime $l(ch_i)$ of cluster head ch_i is calculated as:

$$l(ch_i) = \left\lfloor \frac{E_{residual}(ch_i, r)}{E_{req}(ch_i, r)} \right\rfloor \tag{4}$$

where the $E_{residual}(ch_i, r)$ and $E_{req}(ch_i, r)$ can be calculated respectively as follows:

$$E_{residual}(ch_i, r) = E_{residual}(ch_i, r-1) - E_{req}(ch_i, r) \tag{5}$$

$$E_{req}(ch_i, r) = E_{message}(ch_i, r) + E_{packet}(ch_i, r) \tag{6}$$

$$E_{message}(ch_i, r) = E_{mB}(ch_i, r) + E_{mR}(ch_i, r) + E_{mT}(ch_i, r) \tag{7}$$

$$E_{packet}(ch_i, r) = c_i E_R(ch_i, r) + c_i E_{DA}(ch_i, r) + E_T(ch_i, r) \tag{8}$$

The energy consumed by cluster head ch_i, denoted by $E_{req}(ch_i, r)$, is composed of $E_{message}(ch_i, r)$ which is consumed by broadcasting message and $E_{packet}(ch_i, r)$ which is consumed during delivering packet each round. They are calculated by Eqs. (2) and (3), respectively. Note that $E_{message}(ch_i, r)$ is composed of three parts: (1) $E_{mB}(ch_i, r)$, energy consumption during cluster head ch_i broadcasts message to its member sensor nodes, (2) $E_{mR}(ch_i, r)$, energy consumption during sensor node s_i replies with an acknowledgment, (3) $E_{mT}(ch_i, r)$, energy consumption during cluster head ch_i sends an acknowledgment message to sensor node s_i.

3.4.2 Objective Function

In [29], the objective function is defined as the standard deviation of the cluster heads lifetime and standard deviation of the average distance between cluster heads and cluster membership in order to keep the load balance between cluster heads and cluster membership. However, this objective function ignores the distances between cluster heads and base station. To optimize such an objective function, some cluster heads far away from base station may be assigned too many sensor nodes, which may lead to more energy consumption and these cluster heads will die earlier.

In this paper, we notice that the remaining energy of each cluster head is critical to prolong the network lifetime and a possible way to predict the cluster head lifetime is to take account of the remaining energy which was also considered in [56]. This motivates us to build the objective function as the predicted cluster head lifetime according to residual energy $E_{residual}(ch_i, r)$ and request energy $E_{req}(ch_i, r)$. Each round, starting from each cluster head and ending at the BS, the maximum predicted lifetime L by a cluster head is given as:

$$L = \max_{c \in \Omega} \min_{1 \leq i \leq m} l(ch_i) \tag{9}$$

where Ω is the set of all possible clustering, variable c is a clustering, $(ch_1, ch_2, \ldots, ch_m)$ corresponds to c, $l(ch_i)$ is calculated by Eq. (4). The goal is to maximize the overall minimum l per round so as to extend the network lifetime.

This objective function takes into account not only the energy expenditure between sensor nodes and cluster heads but also the energy consumption between cluster heads and base station. By optimizing this objective function, it is helpful to adaptively adjust the distance between sensor nodes and cluster heads, distance between cluster heads and base station, and the size of clusters, thereby balancing the load of cluster heads and reducing the data delivering overhead. As a result, the network lifetime could be prolonged.

4 Estimation of Distribution Algorithm Based Dynamic Clustering Approach

Estimation of distribution algorithms (EDAs) [32, 40] are a class of evolutionary algorithms (EAs). Compared to other EAs, EDAs employ neither crossover nor mutation. They build an explicit probabilistic model and sample promising candidate solutions according to the probabilistic model. The probabilistic model will be updated with the generated solutions. The main steps of basic EDAs are described as follows [37]:

Let *Pop*(g) denotes the population of solutions at generation g.

- Step 0: **Initialization.** Set $g \leftarrow 0$. Generate P_{size} individuals as randomly initial population *Pop*(0).
- Step 1: **Selection.** Select K promising individuals from *Pop*(g) to consist of the new population $S(g)$ individuals by a selection method based on the fitness function.
- Step 2: **Modelling.** Build a probabilistic model p based on the probability of distribution of the selected $S(g)$.
- Step 3: **Sampling.** Sample new offspring according to the built probabilistic model p.
- Step 4: **Replacement.** Some individual in *Pop*(g) are replaced with new generated offspring to form a new population $Pop(g+1)$.
- Step 5: If the stopping criterion is met, stop and return the best solution. Otherwise, $g \leftarrow g + 1$ and go to Step 1.

As a classic EDA, population-based incremental learning (PBIL) [6, 7] has already attracted significant attentions recently. It is developed by combining genetic algorithms (GA) and competitive learning for maintaining the stochastic search space and using them to guide the search to explore promising solutions by a probability vector constructed by using the elite sub-population. PBIL has been successfully used to address a variety of hard problems, such as job-shop scheduling [54], TSP [22, 39], bin packing [6], inverse problem [26, 59], power system stabilizer design [19], network coding [57], scattering problem [17], multi-objective optimization problems [25] and dynamic optimization problems [58].

In our study, the EDA is incorporated in the MA framework. Memetic algorithms [16, 41, 42] are widely employed as a cooperation between evolutionary or any population-based method and individual learning or local search procedures [39, 54].

The idea of EDA-MADCA is to balance the energy consumption among the cluster heads which transmit to the cluster membership by adopting dynamically clustering, balancing the load of cluster heads and forming appropriate sized clusters so as to prolong

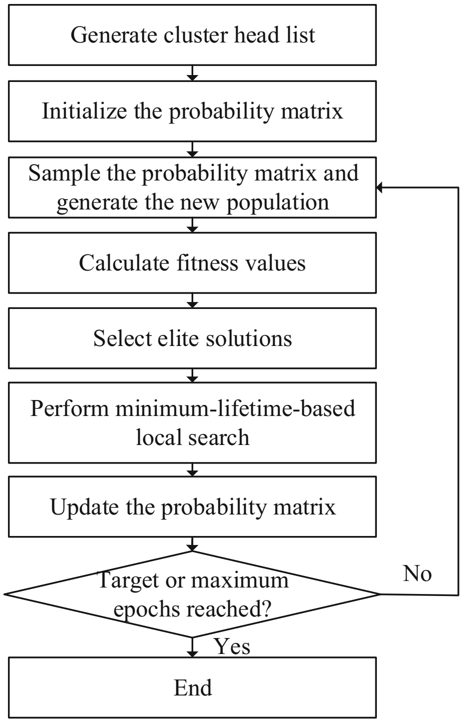

Fig. 2 The flowchart of EDA-MADCA

the entire network lifetime. A flowchart of EDA-MADCA is shown in Fig. 2. The framework of EDA-MADCA is introduced in the subsections as follows.

4.1 Generation of Cluster Head List

In this network, since communication range of sensor nodes is limited, the sensor nodes only select neighboring cluster heads to build wireless link. In other words, if a cluster head is beyond the transmitting range, they can not communicate with each other. Therefore, within its communication range, each sensor node corresponds to some reachable cluster heads. The generation of the cluster head list is shown in Algorithm 1.

Algorithm 1 Generate Cluster Head List

Input: $n, m, S, CH, round_{max}$ (see Table. 1).
while $r \le round_{max}$ **do**
  for $i = 1$ to n **do**
    for $j = 1$ to m **do**
      if $E_{residual}(s_i, r) > 0$ and $E_{residual}(ch_j, r) > 0$ and $d < R$ **then**
        $count(s_i, r) \leftarrow count(s_i, r) + 1$
        $count(ch_j, r) \leftarrow count(ch_j, r) + 1$
        $ClusterHeadsList(i, count(s_i, r)) = j$
      end if
    end for
  end for
end while
Output: $ClusterHeadList$

To explain the generation of cluster head list, an example instance in which 10 sensor nodes and 3 cluster heads are deployed within a sensing area of 200×200 m^2 (others parameters is shown in Table 3) provided in Table 2. One can notice that the sensor nodes can be assigned to the list of candidate cluster head. For instance, s_1 can only be assigned to ch_3, whereas s_4 selects randomly among ch_1, ch_2 and ch_3.

4.2 Solution Representation

A valid encoding is significance to solve LBCP. It is helpful to avoid producing redundant solutions.

In EDA-MADCA, a solution is denoted by a cluster head vector *CH* as show in Fig. 3 (the example is according to the instance given in Table 2). The elements in cluster heads vector *CH* correspond to the IDs of cluster heads. The length of each solution is equal to the number of sensor nodes. In detail, the element ch_i means that sensor node s_i is assigned to cluster head ch_i in clustering phase. Some elements may have the same ID, which means that those sensor nodes belong to the same cluster head ch_i.

As the length of each solution is equivalent to the number of the sensor nodes, adding or removing any sensor node will change the length of a solution and re-clustering is required.

4.3 Fitness Function

Fitness function measures the quality of the population of solutions. A well-defined fitness function is always problem dependent and is helpful to increase the chance of searching promising area of the solution space. The fitness function is defined as follows:

Table 2 Cluster heads list

Sensor node	Candidate cluster heads
s_1	ch_3
s_2	ch_1, ch_3
s_3	ch_1
s_4	ch_1, ch_2, ch_3
s_5	ch_1, ch_2, ch_3
s_6	ch_1
s_7	ch_1, ch_3
s_8	ch_1, ch_3
s_9	ch_2, ch_3
s_{10}	ch_1

Sensor node sequence vector	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
Cluster head vector	**3**	**3**	**1**	**2**	**2**	**1**	**1**	**3**	**3**	**1**

Fig. 3 Solution of the example instance

$$L = \max_{1 \leq j \leq Pop_{size}} \min_{1 \leq i \leq m} l(ch_{ji}) \tag{10}$$

From Eq. (10), we can see that the larger value of L, the higher the fitness value is.

4.4 Design of Probability Model

In some sense, the probabilistic model describes the distribution of solution space. The standard PBIL uses a real-valued probability vector and samples solutions according to the probability vector. In general, the probabilistic model is built based on the characteristics of the solving problem.

In light of the above cluster heads list and solution representation, a single real-valued probability vector may be not good for delivering the information learnt before. Therefore, the probability model is constructed as a probability matrix P, where $P = [p_{ij}]_{n \times m}$, $p_{ij} = 1/count(s_i, r)$, $i \in n, j \in m$. p_{ij} denotes the probability that the sensor node s_i is assigned to the cluster head ch_j and modeled as a uniform distribution within its communication range. Note that if the sensor sensor s_i can not be assigned to the cluster head ch_j in this round, the probability $p_{ij} = 0$.

4.5 Selection of Elite Solutions

At each iteration of the EDA-MADCA, to generate a new solution, a cluster head vector CH is produced by the roulette strategy via sampling the searching space guided by the probability matrix P. To explore the promising searching area, the probability matrix P should be well adjusted by using some elite solutions [54]. The elite solutions form a superior sub-population. Let ES_{size} be the size of the sub-population. In our implementation, ES_{size} is set to $\theta \cdot Pop_{size}$ where $\theta \in [0, 1]$ is a parameter. Elite solutions are selected from the population according to the tournament selection strategy, then they are employed to update the probability matrix P at next iteration.

4.6 Minimum-Lifetime-Based Local Search

In order to enhance the exploitation ability, we present a MLLS strategy. The basic idea is to move some sensor nodes from the cluster head with the minimum lifetime to other cluster nodes, thereby delaying the first death of cluster heads. Using each elite solution as input, it works as follows:

- Find the cluster head with the minimum lifetime and the set of sensor nodes assigned to the cluster head.
- Randomly select a sensor node from the set, then reassign it to another cluster head which is closest in its candidate cluster heads list except its previous cluster head.
- Evaluate the modified fitness value of the elite solution. If the new fitness value is better than the previous one, the elite solution will be replaced by the new solution. Otherwise, the previous solution remains constant.

This process is described in Algorithm 2.

Algorithm 2 Minimum Lifetime Based Local Search

Input: *elitesolution*, *clusterlifetime*, ES_{size}, n, m (see Table 1)

```
for i = 1 to ES_size do
    [clustelifetime_min, N_min] = min(clusterlifetime(i, m))
    solution = elitesolution(i, n)
    cluster_min = find(solution == N_min)
    len_min = size(cluster_min)
    if len_min ≠ 0 then
        sensornode = cluster_min(unidrnd(len_min))
        clusterhead ← 0
        d ← ∞
        if m > 1 then
            for j = 1 to m do
                if j ≠ N_min then
                    d' ← distance(sensornode, j)
                    if d' < d then
                        clusterhead ← j
                        d ← d'
                    end if
                end if
            end for
        end if
        result = find(ClusterHeadsList(sensornode, :) == clusterhead)
        temp = size(result)
        if temp(2) ≠ 0 then
            solution(sensornode) ← clusterhead
        end if
    end if
    for k to n do
        Count(solution(k)) ← Count(solution(k)) + 1
    end for
    CalculateFitness
    if Fitness_elite(i) < Fitness then
        elitesolution(i, n) = solution
        Fitness_elite(i) = Fitness
        clusterlifetime_elite(i) = clusterlifetime
    end if
end for
```

Output: better elite solution

4.7 Updating Mechanism

After the superior sub-population have been modified, the the probability matrix P will be updated by using the historical knowledge of searching and the statistics information of superior sub-population. The updating process is based on the Hebbian-inspire rule [36] which is expressed as follows:

$$p_{ij}(g+1) = (1-\alpha)p_{ij}(g) + \alpha \frac{1}{ES_{size}} \sum_{j=1}^{ES_{size}} {X_{ij}}^k \tag{11}$$

where $i = 1, 2, \ldots n$. $\alpha \in (0, 1)$ is the learning rate, implies the maximum proportion of solutions to be chose. In our paper, $\alpha = 20\%$. ${X_{ij}}^k$ is the following indicator function within the kth solution of the superior sub-population:

$$X_{ij}^k = \begin{cases} 1 & \text{if } ch_j \text{ appears before or in the } i\text{th position} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

In Eq. (11), the first term states the information inherited from the parent and the second term states the statistics information learnt from the superior sub-population. The parameter α controls the contribution of the parent when updating the probability matrix P.

4.8 Computational Complexity Analysis

At each iteration of the EDA-MADCA, the computational complexity is analyzed as below.

As for the selection process of elite solutions, the computational complexity is $O(ES_{size})$. As for the updating process of the probability matrix P, the computational complexity is $O[n(ES_{size} + n)]$. As for the sampling procedure, since it produces the cluster head vector via the roulette strategy, the computational complexity is $O(n^2)$. Besides, the computational complexity of assigning the sensor nodes to their cluster heads is $O(nm)$.

The analysis shows the computational complexity of EDA-MADCA is approximately equal to $O(n^2)$.

5 Experiment Results and Analysis

In this section, we evaluate the performance of EDA-MADCA by comparing it with DECA [29]. To the best of our knowledge, DECA is the best performance in this network model. To be fair comparison, we run EDA-MADCA and DECA with the same parameters, simulation environment, network topology, fitness function and performance metrics. We have done experiments to compare EDA-MADCA with DECA using the objective function in [29]. The result can be found in "Appendix".

5.1 Experiment Setup

We evaluate the performance of EDA-MADCA via simulations in MATLAB (version R2012b). All experiments were run on a PC with Intel (R) Core (TM) i7, 3.60 GHz CPU, 8 G RAM and Windows 7. The values of simulation parameters are listed in Table 3, which is similar to the ones in [23, 29]. Specifically, sensor nodes are randomly deployed within a square field of 200 m × 200 m, and the maximum communication range R was set to 150 m. The number n of sensor nodes was varied from 100 to 500 and the number m of cluster heads was varied from 30 to 50. The initial energy $E_{s_i}(0)$ of each sensor node and $E_{ch_i}(0)$ of each cluster head are 1 and 5 J, respectively.

We take into account three scenarios to corresponding to different locations of the base station in the network. In detail, in *Scenario* 1, the base station is located at the center of the sensing area, i.e., its coordinate is (100, 100). In *Scenario* 2, the base station is located at the top right-hand corner of the sensing area, i.e., at the (200, 200) coordinate. In *Scenario* 3, the base station is located outside the sensing area at (100, 250).

Table 3 Simulation parameter configuration

Parameter	Value
n	100–500
Field	200×200 m^2
BS	(100, 100), (200, 200), (100, 250)
m	30–50
$E_{s_i}(0)$	1.0 J
$E_{ch_i}(0)$	5.0 J
R	150 m
E_{elec}	50 nJ/bit
ϵ_{fs}	10 pJ/bit/m^2
ϵ_{mp}	0.0013 pJ/bit/m^4
d_0	87.0 m
E_{DA}	5 nJ/bit
Packet size	4000 bits
Message size	200 bits

Table 4 Parameter configuration of algorithms

EDA-MADCA parameter	Value	DECA parameter	Value
Pop_{size}	30	Pop_{size}	30
Learning Rate (α)	0.2	*Scaling Factor* (F)	0.5
θ	0.2	*Crossover Rate* (Cr)	0.7
Max Iteration (g_{max})	200–300	*Max Iteration* (g_{max})	200–300
Running Times	5	*Running Times*	5

To perform EDA-MADCA and DECA, the parameters settings of the two algorithms are shown in Table 4. The two algorithms are run 5 times independently on each data point.

5.2 Performance Measures

There are several metrics to evaluate the performance of the clustering approach [18, 45]. In this study, the following measures are employed to evaluate the performance of the EDA-MADCA:

- *First Dead Cluster Head* (FDCH) as the number of rounds until the first cluster head drains out of energy.
- *Half Dead Cluster Head* (HDCH) as the number of rounds until the half number of cluster heads drains out of energy.
- *Last Dead Cluster Head* (LDCH) as the number of rounds until all cluster heads drain out of energy.
- *Stability period* as the time between the start of network operation and FDCH.
- *Instability period* as the time from FDCH to LDCH.
- *Balanced degree of energy consumption* (BDEC) is calculated as:

$$BDEC = (LDCH - FDCH)/LDCH. \tag{13}$$

The BDEC measures the performance of the algorithm to balance the consumed energy. From Eq. 13, we can see the smaller BDEC is, the better performance with balancing energy expenditure.

- *Number of alive sensor nodes per round* as instantaneous measure the total number of sensor nodes which have not drained out all energy.

5.3 Simulation Results

According to the performance metrics mentioned above, we study the performance of EDA-MADCA and DECA from the following aspects.

5.3.1 Stability Period and Network Lifetime

Figures 4 and 5 compare EDA-MADCA with DECA with respect to FDCH for different scale networks in three scenarios. As can be seen that for more than 91% data points (except three data points in Figs. 4, 5) EDA-MADCA can obtain better values of FDCH.

5.3.2 Instability Period and Number of Alive Sensor Nodes

For brevity, we only report part of results here and similar results are obtained for different scale network in three scenarios. Tables 5, 6 and 7 report the results of HDCH for the three scenarios with 30 cluster heads. Tables 8, 9 and 10 present the results of LDCH for the three scenarios with 50 cluster heads. As seen from Tables 5, 6, 7, 8, 9 and 10, EDA-

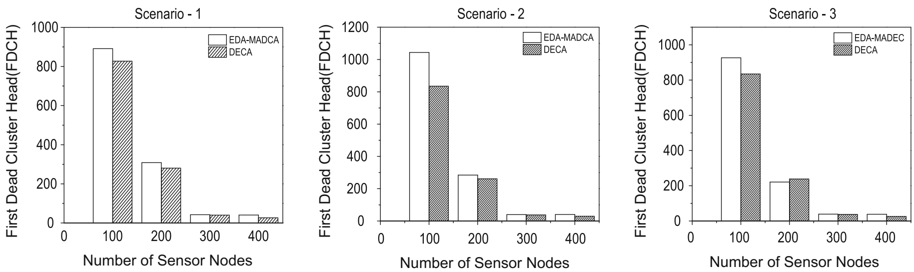

Fig. 4 First dead cluster head (FDCH) of EDA-MADCA and DECA for 30 cluster heads in three scenarios

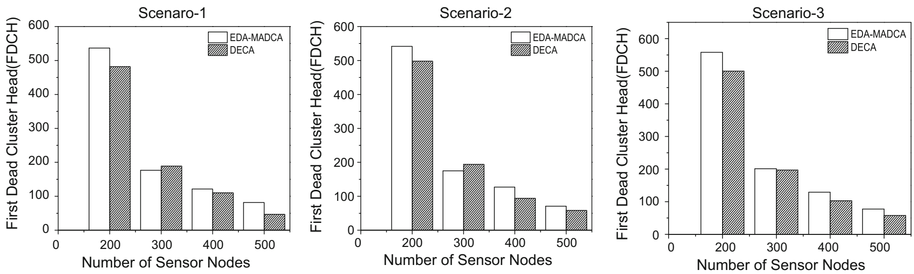

Fig. 5 First dead cluster head (FDCH) of EDA-MADCA and DECA for 50 cluster heads in three scenarios

Table 5 Half dead cluster head (HDCH) of EDA-MADCA and DECA for 30 cluster heads in *Scenario* 1

Algorithm	100 Sensor nodes			200 Sensor nodes			300 Sensor nodes			400 Sensor nodes		
	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
EDA-MADCA	**2247**	**1819**	358.85	**358**	**354.2**	**4.07**	**93**	**92.8**	**0.4**	79	**77**	**1.14**
DECA	1013	986.4	16.21	353	349	4.24	92	90.8	1.17	80	76	3.41

Bold value indicates the better value between the two algorithm in the tables

Table 6 Half dead cluster head (HDCH) of EDA-MADCA and DECA for 30 cluster heads in *Scenario* 2

Algorithm	100 Sensor nodes			200 Sensor nodes			300 Sensor nodes			400 Sensor nodes		
	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
EDA-MADCA	**2159**	**1844**	249.72	**359**	**355**	**3.35**	**95**	**93.6**	**0.8**	**80**	**77.8**	**1.47**
DECA	1022	1007.8	12.72	345	333	9.94	92	91	1.26	79	74.4	2.42

Bold value indicates the better value between the two algorithm in the tables

Table 7 Half dead cluster head (HDCH) of EDA-MADCA and DECA for 30 cluster heads in *Scenario* 3

Algorithm	100 Sensor nodes			200 Sensor nodes			300 Sensor nodes			400 Sensor nodes		
	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
EDA-MADCA	**2198**	**1748.4**	228.96	349	**340.8**	**6.67**	**94**	**93.4**	**0.49**	78	**77.4**	**0.49**
DECA	1021	990.6	22.04	349	337.2	8.91	92	91	0.63	79	74.6	2.65

Bold value indicates the better value between the two algorithm in the tables

Table 8 Last dead cluster head (LDCH) of EDA-MADCA and DECA for 50 cluster heads in *Scenario* 1

Algorithm	200 Sensor nodes			300 Sensor nodes			400 Sensor nodes			500 Sensor nodes		
	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
EDA-MADCA	**805**	**801.4**	**2.58**	**391**	**388.4**	**1.36**	**259**	**254.2**	**2.99**	**182**	**178.6**	**3.07**
DECA	792	783.2	7.36	385	378.2	4.07	251	247.2	3.12	176	170.4	4.13

Bold value indicates the better value between the two algorithm in the tables

Table 9 Last dead cluster head (LDCH) of EDA-MADCA and DECA for 50 cluster heads in *Scenario* 2

Algorithm	200 Sensor nodes			300 Sensor nodes			400 Sensor nodes			500 Sensor nodes		
	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
EDA-MADCA	**803**	**800.2**	**1.72**	**390**	**388**	1.79	**258**	**257.83**	**0.4**	**179**	**176**	**2**
DECA	793	782.8	9.41	385	382.2	1.72	247	245.5	1.70	177	173.8	2.32

Bold value indicates the better value between the two algorithm in the tables

Table 10 Last dead cluster head (LDCH) of EDA-MADCA and DECA for 50 cluster heads in *Scenario* 3

Algorithm	200 sensor nodes			300 sensor nodes			400 sensor nodes			500 sensor nodes		
	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
EDA-MADCA	**806**	**803.6**	**2.50**	**387**	**386**	**1.26**	**258**	**254.8**	**1.94**	**182**	**178**	**2.76**
DECA	792	789.2	3.19	384	379	3.85	254	247.8	4.59	178	172.6	4.72

Bold value indicates the better value between the two algorithm in the tables

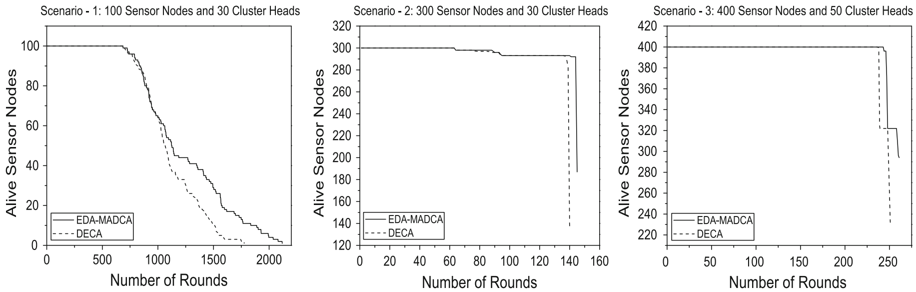

Fig. 6 Alive sensor nodes of EDA-MADCA and DECA for different scale networks in three scenarios

MADCA can achieve better results of the best value, average (mean) value and standard deviations (SD) of HDCH and LDCH except a few data points.

Figure 6 illustrates the distribution of alive sensor nodes with regards to the number of rounds for two algorithms, i.e., EDA-MADCA and DECA. It can be seen that EDA-MADCA declines slower than DECA. In addition, EDA-MADCA can lead to smaller number of inactive sensor nodes than DECA for different scale networks in three scenarios.

5.3.3 *Energy Efficiency*

Energy efficiency is a critical issue in WSNs. In this paper, we study energy efficiency of EDA-MADCA and DECA in terms of BDEC, energy consumption each round, and total energy dissipation.

Figures 7 and 8, EDA-MADCA can find smaller values of BDEC than DECA on 19 out of 24 data points. It means that EDA-MADCA can balance better than DECA. In

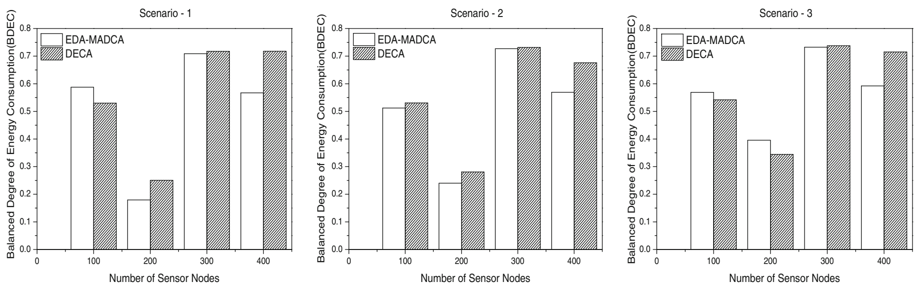

Fig. 7 Balanced degree of energy consumption (BDEC) of EDA-MADCA and DECA for 30 cluster heads in three scenarios

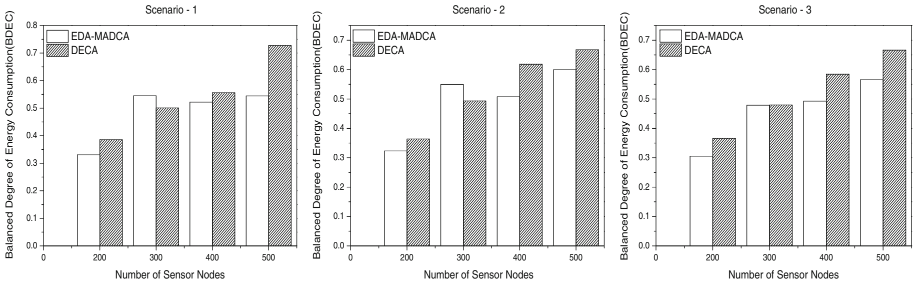

Fig. 8 Balanced degree of energy consumption (BDEC) of EDA-MADCA and DECA for 50 cluster heads in three scenarios

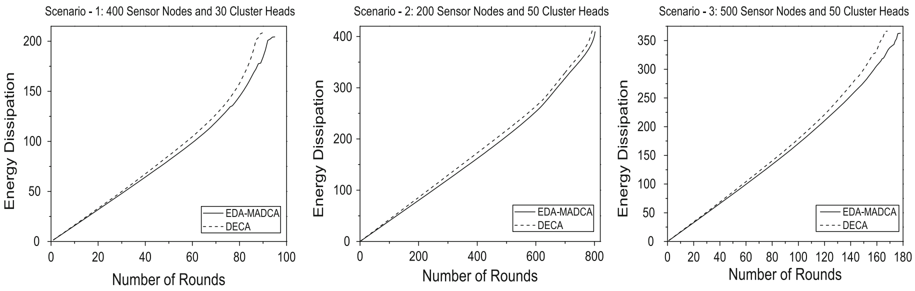

Fig. 9 Energy dissipation of EDA-MADCA and DECA for different scale networks in three scenarios

particular, EDA-MADCA works prominently better than DECA when the network size is $n = 400$ or $n = 500$.

Figure 9 clearly shows the energy consumption of EDA-MADCA and DECA per round for different scale networks in three scenarios. For three different scenarios, we observe that EDA-MADCA consumes less energy than DECA each round. It means that EDA-MADCA is more energy-efficient compared with DECA, because the MLLS strategy considers that the balanced energy dissipation is helpful to improve the lifetime of cluster heads and guarantee the energy consumption of cluster heads evenly. It is noted that the energy dissipation of EDA-MADCA does not vary with the location of the base station. We also notice that similar results are obtained for other scale networks.

From Tables 11, 12, 13, 14, 15 and 16, it is observed that EDA-MADCA is better than DECA in terms of the best value, average (mean) value and standard deviations (SD) of total energy dissipation. At a few data points, the total energy dissipation of EDA-MADCA is slightly higher. For example, when the size of network is $n = 100$, EDA-MADCA consumes more energy than DECA. This is because EDA-MADCA operates a long time compared with DECA. In other words, the LDCH of EDA-MADCA is longer than the one of DECA, so it needs to consume more energy.

5.3.4 The Search Process of Algorithm

Figure 10 shows the search process of EDA-MADCA and DECA for 500 sensor nodes and 50 cluster heads in three scenarios as the fitness value increases. It is observed that DECA has a

Table 11 Total energy dissipation of EDA-MADCA and DECA for 30 cluster heads in *Scenario* 1

Algorithm	100 Sensor nodes			200 Sensor nodes			300 Sensor nodes			400 Sensor nodes		
	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
EDA-MADCA	234.77	236.07	**1.10**	**220.73**	**221.60**	**1.06**	**196.75**	**202.49**	**4.03**	**200.62**	**206.79**	**3.99**
DECA	216.86	219.47	2.35	223.65	225.54	1.24	203.14	209.63	4.13	200.64	212.90	7.43

Bold value indicates the better value between the two algorithm in the tables

Table 12 Total energy dissipation of EDA-MADCA and DECA for 30 cluster heads in *Scenario* 2

Algorithm	100 Sensor nodes			200 Sensor nodes			300 Sensor nodes			400 Sensor nodes		
	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
EDA-MADCA	232.92	235.33	**1.58**	**219.00**	**221.66**	**1.92**	201.50	**204.96**	**2.47**	**196.50**	**209.67**	7.84
DECA	215.30	218.34	2.25	222.59	225.25	2.86	198.18	205.27	4.83	206.58	210.50	3.93

Bold value indicates the better value between the two algorithm in the tables

Table 13 Total energy dissipation of EDA-MADCA and DECA for 30 cluster heads in *Scenario 3*

Algorithm	100 Sensor nodes			200 Sensor nodes			300 Sensor nodes			400 Sensor nodes		
	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
EDA-MADCA	234.30	235.61	**0.74**	**219.42**	**220.66**	**1.32**	198.49	202.71	**2.86**	**201.80**	**208.00**	**3.56**
DECA	217.16	219.69	2.29	222.15	224.90	2.07	195.83	201.53	4.21	208.43	214.28	4.25

Bold value indicates the better value between the two algorithm in the tables

Table 14 Total energy dissipation of EDA-MADCA and DECA for 50 cluster heads in *Scenario* 1

Algorithm	200 Sensor nodes			300 Sensor nodes			400 Sensor nodes			500 Sensor nodes		
	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
EDA-MADCA	**403.81**	**408.12**	3.15	379.97	383.41	**3.03**	372.64	**377.44**	4.66	**363.74**	374.81	7.27
DECA	404.74	408.27	3.14	377.53	381.97	3.81	371.96	378.15	4.12	367.59	374.34	7.03

Bold value indicates the better value between the two algorithm in the tables

Table 15 Total energy dissipation of EDA-MADCA and DECA for 50 cluster heads in *Scenario* 2

Algorithm	200 Sensor nodes			300 Sensor nodes			400 Sensor nodes			500 Sensor nodes		
	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
EDA-MADCA	**401.04**	**406.06**	**3.22**	378.00	**381.63**	**2.88**	**369.08**	375.51	6.49	359.43	**370.31**	**6.23**
DECA	405.12	409.45	3.24	375.60	382.51	5.81	371.62	375.17	3.62	356.25	371.59	9.28

Bold value indicates the better value between the two algorithm in the tables

Table 16 Total energy dissipation of EDA-MADCA and DECA for 50 cluster heads in *Scenario* 3

Algorithm	200 Sensor nodes			300 Sensor nodes			400 Sensor nodes			500 Sensor nodes		
	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
EDA-MADCA	**404.18**	**406.99**	2.59	378.11	379.76	**2.35**	373.69	379.66	**4.47**	362.86	375.82	**10.78**
DECA	410.83	411.91	1.43	375.63	379.30	3.91	367.25	376.35	7.30	353.60	369.92	16.47

Bold value indicates the better value between the two algorithm in the tables

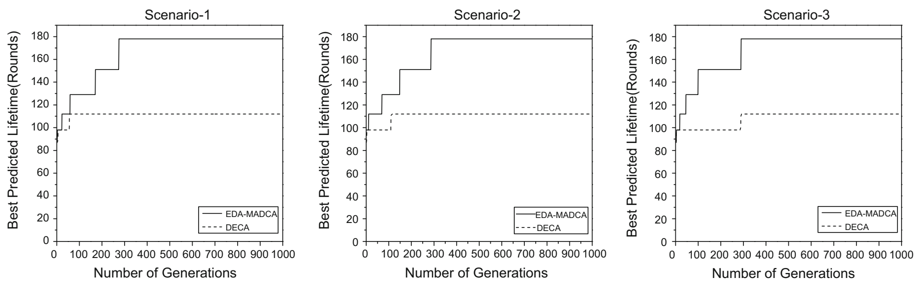

Fig. 10 Search process of EDA-MADCA and DECA for 500 sensor nodes and 50 cluster heads in three scenarios

better convergence rate in *Scenario* 1 and *Scenario* 2, respectively. Compared to DECA, EDA-MADCA can reach a better final solution within acceptable number of generations.

6 Conclusion

In this study, we have introduced a new model for LBCP in which the objective function is to maximize the overall minimum lifetime of the cluster heads in WSNs. We have presented EDA-MADCA, which combines EDA with local search in the framework of MA, to optimize the network lifetime in WSNs by load-balanced clustering. This algorithm has some prominent features: it uses a valid vector encoding to represent a clustering solution and sets up a probability matrix model to guide the individual search. Moreover, it uses a MLLS strategy to avoid invalid search. Experimental results confirms that EDA-MADCA can extend network lifetime over DECA in terms of various performance metrics. As future work, we will study multi-hop routing and multiobjective clustering problems in WSNs.

Acknowledgements This work was supported by National Natural Science Foundation of China (No. 61573277), the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 15201414), the Fundamental Research Funds for the Central Universities, the Open Research Fund of the State Key Laboratory of Astronautic Dynamics under Grant 2016ADL-DW403, and the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, the Open Projects Program of National Laboratory of Pattern Recognition. The authors would like to thank Mr. Xuan Liang and Dr. Ke Shang for their kind help and valuable suggestions. The authors are also thankful to the anonymous referees for their insightful comments and helpful suggestions which significantly improve the quality of manuscript.

Compliance with Ethical Standards

Conflict of interest The authors have no conflicts of interest to declare.

Ethical Standards We promise to comply with ethical standards. All authors have approved the manuscript and have contributed significantly for the paper.

Ethical Approval This article does not contain any studies with human participants performed by any of the authors.

Appendix

For brevity, we run EDA-MADCA and DECA with the two different objective functions at the data point which is $n = 400$ and $m = 50$. Here, we measure some key metrics, including FDCH, alive sensor nodes, BDEC and energy dissipation per round.

In Fig. 11, symbol **1** and symbol **2** denote our defined objective function and objective function of DECA, respectively. We observe that EDA-MADCA outperforms DECA with the two different objective functions in terms of FDCH, alive sensor nodes and BDEC. In our defined objective function, EDA-MADCA and DECA obtain better results, except *Scenario* 2 in Fig. 11c. In Fig. 11d, using our defined objective function, the energy dissipation is slightly higher.

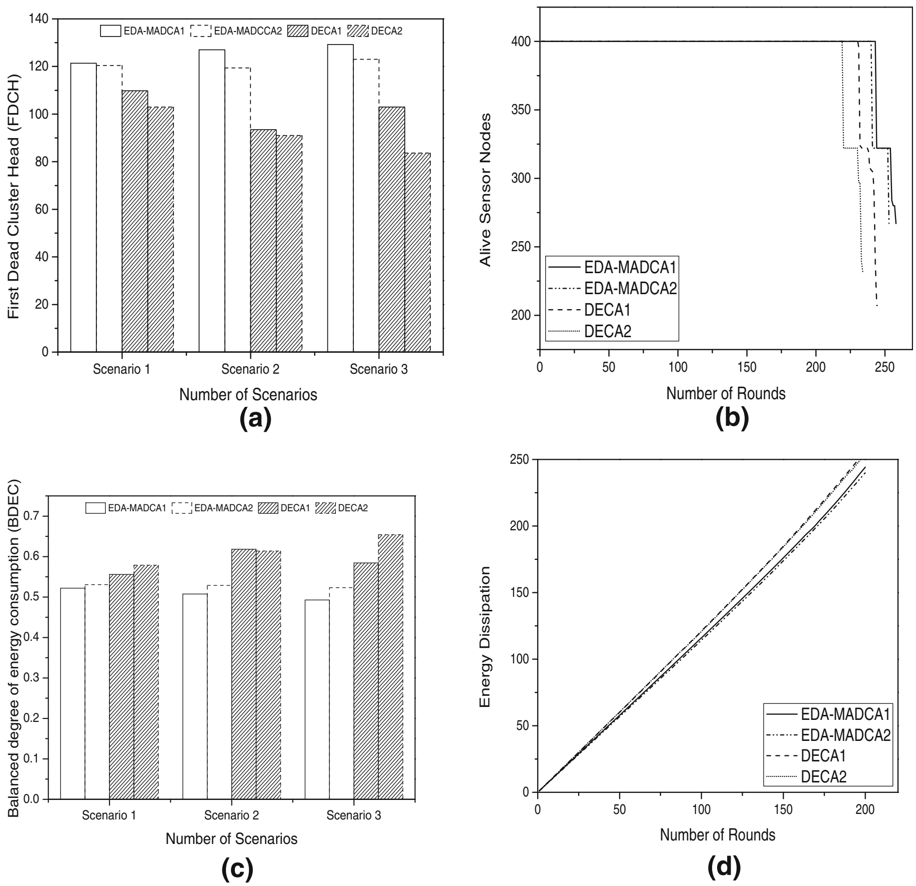

Fig. 11 Comparing EDA-MADCA to DECA for two different objective functions. **a** FDCH in three scenarios. **b** Alive sensor nodes in *Scenario* 2. **c** BDEC in three scenarios. **d** Energy dissipation in *Scenario* 3

References

1. Abbasi, A. A., & Younis, M. (2007). A survey on clustering algorithms for wireless sensor networks. *Computer Communications*, *30*(14), 2826–2841.
2. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, *38*(4), 393–422.
3. Azharuddin, M., & Jana, P.K. (2016). PSO-based approach for energy-efficient and energy-balanced routing and clustering in wireless sensor networks. *Soft Computing* 1–15.
4. Bagci, H., & Yazici, A. (2010). An energy aware fuzzy unequal clustering algorithm for wireless sensor networks. In *2010 IEEE international conference on Fuzzy systems (FUZZ)* (pp. 1–8). IEEE.
5. Bagci, H., & Yazici, A. (2013). An energy aware fuzzy approach to unequal clustering in wireless sensor networks. *Applied Soft Computing*, *13*(4), 1741–1749.
6. Baluja, S. (1994). Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning. Technical Representative, DTIC Document.
7. Baluja, S., & Caruana, R. (1995). Removing the genetics from the standard genetic algorithm. In *Machine learning: proceedings of the twelfth international conference* (pp. 38–46).
8. Bandyopadhyay, S., Coyle, E.J. (2003). An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *INFOCOM 2003, twenty-second annual joint conference of the IEEE computer and communications* (Vol. 3, pp. 1713–1723). IEEE Societies, IEEE.
9. Baranidharan, B., & Santhi, B. (2016). Ducf: Distributed load balancing unequal clustering in wireless sensor networks using fuzzy approach. *Applied Soft Computing*, *40*, 495–506.
10. Bari, A., Jaekel, A., & Bandyopadhyay, S. (2008). Clustering strategies for improving the lifetime of two-tiered sensor networks. *Computer Communications*, *31*(14), 3451–3459.
11. Bari, A., Wazed, S., Jaekel, A., & Bandyopadhyay, S. (2009). A genetic algorithm based approach for energy efficient routing in two-tiered sensor networks. *Ad Hoc Networks*, *7*(4), 665–676.
12. Baronti, P., Pillai, P., Chook, V. W., Chessa, S., Gotta, A., & Hu, Y. F. (2007). Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and zigbee standards. *Computer Communications*, *30*(7), 1655–1695.
13. Calhoun, B. H., Daly, D. C., Verma, N., Finchelstein, D. F., Wentzloff, D. D., Wang, A., et al. (2005). Design considerations for ultra-low energy wireless microsensor nodes. *IEEE Transactions on Computers*, *54*(6), 727–740.
14. Chakraborty, U.K., Das, S.K., Abbott, T.E. (2012). Energy-efficient routing in hierarchical wireless sensor networks using differential-evolution-based memetic algorithm. In *2012 IEEE Congress on Evolutionary Computation* (pp. 1–8). IEEE.
15. Chatterjee, M., Das, S. K., & Turgut, D. (2002). Wca: A weighted clustering algorithm for mobile ad hoc networks. *Cluster Computing*, *5*(2), 193–204.
16. Chen, X., Ong, Y. S., Lim, M. H., & Tan, K. C. (2011). A multi-facet survey on memetic computation. *IEEE Transactions on Evolutionary Computation*, *15*(5), 591–607.
17. Chen, X., Lei, G., Yang, G., Shao, K., Guo, Y., Zhu, J., et al. (2012). An improved population-based incremental learning method for objects buried in planar layered media. *IEEE Transactions on Magnetics*, *48*(2), 1027–1030.
18. Dietrich, I., & Dressler, F. (2009). On the lifetime of wireless sensor networks. *ACM Transactions on Sensor Networks*, *5*(1), 1–39.
19. Dombo, D.A., & Folly, K. (2015). Multi-machine power system stabilizer design based on population based incremental learning. In *2015 IEEE symposium series on computational intelligence* (pp. 1280–1285). IEEE.
20. Gupta, G., & Younis, M. (2003). Load-balanced clustering of wireless sensor networks. In *ICC'03, IEEE international conference on communications, 2003* (Vol. 3, pp. 1848–1852). IEEE.
21. Gupta, I., Riordan, D., Sampalli, S. (2005). Cluster-head election using fuzzy logic for wireless sensor networks. In *3rd Annual communication networks and services research conference (CNSR'05)* (pp. 255–260). IEEE.
22. He, Z., Wei, C., Jin, B., Pei, W., & Yang, L. (1999). A new population-based incremental learning method for the traveling salesman problem. In *Proceedings of the 1999 congress on evolutionary computation-CEC99* (Vol. 2, pp. 1152–1156). IEEE.
23. Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, *1*(4), 660–670.
24. Heinzelman, W.R., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd annual Hawaii international conference on system sciences, 2000* (pp. 1–10). IEEE.

25. Ho, S., Yang, S., & Fu, W. (2011). A population-based incremental learning vector algorithm for multiobjective optimal designs. *IEEE Transactions on Magnetics*, *47*(5), 1306–1309.
26. Ho, S. L., Zhu, L., Yang, S., & Huang, J. (2015). A real coded population-based incremental learning for inverse problems in continuous space. *IEEE Transactions on Magnetics*, *51*(3), 1–4.
27. Hussain, S., Matin, A. W., & Islam, O. (2007). Genetic algorithm for hierarchical wireless sensor networks. *Journal of Networks*, *2*(5), 87–97.
28. Kim, J.M., Park, S.H., Han, Y.J., & Chung, T.M. (2008). Chef: cluster head election mechanism using fuzzy logic in wireless sensor networks. In *ICACT 2008. 10th international conference on advanced communication technology, 2008* (Vol. 1, pp. 654–659). IEEE.
29. Kuila, P., & Jana, P. K. (2014). A novel differential evolution based clustering algorithm for wireless sensor networks. *Applied Soft Computing*, *25*, 414–425.
30. Kuila, P., Gupta, S. K., & Jana, P. K. (2013). A novel evolutionary approach for load balanced clustering problem for wireless sensor networks. *Swarm and Evolutionary Computation*, *12*, 48–56.
31. Kulkarni, R. V., Forster, A., & Venayagamoorthy, G. K. (2011). Computational intelligence in wireless sensor networks: A survey. *IEEE Communications Surveys and Tutorials*, *13*(1), 68–96.
32. Larranaga, P., & Lozano, J. A. (2002). *Estimation of distribution algorithms: A new tool for evolutionary computation* (Vol. 2). New York: Springer Science & Business Media.
33. Lindsey, S., & Raghavendra, C.S. (2002). Pegasis: Power-efficient gathering in sensor information systems. In *Aerospace conference proceedings, 2002* (Vol. 3, pp. 1125–1130). IEEE.
34. Liu, J. S., & Lin, C. H. R. (2005). Energy-efficiency clustering protocol in wireless sensor networks. *Ad Hoc Networks*, *3*(3), 371–388.
35. Low, C. P., Fang, C., Ng, J. M., & Ang, Y. H. (2008). Efficient load-balanced clustering algorithms for wireless sensor networks. *Computer Communications*, *31*(4), 750–759.
36. Lozano, J. A. (2000). Analyzing the population based incremental learning algorithm by means of discrete dynamical systems. *Complex Systems*, *12*, 465–479.
37. Lozano, J. A. (2006). *Towards a new evolutionary computation: Advances on estimation of distribution algorithms* (Vol. 192). New York: Springer Science & Business Media.
38. Martins, F. V., Carrano, E. G., Wanner, E. F., Takahashi, R. H., & Mateus, G. R. (2011). A hybrid multiobjective evolutionary approach for improving the performance of wireless sensor networks. *IEEE Sensors Journal*, *11*(3), 545–554.
39. Meng, X., Li, J., Zhou, M., Dai, X., & Dou, J. (2015). Population-based incremental learning algorithm for a serial colored traveling salesman problem. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* PP(99), 1–12.
40. Mühlenbein, H., & Paass, G. (1996). From recombination of genes to the estimation of distributions i. binary parameters. In *International conference on parallel problem solving from nature* (pp. 178–187). Springer.
41. Neri, F., & Cotta, C. (2012). Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, *2*, 1–14.
42. Nguyen, Q. H., Ong, Y. S., & Lim, M. H. (2009). A probabilistic memetic framework. *IEEE Transactions on Evolutionary Computation*, *13*(3), 604–623.
43. Qing, L., Zhu, Q., & Wang, M. (2006). Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks. *Computer Communications*, *29*(12), 2230–2237.
44. Rault, T., Bouabdallah, A., & Challal, Y. (2014). Energy efficiency in wireless sensor networks: A top–down survey. *Computer Networks*, *67*, 104–122.
45. Sabor, N., Abo-Zahhad, M., Sasaki, S., & Ahmed, S. M. (2016). An unequal multi-hop balanced immune clustering protocol for wireless sensor networks. *Applied Soft Computing*, *43*, 372–389.
46. Saleem, M., Di Caro, G. A., & Farooq, M. (2011). Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions. *Information Sciences*, *181*(20), 4597–4624.
47. Sengupta, S., Das, S., Nasir, M., Vasilakos, A. V., & Pedrycz, W. (2012). An evolutionary multiobjective sleep-scheduling scheme for differentiated coverage in wireless sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(6), 1093–1102.
48. Sert, S. A., Bagci, H., & Yazici, A. (2015). Mofca: Multi-objective fuzzy clustering algorithm for wireless sensor networks. *Applied Soft Computing*, *30*, 151–165.
49. Sim, K. M., & Sun, W. H. (2003). Ant colony optimization for routing and load-balancing: survey and new directions. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *33*(5), 560–572.
50. Singh, B., & Lobiyal, D. K. (2012). A novel energy-aware cluster head selection based on particle swarm optimization for wireless sensor networks. *Human-Centric Computing and Information Sciences*, *2*(1), 2–13.

51. Smaragdakis, G., Bestavros, A., & Matta, I. (2004). Sep: A stable election protocol for clustered heterogeneous wireless sensor networks. Technical Representative, Boston University Computer Science Department.
52. Taheri, H., Neamatollahi, P., Younis, O. M., Naghibzadeh, S., & Yaghmaee, M. H. (2012). An energy-aware distributed clustering protocol in wireless sensor networks using fuzzy logic. *Ad Hoc Networks*, *10*(7), 1469–1481.
53. Wang, G., Wang, Y., & Tao, X. (2009). An ant colony clustering routing algorithm for wireless sensor networks. In *3rd International conference on genetic and evolutionary computing, 2009. WGEC'09* (pp. 670–673). IEEE.
54. Wang, S. Y., & Wang, L. (2016). An estimation of distribution algorithm-based memetic algorithm for the distributed assembly permutation flow-shop scheduling problem. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *46*(1), 139–149.
55. Wei, D., Jin, Y., Vural, S., Moessner, K., & Tafazolli, R. (2011). An energy-efficient clustering solution for wireless sensor networks. *IEEE Transactions on Wireless Communications*, *10*(11), 3973–3983.
56. Wu, Y., Fahmy, S., Shroff, N.B. (2008). On the construction of a maximum-lifetime data gathering tree in sensor networks: Np-completeness and approximation algorithm. In *INFOCOM 2008. The 27th conference on computer communications* (pp. 1013–1021). IEEE.
57. Xing, H., & Qu, R. (2011). A population based incremental learning for network coding resources minimization. *IEEE Communications Letters*, *15*(7), 698–700.
58. Yang, S., & Yao, X. (2005). Experimental study on population-based incremental learning algorithms for dynamic optimization problems. *Soft Computing*, *9*(11), 815–834.
59. Yang, S. Y., Ho, S. L., Ni, G. Z., Machado, J. M., & Wong, K. F. (2007). A new implementation of population based incremental learning method for optimizations in electromagnetics. *IEEE Transactions on Magnetics*, *43*(4), 1601–1604.
60. Ye, M., Li, C., Chen, G., & Wu, J. (2005). Eecs: an energy efficient clustering scheme in wireless sensor networks. In *PCCC 2005. 24th IEEE international performance, computing, and communications conference, 2005* (pp. 535–540). IEEE.
61. Yigitel, M. A., Incel, O. D., & Ersoy, C. (2011). Qos-aware mac protocols for wireless sensor networks: A survey. *Computer Networks*, *55*(8), 1982–2004.
62. Younis, M., Youssef, M., & Arisha, K. (2003). Energy-aware management for cluster-based sensor networks. *Computer Networks*, *43*(5), 649–668.
63. Younis, O., & Fahmy, S. (2004). Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, *3*(4), 366–379.
64. Zungeru, A. M., Ang, L. M., & Seng, K. P. (2012). Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison. *Journal of Network and Computer Applications*, *35*(5), 1508–1536.
65. Jiao, D., Ke, L., Yang, W., & Li, J. (2017). An estimation of distribution algorithm based load-balanced clustering of wireless sensor networks. In *Computational science and engineering (CSE) and embedded and ubiquitous computing (EUC), 2017 IEEE international conference on, IEEE* (Vol. 1, pp. 151–158).

Dongbin Jiao received the B.S. degree in Information and Computing Sciences from Lanzhou University of Technology, Lanzhou, China. The M.S. degree in Software Engineering from Dalian Maritime University, Dalian, China. He is currently working towards the Ph.D. degree in Control Science and Engineering at the Xi'an Jiaotong University, Xi'an, China. His research interests include energy-efficient design, evolutionary computation and combinatorial optimization in WSNs.

Liangjun Ke received the B.Sc. and M.Sc. degrees in mathematics from Wuhan University, Wuhan, China, in 1998 and 2001, respectively, and the Ph.D. in systems engineering from Xi'an Jiaotong University, China, in 2008. He is now an associate Professor with Xi'an Jiaotong University, China. From 2011 to 2012, he visited the School of Computer Science and Electronic Engineering, University of Essex, UK. His main research interests include optimization theory and applications, especially, multiobjective optimization, evolutionary computation and robust optimization.

Weibo Yang received the B.S. degree in Automation from Xi'an Jiaotong University, Xi'an, China. He is currently working towards the Ph.D. degree in Control Science and Engineering at the Xi'an Jiaotong University, Xi'an, China. His research interests include combinatorial optimization and evolutionary computation.

Jing Li received her B.S. degree in electronic engineering from Xidian University, Xi'an, China, in 1984, and received her M.S. and Ph.D. degrees in wireless communication from National University of Defense Technology, Changsha, China in 1996 and 2005 respectively. She is currently working as a professor in the State Key Laboratory of Astronautic Dynamics, Xi'an Satellite Control Center. Her main research interests are aerocraft mission planning and new TT&C technique, inter satellite link TT&C application and space network TT&C application technique.