Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 4055 – 4069
© TÜBİTAK
doi:10.3906/elk-1904-14

A modified gravitational search algorithm and its application in lifetime maximization of wireless sensor networks

Sepehr EBRAHIMI MOOD, Mohammad Masoud JAVIDI*

Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran

Received: 01.04.2019 • **Accepted/Published Online:** 21.08.2019 • **Final Version:** 26.11.2019

Abstract: Recently, academic communities and industrial sectors have been affected by significant advancements in wireless sensor networks (WSNs). Employing clustering methods is the dominant method to maximize the WSN's lifetime, which is considered to be a major issue. Metaheuristic algorithms have attracted wide attention in the research area of clustering. In this paper, first a novel nature-inspired optimization algorithm based on the gravitational search algorithm (GSA) is defined. To control the exploitation and exploration capabilities of this algorithm, along with calculating the masses value, the tournament selection method is employed. Tournament size, the parameter of this method, is computed automatically using a function during the computational process of the algorithm. The abilities of the algorithm are balanced using this problem-independent parameter. Therefore, the performance of the proposed algorithm is improved in this paper. Moreover, a modified GSA is applied to an energy-efficient clustering protocol for WSNs to minimize the objective function defining the compact clusters that have cluster heads with high energy. The proposed search algorithm is evaluated in terms of some standard test functions. The results suggest that this method has better performance than other state-of-the-art optimization algorithms. In addition, simulation results indicate that the proposed method for the clustering problem in WSNs has better performance on network lifetime and delivery data packets in BS than other popular clustering methods.

Key words: Wireless sensor network, energy-efficiency protocol, clustering method, network life, gravitational search algorithm, tournament selection

1. Introduction

In recent years, researchers have focused on wireless sensor networks (WSNs) in both theoretical and industrial fields, as they are effective means in monitoring and tracking applications. In addition, there are other significant designations possible for WSNs in numerous ranges of applications such as classification, health care, or military [1, 2]. WSNs contain many cheap independent nodes known as sensor nodes, which are deployed manually or randomly throughout the target area. Each sensor node has units including sensing, processing, communicating, and power units. They sense and collect a variety of data such as pressure, temperature, humidity, and sound from the target area. These data are processed cooperatively and transmitted to the base station to implement appropriate decisions. However, the sensors are not accessible in most applications once they are placed in the area of interest. Therefore, it is not possible to either replace their batteries or provide additional supply sources. Hence, a significantly challenging issue in WSNs is optimizing the sensor nodes' energy conservation in the network [3].

*Correspondence: javidi@uk.ac.ir

There are several mechanisms to increase the lifetime of WSNs [4, 5]. One of the most common techniques for energy-efficient consumption in WSNs is clustering. In the clustering method, each sensor belongs to a cluster, with each cluster having a certain node named the cluster head (CH). The sensing data from each sensor are transmitted to CH nodes using single-hop communication. In CH sensors, the received data are aggregated, compressed, and sent to the base station. The clustering protocol reduces the amount of data transmitted. In [6], the low energy adaptive clustering hierarchy (LEACH) was introduced as a well-known clustering protocol. In this protocol, most sensor nodes transmit the information to the CHs, and the CHs aggregate and compress the data and forward them to the BS. Each node uses a stochastic algorithm in each round to determine whether it will become a cluster head in that round. Nodes that have been cluster heads cannot become cluster heads again for P rounds, where P is the desired percentage of cluster heads. However, the LEACH algorithm does not determine the desired number of clusters. Following this algorithm, LEACH-C was proposed in [7] as a clustering algorithm for WSNs, which improves the LEACH performance by using a centralized algorithm in the base station to form the clusters at the beginning of each round. Moreover, there are various clustering algorithms utilized in WSNs that efficiently manage the network energy consumption by organizing the nodes into clusters [8].

There are various heuristic algorithms employed in several clustering methods of WSNs [9]. For instance, in [10], a special clustering method using particle swarm optimization (PSO) was used for energy-efficient routing schema in heterogeneous WSNs. In [11], a new cost function was introduced for the WSN clustering method. A novel objective function was defined to determine the CHs and organize the nodes in some clusters. This objective function tried to minimize the intracluster distance while simultaneously optimizing the network energy consumption. PSO and a genetic algorithm (GA) were used to solve this optimization problem and find the best answers for this novel objective function. Furthermore, in [12], a novel cognitively inspired artificial bee colony clustering algorithm was presented. This algorithm was employed in cognitive WSNs to improve their energy consumption.

The gravitational search algorithm (GSA) was proposed by Rashedi et al. [13, 14] and is one of the latest evolutionary algorithms. This algorithm, inspired by Newton's gravity laws and motion, is employed to solve optimization problems. There is a large body of research aiming to improve the GSA's performance [15]. In [16], a new method was defined for mass calculation in GSA using sigma scaling and Boltzmann selection functions. Moreover, in [17], a new operator, called mutation, was added to the GSA to overcome the premature convergence problem in multimodal functions. Even though the GSA has high exploring capability, there are some problems, such as the GSA falling into local optima. In this paper, the tournament selection technique is employed to control the exploration and exploitation capabilities of the GSA and improve its performance. Then this modified version of the GSA is utilized in WSN protocols to increase the lifetime for the network.

In this paper, a novel version of the GSA is presented. In this algorithm a tournament selection mechanism is used for mass calculation instead of the raw fitness values. The exploration and exploitation abilities of the search algorithm can be controlled using this method. The performance of the proposed method is compared with state-of-the-art heuristic algorithms. The experimental results and statistical analysis reveal that the proposed method outperforms other approaches in many test functions. This novel version of the GSA is then used in WSN protocols to optimize the objective function and maximize the network lifetime. The objective function defined in this paper tries to improve the energy consumption in WSNs by determining compact clusters with cluster heads with high remaining energy. The simulation results suggest that the proposed clustering method has better performance than other popular clustering algorithms. The main contributions of this paper are as follows:

– Presenting tournament selection along with its capabilities and properties.

– Defining a novel GSA algorithm using tournament selection.

– Determining the value of tournament size, the parameter of the proposed algorithm, automatically during the computational process of the algorithm.

– Comparing the proposed method with state-of-the-art metaheuristic algorithms.

– Utilizing the proposed method on WSNs to maximize the network lifetime along with comparing the algorithm performance in this case.

The rest of this paper is organized as follows. Section 2 describes the basic concepts required to present the proposed method. Section 3 presents the proposed method and discusses its properties. Section 4 represents the adaptation and implementation of the proposed method for clustering in WSNs. Moreover, Section 5 evaluates the experimental results. Finally, Section 6, presents our conclusions and future work.

2. Basic concepts

In this section, the basic information regarding the network model, gravitational search algorithm, tournament selection, and their properties is introduced.

2.1. Network model

In this paper, we employ the network and radio model used in [11]. This network model possesses some properties as follows:

- Sensor nodes sense information received from the area of interest. Therefore, there is always information to be sent to the cluster head.
- The base station is a fixed node located outside the target area.
- All sensor locations are fixed.
- All sensors have limited battery life.
- Sensor nodes control and revise the energy consumed in data transmission according to the destination node distance.
- All sensor nodes can potentially be cluster heads.
- Sensed data are compressed to decrease the information transmitted.

The radio model is the first-order radio model proposed in [6]. Moreover, the radio channel is symmetric, which the energy consumption between two nodes' transmissions is equally bidirectionally. The transmitter and receiver components in each node consume energy to run the radio electronics and power amplifiers. In this model, the energy consumption for data transmission between nodes i and j is dependent on the distance between these nodes, which is denoted by d_{ij}. The model battery usage is d_{ij}^2 and d_{ij}^4 for short distances and long distances, respectively. Hence, the total consumed energy to transmit k bits of data over a distance d is determined as follows [6]:

$$E_T(k,d) = \begin{cases} k.E_{el} + k.\varepsilon_{fs}.d^2, & \text{if } d < d_0 \\ k.E_{el} + k.\varepsilon_{tr}.d^4, & \text{if } d \geq d_0, \end{cases} \tag{1}$$

where E_{el} is the required energy for each bit to run the receiver or transmitter, ε_{fs} and ε_{tr} are variables of the model in the employed transmitter amplifier, and d_0 is a threshold for the transmitter distance. E_R, the energy expended for receiving k bits of data, is determined as follows:

$$E_R(k) = k.E_{el}. \tag{2}$$

The model parameters in experiments and simulation results are defined as follows: $E_{el} = 50\frac{nJ}{bit}, \varepsilon_{fs} = 10\frac{pJ}{bit.m^2}$, and $\varepsilon_{tr} = 0.0013\frac{pJ}{bit.m^2}$. Furthermore, the data aggregation and compression method used in this model consumes $E_{da} = 5\frac{nJ}{bit}$ energy.

2.2. Gravitational search algorithm

Swarm intelligence and metaheuristic search algorithms are two techniques employed to solve complicated and large problems where the classical methods are not successful. Nowadays, these heuristic random search algorithms are exploited in several real-world problems such as image processing, robotics, and medicine [18]. The GSA was inspired by Newtonian laws of gravity, motion, and mass interaction. It is among the latest metaheuristic search algorithms. In this algorithm, the mass of each agent implies their performances. Therefore, heavier masses are more suitable solutions for the problems. Objects attract each other according to the gravitational force, so the global agents' movement is towards the heavier and more suitable agents. The position of the ith agent in m-dimensional search space is represented as $X_i = (x_i^1, x_i^2, \cdots, x_i^m)$. The mass value for the ith object is determined by Eq. (3), where $fit_i(t)$ represents the fitness value for the ith agent in iteration t, while worst(t) is the worst fitness value for the swarm in this iteration [13]:

$$M_i(t) = \frac{fit_i(t) - worst(t)}{\sum_{j=1}^{n}(fit_j(t) - worst(t))}. \tag{3}$$

The overall force of gravity on the ith agent in the dth dimension at iteration t is calculated as:

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i} G(t)\frac{M_i(t) \cdot M_j(t)}{R_{ij}(t) + \varepsilon}\left(x_j^d(t) - x_i^d(t)\right), \tag{4}$$

where $Kbest$ is the set of K heavier objects, which is a function of time; $R_{ij}(t)$ is the Euclidean distance between agents i and j, and ε is a small value. $G(t)$, reflecting a decreasing function of time, is the gravitational constant in the tth iteration. $G(t)$ is computed as follows:

$$G(t) = G_0 exp(-\gamma\frac{t}{t_{Max}}). \tag{5}$$

Acceleration, velocity, and the position of the ith agent in the dth dimension at time $t+1$, based on the law of motion, are computed according to the following equations:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)}, \tag{6}$$

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t), \tag{7}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1). \tag{8}$$

2.3. Tournament selection

Tournament selection is an appropriate and robust selection mechanism employed to improve some heuristic search algorithms such as genetic [19], particle swarm optimization [20], and harmony search algorithms [21]. This method's functions are described as follows. First, k (tournament size) agents were randomly selected, and then the best agent (with the heaviest mass) was selected. This procedure was iterated N times, where N is the number of agents in the algorithm, to produce a new population.

Tournament selection is utilized to redirect the search algorithm towards the most profitable areas in the search space. Therefore, the selection scheme is capable of balancing the exploitation and exploration abilities of the algorithm. In the present study, the properties and effects of tournament selection scheme on the GSA were analyzed. Thus, some notations should be defined a priori based on [19].

Definition 1 (Mass distribution and cumulative mass distribution) *The function $s : R \rightarrow Z^+$ assigns the iteration number of each population value to each mass value, where the masses' values are sorted ascendingly. Hence, m_1 is the mass for the worst agent, while m_n is the heaviest and the masses' value for the best agent in the current population. $S(m_i)$ is a cumulative mass distribution, which denotes the number of agents with mass values m_i or worse, and is determined as follows:*

$$S(m_i) = \sum_{j=1}^{i} s(m_j). \tag{9}$$

The tournament selection method function, T, transforms a mass distribution, s, into another mass distribution s' as:

$$s' = T(s). \tag{10}$$

Definition 2 (Expected mass distribution) *The expected mass distribution of tournament selection T^* on the mass distribution s is denoted by s^* and is determined according to the following equation:*

$$s^* = T^*(s). \tag{11}$$

Theorem 1 *The expected mass distribution of N agents and tournament selection T^* on mass distribution s is determined according to the following:*

$$s^*(m_i) = N\left(\left(\frac{S(m_i)}{N}\right)^k - \left(\frac{S(m_{i-1})}{N}\right)^k\right). \tag{12}$$

Proof According to Definition 1, $s^*(m_i) = S^*(m_i) - S^*(m_{i-1})$. In this equation, it is only necessary to calculate the expected number of agents with mass values of m_i or worse, $S^*(m_i)$. An agent with the mass value of m_i or worse is able to win the tournament if all k agents have the mass values of m_i or worse. Therefore, the probability of having t agents with mass values of m_i or worse should be determined, which is equal to $\left(\frac{S(m_i)}{N}\right)^k$. Moreover, this probability should be repeated N times. Hence, Eq. (12) is determined for the expected mass distribution of tournament selection T^*. □

As can be seen from Theorem 1, the tournament selection performance is highly dependent on tournament size k. For instance, if $k = 1$, there is no influence of the tournament selection on the mass distribution, and $s^*(m_i) = s(m_i)$.

3. Proposed GSA with tournament selection

Exploitation and exploration are two important issues in heuristic search algorithms. Exploration is the algorithm's capability to search the complete search space in order to find a better solution and prevent the algorithm from getting trapped in local optima solutions.Exploitation is the capability to search a limited area in the neighborhood of the best solution to perform improvement. Therefore, exploitation and exploration abilities are in contrast with one another, and exploitation is decreased whenever the exploration of the algorithm is increased and vice versa [22].

Exploitation and exploration capabilities in each algorithm must be modified over time. In a heuristic search algorithm, controlling such capabilities is a significant factor influencing the algorithm's performance. At the initial stage, it prevents the whole search space from falling into local optima. Hence, exploration capability should be high. However, towards the end of the algorithm, the exploitation capability should be high enough to enable the algorithm to converge towards the best solution.

In the GSA, the fitness function is initially distributed among the agents. Therefore, the standard deviation is large for the masses' values, which are computed by Eq. (1). Consequently, heavier masses interact with other masses and convergence occurs, and while the algorithm is converging towards the end, the masses' variance is decreased. Therefore, the masses' values are close, and they would not probe the feasible area. In this paper, tournament selection is employed to maintain the trade-off among exploitation and exploration. This method is performed as follows. In each iteration, the following step computes the value for all masses, and k agents are selected randomly. The best agent with the heaviest mass is then selected for the next population. This procedure is repeated N times, and the corresponding population is generated. Then the algorithm is continued using this new population. Two new terms, namely selection intensity and selection variance, are defined to analyze the tournament selection performance in the GSA using various values of k [19].

Definition 3 (Selection intensity) *The expected average for the masses' value in the population once selection operator T is employed on the Gaussian distribution $G(0,1)(f) = \frac{1}{\sqrt{2\pi}}e^{-\frac{f^2}{2}}$ as an initial mass distribution is called selection intensity I and is determined by:*

$$I = \int_{-\infty}^{\infty} f\overline{T^*}(G(0,1))(f)df. \tag{13}$$

Based on this definition, selection intensity is a term to compute the variations of average mass values in different iterations. On Gaussian distribution $G(0,1)$, it is defined as an initial mass distribution. The following equation needs to be solved [19] to determine the selection intensity for tournament selection with tournament size k:

$$I_T(k) = \int_{-\infty}^{\infty} kx \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \left(\int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy \right)^{k-1}. \tag{14}$$

The selection intensity values for the tournament selection can be determined using a different value for k. As demonstrated in Eq. (14), as the tournament size grows, the selection intensity is increased, which results in an increase in the masses' average value. Furthermore, according to Eq. (4), since the masses are heavier, the gravity forces are more powerful.

Definition 4 (Selection variance) *The expected variance for mass distribution of the population once selection operator T is employed on the Gaussian distribution $G(0,1)(f) = \frac{1}{\sqrt{2\pi}} e^{-\frac{f^2}{2}}$ as an initial mass distribution is called selection variance V and is determined by:*

$$V = \int_{-\infty}^{\infty} (f - I)^2 \overline{T^*}(G(0,1))(f) df. \tag{15}$$

In a similar manner to selection intensity, to determine the selection variance for tournament selection with tournament size k, the following equation needs to be solved:

$$V_T(k) = \int_{-\infty}^{\infty} k \left(x - I_T(k)\right)^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} * \left(\int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy \right)^{k-1}. \tag{16}$$

As the tournament size of the tournament selection grows, the selection variance is decreased. Therefore, the difference between the masses' values is decreased, leading to a convergence in the masses' values. The tournament selection performance is dependent on tournament size. In this paper, tournament size is defined as a time function to control the exploration and exploitation GSA capabilities in different algorithm iterations, as well as improve its performance. In the proposed algorithm, selection intensity and variance are controlled by the appropriate definition of tournament size, which is determined according to equation (17):

$$k = \ln\left(t^c\right) + 1, \tag{17}$$

where t is the iteration number of the GSA and c is a constant. In the proposed algorithm, initially t is a small number and so is k. Moreover, the masses' variance is high, the amount of gravity force is small, and the agents probe the feasible area while the exploration capability is high. The proposed algorithm reaches the end and if t has greater value then k is increased. The selection variance is decreased, while the selection intensity is increased. Therefore, the masses are attracted due to the high power of gravity force. Thus, the convergence of all the masses towards the best solution is carried out with high acceleration, which implicates the increase in exploitation power of the proposed algorithm towards the final stages. Therefore, the trade-off between exploitation and exploration capabilities of our proposed algorithm is maintained. The procedure of the modified GSA is shown in Algorithm 1.

Algorithm 1 Pseudocode of the modified GSA.

Inputs: Fitness function f, feasible search space.
Output: X (The best solution in feasible search space).
Agents are located randomly.
while stopping criterion is not satisfied **do**
    Evaluate the fitness for each agent.
    Update $G(t)$ and $worst(t)$ of the population.
    Calculate the tournament size using Eq. (17).
    Update M using tournament selection method, and a.
    Update velocity and the position of the agents.
end while
Return the best solution.

4. Adapting the proposed GSA in energy-efficient clustering for WSN

Two major constraints of WSNs are the limited nodes' batteries and irreplaceable power source. Thus, when designing protocols for WSNs, optimization of energy consumption is the main issue, and a popular solution for this issue is employing clustering techniques. In the clustering scheme, the aim is to efficiently and effectively manage the WSN energy consumption by dividing the sensors into small groups, called clusters. In addition, the aim is compressing data in clusters and transferring the compressed data to the base station using a limited number of sensors, namely cluster heads. Based on the results, clustering protocols provide two major advantages. First, the amount of information is reduced, and second, the limited number of sensors must have long distance communication with the base station.

Finding the best cluster head candidates and selecting the right cluster heads from this list influences the performance of clustering protocols. The present study attempts to employ the adaptive proposed GSA method in finding the best cluster head in the clustering algorithm. A fitness function consisting of two major parts was considered [11] to do this. The first part minimizes the distance of nodes to their corresponding cluster head in each cluster, while the second part selects the best cluster head according to the remaining energy factor of the nodes. In this paper, the proposed GSA method is utilized to optimize this fitness function, which is minimizing the cost function.

In the proposed method, the WSN protocol is divided into rounds, with each round consisting of setup phases followed by steady-state phases, similar to the protocol in [6]. At the beginning of each round in the setup phase, clusters are recognized and the cluster heads are introduced. This part of the protocol is computed in the base station, which is a node with high energy supply. This node receives all information, such as remaining energy and its location. First, the base station computes the average remaining energy for all nodes in the network and the node with remaining energy of more than the average is selected as a cluster head candidate. Then the proposed GSA method in the base station solves the optimization problem (i.e. minimizing the cost function) to determine the best cluster heads. The cost function is defined as follows [11]:

$$\cos t = \alpha \times f_1 + (1-\alpha) \times f_2, \tag{18}$$

$$f_1 = \max_{h=1,2,\cdots,H} \left\{ \sum_{\forall n_i \in C_{p,h}} d\left(n_i, CH_{p,h}\right) / \left|C_{p,h}\right| \right\}, \tag{19}$$

$$f_2 = \sum_{i=1}^{N} E\left(n_i\right) / \sum_{h=1}^{H} E\left(CH_{p,h}\right), \tag{20}$$

where the objective of the cost function f_1 is to minimize the maximum distance of nodes to their respective cluster heads, for which f_2 aims to select the best cluster head according to the remaining energy factor of the nodes. α is a constant that controls the effect of each function in the total cost function. n_i shows the ith node, d is a function that computes the Euclidean distance of nodes, and $|C_{p,h}|$ determines the number of nodes belonging to cluster C_h of the pth agent. If an agent is capable of organizing the compact clusters with high-energy cluster heads, it can be employed to optimize this cost function.

Once all clusters are organized, all cluster heads are selected in the setup phase and the information containing the ID is transmitted in the base station for the cluster head to all nodes. In each cluster, a time-division multiple access (TDMA) schedule is organized by the cluster head to avoid data collisions. Moreover, each node can switch off its radio device except during transmission time to decrease the energy consumption. When the cluster head has received all data transmitted by all nodes in each cluster, data fusion is performed in the cluster head and the data are compressed and transmitted to the base station. Algorithm 2 summarizes the procedure of the proposed method.

Algorithm 2 Pseudocode of the proposed algorithm.

```
Inputs: Location and remaining energy of sensor nodes.
while Network is alive do
    if t=setup phase then
        Compute the average remaining energy for all nodes.
        Determine the cluster head candidate set.
        solve Eq. (18) using modified GSA.
        Select the CHs and organize clusters.
    end if
    CHs receive information from sensor nodes.
    BS receives the compressed data from CHs.
    Update the remaining energy of nodes Eq. (1).
end while
```

5. Simulation and experimental results

In this section, we initially evaluate the performance of the proposed modified GSA method. Then the performance is compared with various metaheuristic search algorithms. Once the experimental results are obtained, statistical tests are employed to provide additional analysis and to compare the performance of the proposed method with other algorithms. In this paper, Friedman's method [23] is employed to analyze the results, which is a nonparametric statistical test method. Furthermore, the performance of the proposed clustering method explained in Section 4 is compared with various known energy-efficient clustering methods used in WSNs.

5.1. Results of the modified GSA

The performance of our modified GSA, which is named MO-GSA, was evaluated according to the standard benchmark function CEC [24]. The test functions in this standard were divided into three categories including unimodal functions $(F_1 - F_5)$, basic multimodal functions $(F_6 - F_{20})$, and composition functions $(F_{21} - F_{28})$. Further details and descriptions regarding these functions can be found in [24]. The performance of the new version of the GSA is evaluated on these functions, and the results are then compared with some state-of-the-art heuristic search algorithms such as joint approximation diagonalization of eigenmatrices (JADEEP) [25], gradient-based PSO (GPSO) [26], GSA [13], GA [27], cuckoo [28], and clustered-GSA (C-GSA) [29].

In GPSO, ω is set to 0.9-–0.4, while the acceleration coefficients are set to $c_1 = c_2 = 2$. The parameters of JADEEP are set as in [25]. For every type of GSA algorithm (GSA, C-GSA, and MO-GSA), the parameters are set as described in [13]. In these algorithms, G_0 is set to 100 and α is equal to 20. In the GA, $P_c = 0.3$ and $P_m = 0.1$, and $P_a = 0.25$ in the cuckoo algorithm. The parameters for the CGSA are set as described in [29]. The error values resulting from these algorithms on CEC standard test functions with dimension $50\,(n = 50)$ after $1e + 5$ fitness evaluations (FEs) are presented in Table 1. The results were obtained on a PC with a 2.6 GHz Intel Core i7-6700HQ processor and 16 GB RAM. The operating system was Windows 10 and all the codes were written and executed in MATLAB. This table illustrates the median of the error values computed through 51 independent runs of the algorithms as well as the proposed algorithm. The boldfaced values indicate the best solution for each test function.

As demonstrated in Table 1, our proposed algorithm, MO-GSA, provides the best performance for test functions $F_1, F_3, F_8, F_9, F_{15}, F_{16}, F_{17}, F_{18}, F_{19}, F_{22}, F_{23}, F_{24}, F_{25}, F_{27}$, and F_{28}. On the other hand, GPSO excels in the optimization problem for function F_7, while JADEEP derives the best solutions for functions F_1, F_4, F_5, F_{11}, and F_{26}. Moreover, cuckoo performs well for functions $F_2, F_6, F_{10}, F_{12}, F_{13}, F_{14}$, and F_{21}. This method results in good solutions for the composition functions. The performance of the GSA is best for function F_1. Furthermore, C-GSA obtains the best solutions for functions F_{19} and F_{20}. As is evident, the results of this table reveal that MO-GSA performs better compared to the other state-of-the-art optimization algorithms, especially for multimodal and complex functions. This superiority is enabled by controlling the exploitation and exploration capabilities, performed using the tournament selection method.

In this paper, the Friedman test is employed to make comparisons and to assign rankings to the performances of different algorithms. The Friedman test is a nonparametric statistical test that does not make assumptions on the population distribution parameters of the data. In this method, the significance level is set to 0.1, where the null hypothesis is that the average performances of the algorithms are equal, while the alternative hypothesis suggests otherwise. r_i^j is considered as the rank of the jth of k methods on the ith of n benchmark functions such that ranking 1 is assigned to the best of them and ranking k to the worst. The Friedman test needs the computation of the average ranks of different approaches, $R_j = \frac{1}{n}\Sigma_i r_i^j$. Under the null hypothesis, the Friedman statistic is calculated with the following equation:

$$\chi_F^2 = \frac{12n}{k(k+1)}[\Sigma_j R_j^2 - \frac{k(k+1)^2}{4}]. \tag{21}$$

This statistic is distributed according to χ_F^2 with $k-1$ degrees of freedom, where n and k are big enough. In addition, the STAC platform [30] was utilized to perform statistical tests. Table 2 illustrates the Friedman test results to minimize the results of Table 1. The rank values in Table 2 indicate that the proposed method leads to better performance among other algorithms. The measures and the P-value for this test are presented in the last row of Table 2. Hence, the results of this Table reveal that MO-GSA performs better than the other optimization algorithms, especially in multimodal and complex functions. This superiority is enabled

Table 1. Minimization results on test functions.

	GPSO	JADEEP	GSA	GA	Cuckoo	C-GSA	MO-GSA
F1	5.29E+03	**0.00E+00**	**0.00E+00**	5.14E+02	8.93E-12	1.42E-12	**0.00E+00**
F2	8.34E+07	5.43E+03	1.88E+06	9.06E+02	**8.17E+02**	2.09E+06	3.41E+06
F3	9.81E+10	6.19E+06	1.27E+08	4.70E+10	8.93E+06	1.37E+08	**3.20E+06**
F4	1.58E+04	**5.00E+03**	1.75E+04	6.20E+04	1.70E+04	1.88E+04	8.72E+03
F5	1.59E+03	**0.00E+00**	5.74E-05	8.70E+01	6.31E-03	5.97E-05	2.20E-05
F6	4.88E+02	9.09E-01	4.97E+01	3.02E+02	**1.14E-10**	5.97E-05	2.13E-03
F7	**1.63E+02**	4.67E+00	2.17E+01	2.14E+02	6.39E+00	1.83E+01	4.32E+00
F8	2.12E+01	2.09E+01	2.04E+01	2.18E+01	2.06E+01	2.10E+01	**2.02E+01**
F9	4.43E+01	2.69E+01	4.14E+00	1.34E+01	3.12E+00	4.16E+00	**2.65E+00**
F10	1.45E+03	3.75E-02	1.03E-02	1.08E+00	**5.13E-04**	6.62E-03	2.34E-03
F11	1.56E+02	**0.00E+00**	2.53E+01	1.74E+02	9.07E-12	2.60E+01	3.14E+01
F12	3.73E+02	2.06E+01	2.40E+01	1.64E+02	**8.15E+00**	2.30E+01	1.12E+01
F13	5.87E+02	4.16E+01	4.46E+01	2.73E+02	**7.80E+00**	4.70E+01	2.43E+01
F14	2.59E+03	4.39E-02	8.99E+02	9.47E+02	**1.43E-06**	8.82E+02	6.54E+00
F15	7.76E+03	3.20E+03	4.91E+02	8.60E+02	4.38E+03	4.89E+02	**3.73E+02**
F16	2.09E+00	1.75E+00	1.56E-02	2.29E+00	3.18E-04	1.49E-02	**8.38E-04**
F17	3.46E+02	3.04E+01	1.28E+01	6.89E+01	7.92E+01	1.30E+01	**1.26E+01**
F18	3.45E+02	7.31E+01	1.28E+01	4.35E+01	1.80E+01	1.35E+01	**1.23E+01**
F19	4.36E+04	1.43E+00	1.22E+00	1.40E+02	9.61E+00	**1.14E+00**	**1.14E+00**
F20	2.22E+01	1.01E+01	4.08E+00	8.84E+00	3.38E+01	**4.70E+00**	5.12E+00
F21	9.33E+02	2.98E+02	7.80E+02	4.00E+02	**1.20E+02**	4.00E+02	4.00E+02
F22	4.25E+03	1.93E+03	1.93E+03	3.61E+03	6.18E+03	1.98E+03	**1.76E+03**
F23	1.06E+04	3.25E+03	1.28E+03	3.19E+03	5.24E+03	1.33E+03	**1.27E+03**
F24	3.37E+02	2.10E+02	2.20E+02	2.75E+02	7.36E+02	2.24E+02	**1.56E+02**
F25	4.81E+02	2.63E+02	2.15E+02	3.06E+02	2.40E+02	2.15E+02	**1.87E+02**
F26	4.17E+02	**2.09E+02**	3.25E+02	1.18E+03	3.27E+02	3.82E+02	2.41E+02
F27	1.68E+03	5.34E+02	4.00E+02	6.85E+02	4.01E+02	4.00E+02	**3.31E+02**
F28	4.33E+03	3.00E+02	6.46E+02	7.28E+02	2.63E+02	6.38E+02	**2.42E+02**

by controlling the exploitation and exploration capabilities, performed using a tournament method for mass calculation.

Figure 1 provides a performance comparison between the proposed method and the standard GSA algorithm for functions F_2 and F_{23}. Based on these graphs, the standard GSA performance can be improved by employing a tournament method to perform the mass calculation. Moreover, the proposed algorithm can control and balance exploration and exploitation, as well as increase its performance. According to the figure and the results of Tables 1 and 2, the proposed method provides good performance, while the tournament selection can improve the performance of the GSA. In our proposed method, we initially determine the tournament size, because the selection variance is high while the selection intensity is low. Hence, the agents can explore the feasible area, which heightens the exploration capability of the algorithm. On the other hand, towards the end of the algorithm, the tournament size and agents were calculated while decreasing the selection variance

Table 2. Statistical analysis.

Algorithms	Rank
MO-GSA	2.1964
Cuckoo	2.4713
JADEEP	3.6964
GSA	3.8035
CGSA	4.1607
GPSO	5.2978
GA	6.4205
Statistic	P-value
33.2770	le-5 = Rejected

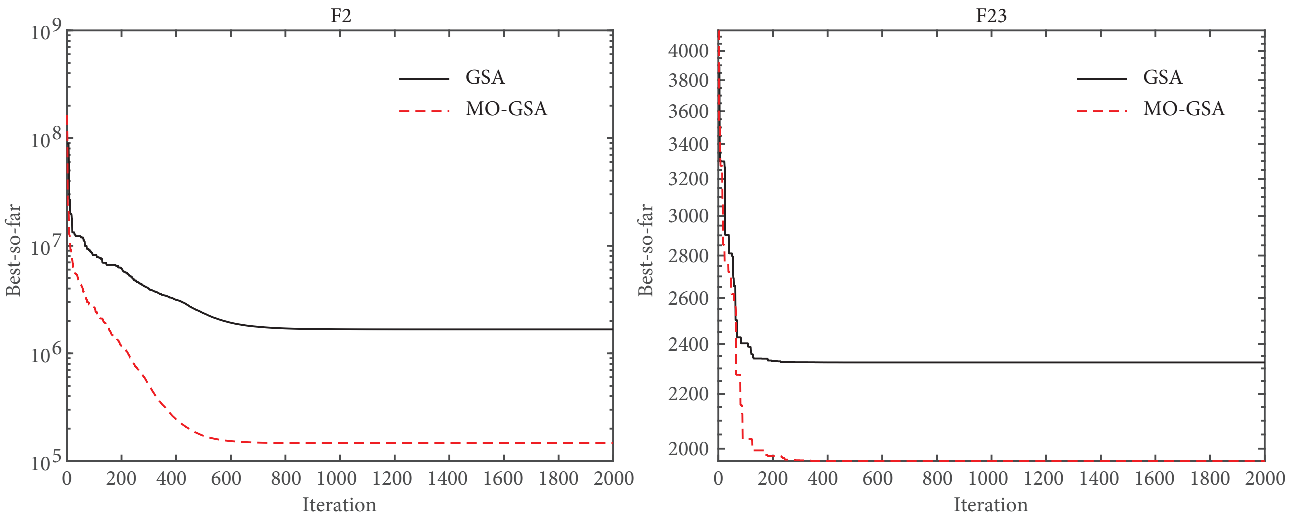
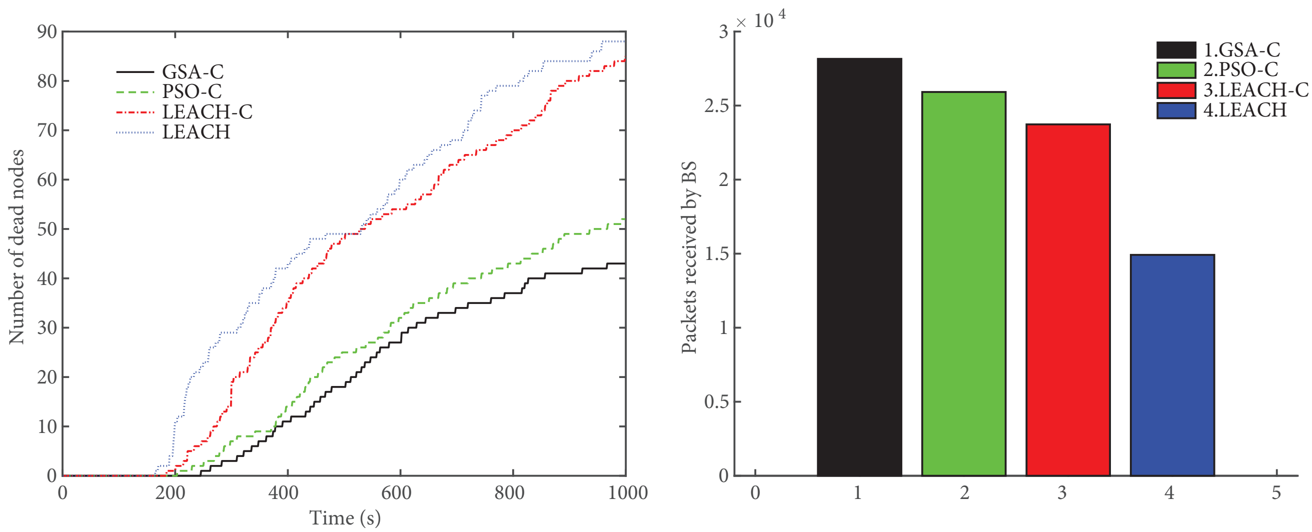


Figure 1. Comparison of performance of the proposed method and GSA for functions F_2 and F_{23}.

and increasing the selection intensity. Therefore, agents converged to the best solution, which heightens the exploitation capability of the algorithm.

5.2. Results of the clustering method

In the following section, the performance of the proposed method in an energy-efficient clustering method on WSNs was evaluated using MATLAB software. The simulation was performed in a 100 m $\times$ 100 m area with 100 sensor nodes scattered randomly in the area. The initial amount of energy for these sensors is 0.5 J. In addition, the number of clusters, H, is set to 10. In this section, the proposed protocol, GSA-C, is compared with mainstream clustering protocols employed in WSNs, such as LEACH [6], LEACH-C [7], and the PSO-C method [11]. The base station is located far outside the area in location $(50,175)$ and the simulation is performed until all the sensors in the network are dead and consume their energy. Moreover, the length of the packet sent from the cluster head to base station is considered as 6400 bits, and the length of the packet sent from the cluster nodes to their respective cluster heads is considered to be 200 bits. The GSA parameters are set according to [13]. Furthermore, we set $c = 5$ in the tournament selection scheme and $\beta = 0.5$ in the GSA-C protocol. For the PSO-C method, the PSO parameters are similar to those of [11]. Finally, we consider $p = 0.1$ for the LEACH method.



(a) Number of dead nodes over time

(b) Total data packets received by the BS

Figure 2. Comparison of performance of the proposed method and other clustering methods.

In Figure 2, the sensor network lifetime is evaluated in terms of the number of the dead nodes. It is concluded that the performance of the proposed method is better than PSO-C, LEACH, and LEACH-C. This figure shows that the network lifetime would increase by 18 percent when the proposed method is employed. On the other hand, Figure 2 shows the total data number of messages and packets received by the base station. Also, this line chart indicates the effectiveness of the proposed method in delivering a larger number of data packets to the base station.

The reason for this significant achievement in performance is the modification imposed on the GSA and its performance. The tournament selection method, as demonstrated in Sections 2 and 3, controls the exploration and exploitation of the search algorithm. In addition, this algorithm can balance the agents' selection intensity and variance using an appropriate definition for the tournament size during the computation. Consequently, the modified GSA can derive better solutions for the optimization problem. Furthermore, better nodes were determined as cluster heads by employing this method, and the network architecture was defined in a better form. Therefore, compact clusters with higher-energy nodes as their heads were organized. Hence, the network retains more energy during data transitions. Accordingly, the proposed method outperforms other mainstream clustering methods.

6. Conclusions and future work

In this paper, a novel GSA method was proposed. In this algorithm, the tournament selection method was employed to determine the values of the masses. Tournament size, the parameter of this algorithm, was determined using a function during the computational process of the algorithm. Therefore, tournament size was a problem and user-independent parameter. We proved that this algorithm can control the exploitation and exploration abilities of the method using this parameter. Therefore, the anticipated performance of the algorithm was achieved. In this paper, the new version of the GSA was applied to energy-efficient clustering protocols employed in WSNs to minimize the objective function. This objective function determines the compact clusters with high-energy cluster heads. This method grouped all the sensor nodes into clusters automatically. The observed experiments and statistical analyses indicate that the performance of the modified GSA is better than other state-of-the-art heuristic search algorithms. Thus, the tournament selection is a remarkable method to improve the results despite its simplicity in implementation. Furthermore, the proposed clustering method was evaluated and the simulation results reveal that the proposed clustering method had good performance for network lifetime and delivery of data packets in the BS. The results demonstrated that the proposed clustering method outperforms other popular energy-efficient clustering methods for WSNs.

There are some parameters in the proposed optimization algorithm that are set for the energy-efficient clustering problem. A possibility for future work is to introduce a controller or a function that can determine the value of the algorithm parameters user- and problem-independently. The exploration and exploitation abilities of the algorithm can be balanced using this strategy. Moreover, time complexity and computational complexity of metaheuristic algorithms such as the proposed method may be an interesting topic for future research; we intend to explore these topics in future work.

References

[1] Kuorilehto M, Hännikäinen M, Hämäläinen TD. A survey of application distribution in wireless sensor networks. EURASIP Journal on Wireless Communications and Networking 2005; 5 (1): 2688-2710. doi: 10.1155/WCN.2005.7

[2] Tong Y, Tian L, Li J. Novel node deployment scheme and reliability quantitative analysis for an IoT-based monitoring system. Turkish Journal of Electrical Engineering and Computer Sciences 2019; 27 (3): 2052-2067.

[3] Rahbari D, Nickray M. Low-latency and energy-efficient scheduling in fog-based IoT applications. Turkish Journal of Electrical Engineering and Computer Sciences 2019; 27 (2): 1406-1427. doi:10.3906/elk-1810-47

[4] Kumar V, Kumar A. Improving reporting delay and lifetime of a WSN using controlled mobile sinks. Journal of Ambient Intelligence and Humanized Computing 2019; 10 (4): 1433-1441. doi: 10.1007/s12652-018-0901-5

[5] Edla DR, Kongara MC, Cheruku R. A PSO based routing with novel fitness function for improving lifetime of WSNs. Wireless Personal Communications 2019; 104 (1): 73-89. doi: 10.1007/s11277-018-6009-6

[6] Heinzelman WR, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences; Maui, HI, USA; 2000. pp. 10-17. doi: 10.1109/HICSS.2000.926982

[7] Muruganathan SD, Ma DC, Bhasin RI, Fapojuwo AO. A centralized energy-efficient routing protocol for wireless sensor networks. IEEE Communications Magazine 2005; 43 (3): S8-13. doi: 10.1109/MCOM.2005.1404592

[8] Pradhan N, Sharma K, Singh VK. A survey on hierarchical clustering algorithm for wireless sensor networks. International Journal of Computer Applications 2016; 134 (4): 30-35.

[9] Rostami AS, Badkoobe M, Mohanna F, Hosseinabadi AA, Sangaiah AK. Survey on clustering in heterogeneous and homogeneous wireless sensor networks. Journal of Supercomputing 2016; 74 (1): 277-323. doi: 10.1007/s11227-017-2128-1

[10] Wang J, Gao Y, Liu W, Sangaiah AK, Kim HJ. An improved routing schema with special clustering using PSO algorithm for heterogeneous wireless sensor network. Multidisciplinary Digital Publishing Institute 2019; 19 (3): 671-692. doi: 10.3390/s19030671

[11] Latiff NA, Tsimenidis CC, Sharif BS. Energy-aware clustering for wireless sensor networks using particle swarm optimization. In: 2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications; Athens, Greece; 2007. pp. 1-5. doi: 10.1109/PIMRC.2007.4394521

[12] Kim SS, McLoone S, Byeon JH, Lee S, Liu H. Cognitively inspired artificial bee colony clustering for cognitive wireless sensor networks. Cognitive Computation 2017; 9 (2): 207-224. doi: 10.1007/s12559-016-9447-z

[13] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: A gravitational search algorithm. Information Sciences 2009; 179 (13): 2232-2248. doi: 10.1016/j.ins.2009.03.004

[14] Rashedi E, Nezamabadi-Pour H, Saryazdi S. BGSA: Binary gravitational search algorithm. Natural Computing 2010; 9 (3): 727-745. doi: 10.1007/s11047-009-9175-3

[15] Rashedi E, Rashedi E, Nezamabadi-Pour H. A comprehensive survey on gravitational search algorithm. Swarm and Evolutionary Computation 2018; 41: 141-158. doi: 10.1016/j.swevo.2018.02.018

[16] Mood SE, Rasshedi E, Javidi MM. New functions for mass calculation in gravitational search algorithm. Journal of Computing and Security 2016: 2 (3); 233–246.

[17] Kherabadi HA, Mood SE, Javidi MM. Mutation: A new operator in gravitational search algorithm using fuzzy controller. Cybernetics and Information Technologies 2017; 17 (1): 72-86. doi: 10.1515/cait-2017-0006

[18] Elbes M, Alzubi S, Kanan T, Al-Fuqaha A, Hawashin B. A survey on particle swarm optimization with emphasis on engineering and network applications. Evolutionary Intelligence 2019; 2019: 1-17. doi: 10.1007/s12065-019-00210-z

[19] Blickle T, Thiele L. A mathematical analysis of tournament selection. International Computer Games Association 1995; 1995: 9-16.

[20] Angeline PJ. Using selection to improve particle swarm optimization. In: 1998 IEEE International Conference on Evolutionary Computation Proceedings; Anchorage, AK, USA; 1998. pp. 84-89. doi: 10.1109/ICEC.1998.699327

[21] Karimi M, Askarzadeh A, Rezazadeh A. Using tournament selection approach to improve harmony search algorithm for modeling of proton exchange membrane fuel cell. International Journal of Electrochemical Science 2012; 7: 6426-6435.

[22] Patrick SC, Pinaud D, Weimerskirch H. Boldness predicts an individual's position along an exploration–exploitation foraging trade-off. Journal of Animal Ecology 2017; 86 (5): 1257-1268. doi: 10.1111/1365-2656.12724

[23] Friedman M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association 1937; 32 (200): 675-701.

[24] Li X, Tang K, Omidvar MN, Yang Z, Qin K. Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. Gene 2013; 7 (33): 8.

[25] Zhang J, Sanderson AC. JADE: Adaptive differential evolution with optional external archive. IEEE Transactions on Evolutionary Computation 2009; 13 (5): 945-958. doi: 10.1109/TEVC.2009.2014613

[26] Noel MM. A new gradient based particle swarm optimization algorithm for accurate computation of global minimum. Applied Soft Computing 2012; 12 (1): 353-359. doi: 10.1016/j.asoc.2011.08.037

[27] Baroudi U, Bin-Yahya M, Alshammari M, Yaqoub U. Ticket-based QoS routing optimization using genetic algorithm for WSN applications in smart grid. Journal of Ambient Intelligence and Humanized Computing 2019; 10 (4): 1325-1338. doi: 10.1007/s12652-018-0906-0

[28] Rajabioun R. Cuckoo optimization algorithm. Applied Soft Computing 2011; 11 (8): 5508-5518. doi: 10.1016/j.asoc.2011.05.008

[29] Shams M, Rashedi E, Hakimi A. Clustered-gravitational search algorithm and its application in parameter optimization of a low noise amplifier. Applied Mathematics and Computation 2015; 258: 436-453. doi: 10.1016/j.amc.2015.02.020

[30] Rodríguez-Fdez I, Canosa A, Mucientes M, Bugarín A. STAC: a web platform for the comparison of algorithms using statistical tests. In: 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE); İstanbul, Turkey; 2015. pp. 1-8. doi: 10.1109/FUZZ-IEEE.2015.7337889