IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 6, 2020, accepted March 20, 2020, date of publication March 26, 2020, date of current version April 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2983483

CSOCA: Chicken Swarm Optimization Based Clustering Algorithm for Wireless Sensor Networks

WALID OSAMY[1], AHMED A. EL-SAWY[1], AND AHMED SALIM[2,3]
[1]Computer Science Department, Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt
[2]Mathematics Department, College of Sciences, Zagazig University, Zagazig 44516, Egypt
[3]Department of Computer Science, College of Sciences and Arts in Al-Methnab, Qassim University, Buridah 51931, Saudi Arabia

Corresponding author: Ahmed A. El-Sawy (ahmed.el_sawy@fci.bu.edu.eg)

ABSTRACT The rapid advancement in Wireless Sensor Network (WSN) technology has enabled smart environments to provide ubiquitous real-time applications in various fields such as industry, smart city, transport, health and Internet of Things (IoT). Energy is the most significant resource in WSNs as it has a direct effect on their lifetime. The efficient use of energy is required for the lifetime extension of WSNs. One of the well-known methods for achieving high scalability and efficient resource allocation in WSN is a clustering of sensor nodes. In this paper, the Chicken Swarm Optimization based Clustering Algorithm (CSOCA) is proposed to improve energy efficiency in WSNs. The chicken swarm optimization is discretized by applying a sigmoid function to individuals. Moreover, we proposed CSOCA with Genetic Algorithm (CSOCA-GA) which is an improvement to CSOCA by employing the Genetic Algorithm's processes in CSOCA. CSOCA-GA utilizes crossover and mutation processes for individuals with low fitness value to extend the population diversity. CSOCA and CSOCA-GA are tested and compared with other similar algorithms to confirm their effectiveness in terms of extending WSN lifetime and reducing energy consumption.

INDEX TERMS Wireless sensor network, clustering, Internet of Things, genetic algorithm, chicken swarm optimization.

I. INTRODUCTION

The growing importance of Internet of Things (IoT) applications in everyday life has revolutionized the lifestyle choices of people. Many such IoT based applications require node position and node location for efficient communication of data between nodes [1]. WSN is known as the major part of IoT and has many vast applications in IoT [2], [3]. WSNs are expected to be integrated into the IoT, in which the sensor nodes connect to the internet dynamically and accomplish their given tasks. WSNs can be integrated with Socially Aware Networking(SAN) [5]–[7] which leads to various applications(e.g., advanced feedbacks about fire occurrence). Moreover, the combination of WSN and the Internet of Vehicles (IoV) [4] can get a better understanding of the surrounding environment to prevent any hazardous situations. This makes WSNs have many potential applications in military, science, engineering, health care, environment, home applications, area monitoring, forest fire detection, landslide detection, and earthquake prediction, etc. WSN has to face various constraints due to its limited resources such as limited computation and processing, limited energy, limited memory, limited storage, and limited communication capacity [8]. As a result, the intra-WSN data traffic should be minimized; otherwise, the availability of the IoT may not be properly utilized due to the limited lifetime of the WSN.

The clustering scheme is often used to divide sensor nodes into groups as it has many benefits such as scalability, resource sharing, energy saving, reduction in communication overheads, and effective resource allocations.

In the process of clustering, the sensor member nodes with a specially designated cluster head (CH) constitute a cluster. In specific, the role of CH concentrates on the coordination of the sensor member nodes of each cluster node with the other sensor nodes associated with the other clusters for the objective of sending the data to the sink or base station(BS).

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Guo.

Clustering schemes utilize data aggregation methods [9] which decrease the collected data at CH in the form of considerable information. Then, the CHs transmit the aggregated data to the BS.

On the other hand, the problem of unbalanced energy necessitates effective exploitation in the sensor nodes of the clusters for maximizing the partitioning process in order to sustain the challenges that are imposed by the destabilized characteristics of energy in the network. Further, the problem of CH selection in sensor network is a NP problem since the energy balanced optimal data aggregation cannot take place in a polynomial time [10]. Thus, a meta-heuristic optimization algorithms are determined to be potent in facilitating efficient CH selection process [11].

Chicken Swarm Optimization (CSO) algorithm is a bio-inspired algorithm [12], [13], which simulated foraging activities of the chicken swarm by divided chickens into different subgroups. In each subgroup, every individual iterated simultaneously towards the optimal one. Thus, CSO motivated the idea of using it in the process of cluster head selection. The main contribution of this paper includes the following points:

- Modified CSO is utilized to optimize the CHs selection in WSNs and to minimize consumed energy.
- Provide a hybrid algorithm (CSOCA-GA) that employs the Genetic Algorithm's processes into CSO to increase the population diversity and increase the chance of escaping from local solutions.
- A fitness function is formulated that considers not only minimizing energy consumption but also concentrating on balancing the consumed energy by utilizing the CHs rotation factor.
- Provide extensive simulation results to prove that CSOCA and CSOCA-GA can extend network lifetime and outperforms the other related algorithms in terms of energy consumption, and the network lifetime.

The rest of the paper is organized as follows: Section II reviews related work in brief. The system model is described in Section III. In Section IV, a brief definition for CSO algorithm is presented. Our approach is introduced in Section V. In Section VI, the simulation results of our approach are presented. In Section VII, we conclude our work. Used notations and abbreviations through the paper are given in Table 1.

TABLE 1. Table of abbreviations and notations.

Notation / Abbreviation	Definition
N_r	Number of Roosters
N_h	Number of Hens
N_c	Number of Chicks
G	Update time steps
X	Population
BS	Base station
CH	Cluster head
CM	Cluster member
$popSize$	Population size
t_{max}	Number of Iterations
sigmf(x)	Sigmoid function
F	Fitness function

II. RELATED WORK

Extending network lifetime in WSNs is a very important problem as a result researchers undertook this problem from different perspectives. One of these perspectives is clustering. According to the assistance of Artificial Intelligence(AI) techniques, clustering algorithms in WSN can be generally classified into two categories: AI-based methods or non-AI based methods.

A. NON-AI-BASED CLUSTERING METHODS

Low Energy Adaptive Clustering Hierarchy (LEACH) [14] is the first and widely accepted clustering protocol that extends network lifetime. The selection operation of CHs in LEACH is done in a random way which leads to uneven CH distribution inside the network, as a result, the network performance is degraded. In [22] CH threshold equation in LEACH optimized based on Distributed Address Assignment Mechanism (DAAM) of ZigBee by considering the node's network address and remaining energy. Another improved LEACH called IBLEACH (intra-balanced LEACH) presented in [23], which balances the energy consumption in the LEACH protocol.

Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [15] is one of the most known chain-based clustering protocols. PEGASIS constructs a chain based on a greedy algorithm and then each node in the constructed chain takes its turn as a chain leader node. Enhanced PEGASIS (EPEGASIS) algorithm was proposed in [16] to improve the performance of PEGASIS by utilizing sink mobility.

Entropy-based clustering scheme presented in [26]. This clustering scheme takes the benefits from the nodes' local information (such as remaining energy, density, and distance between the node and the BS) and uses this information (measured in terms of entropy) as criteria for CH selection and cluster formation. The work in [18] extends the entropy-based clustering scheme [26] by adopting a compressive sensing technique at CHs and employ tree structure as the backbone of the network.

Another schema proposed in [17] called Asynchronous Clustering and Mobile Data Gathering schema based on the Timer Mechanism (ACMDGTM). ACMDGTM considers the node's location information and remaining energy as criteria for CHs selection. Moreover, ACMDGTM utilizes a single mobile sink for data gathering from CHs.

In [29] two deployment scenarios, a 2d homogeneous spatial Poisson point process was used for the random deployment model for homogeneous sensor networks and deterministic deployment. In these scenarios average energy spend in WSN at each round is calculated in order to find the optimal probability of a node to a CH thereby the optimal number of CHs is determined.

Affinity propagation-based self-adaptive (APSA) clustering technique is presented in [19]. In this technique, the optimal number of clusters and initial cluster centers are determined by employing an affinity propagation algorithm, then based on these initial cluster centers, modified K-medoids method is utilized to construct the final clusters.

B. AI-BASED CLUSTERING METHODS

Artificial intelligence includes a number of techniques (e.g., Particle Swarm Optimization, Neural Networks, Genetic Algorithms, and Ant Colony Optimization) that help to improve the performance of the WSNs. These techniques may be used at different stages in WSNs.

An optimal probability of CHs in LEACH is estimated using a genetic algorithm in [21]. Moreover, an integration between LEACH and Particle Swarm Optimization (PSO) named LEACH-PSO proposed in [24]. Cluster selection in LEACH-PSO is done based on energy consumption. Monkey Search (MS) Algorithm integrated with LEACH (LEACH-MS) to create a hybrid algorithm [25]. In LEACH-MS, the network operations start based on LEACH followed by Monkey Search Algorithm for the remaining operations, which is to take the benefits from both algorithms.

A PSO based clustering scheme is developed in [27] to address the hot spot problems by using uneven clustering. This scheme divides into two phases, routing phase and clustering phase. In the routing phase, load balancing is achieved between CHs. While, in the clustering phase the CH lifetime is improved by allocating only fewer nodes.

In [28], PSO-HSA (Particle Swarm Optimization-Harmony Search Algorithm) algorithm is proposed, where PSO is used for optimizing the CH selection, and HSA is adopted for choosing the best route. The PSO-HSA technique is therefore efficient as it selects the optimal cluster and route by combining the benefits of both PSO and HSA.

Fuzzy-based clustering protocol is proposed in [30] which determines the cooperative node (CN) that joins a cluster and establishment of a communication path between a CN and CH is done using PSO. Shuffled Frog-leaping and Firefly Algorithms (SFFA) [31] is a clustering protocol for WSNs. SFFA considers different criteria s(CHs' distances from the BS, residual energy of nodes, inter and intra-cluster distances and a load of clusters) as the multi-objective fitness function to select the most proper CHs at each round.

A clustering algorithm based on PSO and Manhattan distance called EODC (Energy Optimized Dynamic Clustering) proposed in [32]. In EODC, the fitness function formulated based on remaining energy, node location, and link quality. Moreover, EODC uses the shortest path approach between CHs and BS.

Lately, a novel swarm intelligence (SI) algorithm called Chicken Swarm Optimization (CSO) [12] imitates the shape of the movement and the behaviors in the chicken swarm. In CSO, individuals adopt diverse evolutionary methods based on their fitness values, which are absent in most traditional evolutionary computation (EC) and SI algorithms such as genetic algorithms (GA), differential evolution (DE), and PSO. Statistical analyses on twelve benchmark problems demonstrate CSO dominance in terms of precision, robustness, and performance. There have also been several efforts to further boost CSO performance [13], [34], [35].

In WSNs, CSO has been used to address various problems. A SI optimization algorithm called cuckoo search chicken swarm optimization (CSCSO) is proposed in [34] for optimal selection of the sensor nodes to form a virtual node antenna array. LEACH is improved in [36] by utilizing CSO to find the optimal path of data transfer between CHs and BS. CSO adopted to solve WSN localization problems in [37], [38]. The work in [39], [40] proposed an efficient compressive sensing matrix optimization algorithm (called CSMO-CSO) to optimize the compressive sensing matrix using CSO. In this paper, CSO is adapted and modified to select the best group of nodes to work as cluster heads at each round considering minimizing total energy consumption per round and therefore prolonging the lifetime in the sensor networks.

III. SYSTEM MODEL

A. NETWORK MODEL

The network model can be described as n sensors are deployed randomly in a region R with a specific dimension $M \times M$ with fixed BS. Every sensor has a distinct ID and has information (IDs and coordinates) of its communicating neighbors which can be obtained using *hello* message. We consider the following assumptions that employed in many related works [16], [17], [19] [20]:-

1) Sensor nodes have the same initial energy (i.e., homogeneous) and they are non-rechargeable.
2) Ideal communication channels i.e., no collision occurring during transmission processes.
3) Sensor nodes are immobile and can be identified by their unique ID.
4) Sensor nodes collect environmental information (e.g., temperature and humidity) and transmit the collected data to their respective head nodes.

B. ENERGY MODEL

Here, we consider energy costs for data transmission of each node in the cluster, as well as, consumed energy for data receiving, processing, and transmission of each CH. While energy costs for environmental sensing are not considered because these energy costs are generally much less than communication and processing costs. Furthermore, the power used for transmission is considered to vary depending on the distance of the devices under communication. The energy model in [41] is used, since it is the widely accepted model incorporated in most of the clustering schemes. In this model the energy consumption for sending a packet is:

$$E_{Tx}(b, d) = E_{elec} \times b + \upsilon \times b \times d^{p}. \quad (1)$$

and to receive this packet is:

$$E_{Rx}(b, d) = E_{elec} \times b. \quad (2)$$

Here, E_{elec} is the electronics energy, b the packet size and d the transmission distance. υ denotes the expended amplification energy based on d to overcome multi-path/free space loss. The propagation loss is inversely proportional to d^2 or d^4 for shorter or longer distance respectively and p ($2.0 \leq p \leq 4.0$) is the path loss factor.

IV. BACKGROUND ON CHICKEN SWARM OPTIMIZATION

The CSO simulate the chickens' movement and the behavior of the chicken swarm, the CSO can be described as follows: In CSO there are many groups and each group consisting of a dominant rooster, a few of hens, and chicks. Roosters, hens, and chicks in the group are determined based on their fitness values. Roosters (group head) are the chicken that has the best fitness values. While chicks are the chickens that have the worst fitness values. The majority of the chickens would be the hens and they choose randomly which group to stay in. In fact, the mother-child relationship between the hens and the chicks is performed arbitrarily. The dominance relationship and mother-child relationship in a group will stay unaltered and updated every several (G) time steps. The flowchart of CSO is as shown in Figure 1. The movement of the chickens can be formulated below:

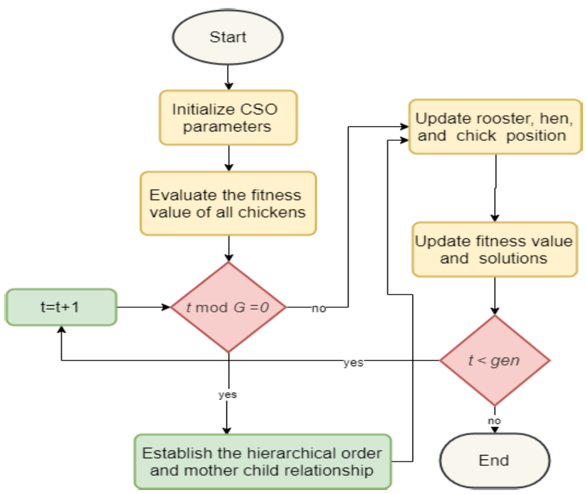

FIGURE 1. **CSO flow diagram [13].**

1) **The formula that used for the roosters' position update is given by** Eq.3:

$$X_{i,j}^{t+1} = X_{i,j}^{t} * (1 + randn(0, \sigma^2)) \quad (3)$$

where

$$\sigma^2 = \begin{cases} 1 & \text{if } f_i \leq f_k \\ \exp(\frac{f_k - f_i}{|f_i + \varepsilon|}) & \text{Otherwise} \end{cases}$$

where $k \in [1, N_r], k \neq i$ and N_r is the number of selected rooster. $X_{i,j}$ represents the position of rooster number i in jth dimension during t and $t+1$ iteration, $randn(0, \sigma^2)$ used to generate Gaussian random number with mean 0 and variance σ^2, ε is a constant with low value, and f_i is the fitness value for the corresponding rooster i.

2) **The formula that used for the hens' position update is given by** Eq.4, Eq. 5 and Eq. 6:

$$X_{i,j}^{t+1} = X_{i,j}^{t} + S_1 randn(X_{r1,j}^{t} - X_{i,j}^{t}) + S_2 randn(X_{r2,j}^{t} - X_{i,j}^{t}) \quad (4)$$

where,

$$S_1 = \exp(\frac{f_i - f_{r1}}{|f_i| + \varepsilon}) \quad (5)$$

and

$$S_1 = \exp(f_{r2} - f_i) \quad (6)$$

where, $r_1, r_2 \in [1, \ldots, N], r_1 \neq r_2$, r_1 is the index of a rooster, while r_2 is a chicken from the swarm that can be a rooster or a hen and a uniform random number is generated by $randn$.

3) **The formula that used for the chicks' position update is given by** Eq.7:

$$X_{i,j}^{t+1} = X_{i,j}^{t} + FL(X_{m,j}^{t} - X_{i,j}^{t}), \quad FL \in [0, 2] \quad (7)$$

where, $X_{m,j}^{t}$ is the position of the i^{th} chick's mother.

V. PROPOSED ALGORITHM

In this section, we provide CSOCA for selecting a group of nodes to work as CHs such that minimizing the consumed energy for data transmission. I.e., the optimization objective is to determine the best group of nodes to work as heads, that prolongs the network lifetime and minimize the consumed energy. The objective function of our algorithm is to maximize the lifetime of the network (δ_n^n) that ends as soon as the first node dies and it is calculated as follows [42]:

$$\delta_n^1 = \min_{s \in S} \delta_s \quad (8)$$

where, δ_s is the lifetime of node s and S is the set of nodes in the network.

Let n sensor nodes are distributed uniformly, and there are k clusters. Therefore, there are n/k nodes per cluster (one CH and (n/k)-1 cluster members (CMs)). The total consumed energy by the CH (e_{CH}) for a single round is given by [26]:

$$e_{CH} = (\frac{n}{k} - 1).E_{Rx}(b) + \frac{n}{k}.b.E_{DA} + E_{Tx}(b, d_{toBS}). \quad (9)$$

The CM node sends its data to its CH, as a result, the total consumed energy by the CM node over round can be as follows:

$$e_{CM} = E_{Tx}(b, d_{toCH}). \quad (10)$$

where, d_{toBS} and d_{toCH} are the average distance between the head node and the BS, and the average distance between member nodes and the CH, respectively.

Algorithm 1 CSOCA:Cluster Heads Election Phase Algorithm

Input: K number of CHs, CSO parameters
Ouput: The indices list that includes the indices of the nodes that work as CHs.

Initialize the matrix X that represents population by random values from 0 to 1.
Repair the infeasible solutions that do not has K number of CH.
For each row of X, calculates the fitness value.
bestX = the row in X which has the corresponding to best fitness value
for $t = 1$**to** t_{max} **do**
  if $(t\%G == 0||t == 1)$ **then**
    All the fitness values taken in the ascending order.
    Divide X into three categories (rooster, hens, and chicks).
  end if
  for each row y in X **do**
    if y represents a rooster **then**
      Using Eq.3, update y values.
    end if
    if y represents a hen **then**
      Using Eq.4, update y values.
    end if
    if y represents a chick **then**
      Using Eq.7, update y values
    end if
    Transform y to its binary representation b using equation 13
    Repair the infeasible solutions that do not has K number of CH.
    Update the fitness values for each row of X.
    if a new solution is better than the previous one, update bestX with the new best solution .
  end for
end for

The total energy consumption in a cluster during a round is

$$E^{cluster}_{consumed} = e_{CH} + e_{CM} \tag{11}$$

The objective is to maximize δ^1_n by minimizing total consumed energy in the network per round such that CH rotation achieved for balancing consumed energy. As a result, the fitness function is given by:

$$F = \frac{\sum_{i=1}^{k} E^{cluster}_{consumed}(i)}{a + \sum_{i=1}^{k} E^{cluster}_{consumed}(i)} + \left(\frac{\beta}{a+\beta}\right) \tag{12}$$

where, the number of CHs represented by k, β is the total number of time the selected nodes work as CHs and a is constant greater than zero.

A. CHICKEN SWARM OPTIMIZATION BASED CLUSTERING ALGORITHM

In this section, we discuss our proposed algorithm (CSOCA). The main target of CSOCA is to extend the network lifetime and minimizes the energy consumption in the network. Each chicken individual in CSO is presented by a real-valued vector (i.e., CSO work in continuous space).

However, in our algorithm, we consider the binary representation of an individual, where the index represents the node id and the 0 value represents that node is not CH node, while 1 represents that node is a CH node. We use the sigmoid function to transform the real value of each chicken to binary value. The transformation the formula is as follows:

$$b_x = \begin{cases} 1, & \text{if } \mathrm{sigmf}(x) > 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where $\mathrm{sigmf}(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function.

The CSOCA includes three phases that repeated each round: Cluster Heads Election, Cluster Formation, and Data Collection phase. In Cluster Heads Election phase, CSO is used for election decision problem by selecting the best nodes that work as CHs considering minimizing the consumed energy per round(i.e., minimize the value in equation Eq.12).

In the Cluster Formation phase, each non-head node selects the CH which minimizes communication cost. In the Data Collection phase, the CH collects cluster data and transmits it to BS.

1) CLUSTER HEADS ELECTION PHASE

This phase consists of three steps: Initialization step, Selection step and Output step (the proposed algorithm of this phase can be describe in Algorithm 1). These steps executed by BS and can be described as following:

Initialization: In this step our algorithm initialize all parameters, the number of CH (K) and other related parameters of CSO. Set the initial CSO parameters such as the population size (*popSize*), the number of roosters (R_n), the number of hens (H_n), the number of chicks (C_n),the swarm updating frequency (G) and the maximum number of iteration (t_{max}).

Selection: In this step, the CSO algorithm is utilized to find the best group of nodes that work as CHs and minimizing consumed energy Eq.12. The procedures described as follows:

Step 1: Initialize the population matrix X.
Step 2: For each row in X calculates the fitness values.
Step 3: Arrange the individuals fitness values descending.
Step 4: Divide X into three groups (rooster, hens, and chicks) according to their fitness value.
Step 5: Updates the fitness value of rooster,hens, chicks using Eq.3,4,7, respectively.
Step 6:Convert population to its binary form using equation 13
- repair the infeasible solutions that do not have K CHs.
- for each infeasible individual if the total number of ones greater or less than K; then

randomly set or unset items to one in order to keep only K items with ones.

Step 7: Compute the fitness value of each row in X, and then update the best solution $bestX$.

Step 8: Repeat from step 2 to step 7 till reaching the maximum number of iterations t_{max}.

Output: Finally, the algorithm transforms the output ($bestX$) to the binary form and returns the indices list that includes the indices of the nodes that work as CHs (where, one value indicates that the index at this position is a CH).

2) CLUSTER FORMATION PHASE

After the BS determines the group of nodes (C) that work as CHs, it sends a message to inform each node in the group C to work as CH. Each node that receives an inform message, it announces its role through an advertising message containing its ID. Each node that does not belong to the group C and receives the advertising message chooses the CH which has minimum communication costs and then sends a join message to it. Finally, each CH constructs a transmission schedule for its member nodes then send it to cluster members(CMs).

3) DATA COLLECTION PHASE

In this phase, each CM node sends its collected data to its respective CH in the time slot assigned to it. The data message also contains the node's ID and its remaining energy. Where this local information of CM nodes can be utilized for deciding which group of nodes will work as CHs in the next round. Each CH begins after the end of the schedule evaluate the received data to remove duplicate data using the data aggregation process(a data fusion algorithm is used to merge the received data). Finally, the CH sends to the BS the reformed information along with local information of its CMs.

B. CSOCA WITH GENETIC ALGORITHM (CSOCA-GA)

Genetic algorithm (GA) belongs to evolutionary algorithms and it is inspired by the process of natural selection. GA utilizes crossover and mutation processes along with the current generation to generate the next generation. Crossover and mutation are essential processes in GA. The crossover process help in extracting the best individual from different individuals. On the other hand, the mutation process adds diversity to the population and thus increases the probability of the GA to generate individuals with better fitness values [43].

Chicken Swarm Optimization Based Clustering Algorithm is enhanced with Genetic Algorithm main processes (crossover and mutation). The main goals of CSOCA-GA are the same goals of CSOCA is to save energy and extend the lifetime of the network. The CSOCA-GA includes three phases that repeated each round (A flow diagram of CSOCA and CSOCA-GA is shown Figure 2). The second and the third phases (Cluster Formation, and Data Collection phase) are the same as in CSOCA, while the first phase (Cluster

Algorithm 2 CSOCA-GA:Cluster Heads Election Phase Algorithm

Input: K number of CHs, number of roosters N_r,*popSize*), one point crossover, mutation rate.

Ouput: the indices list that includes the indices of the nodes that work as CHs.

Initialize the matrix X that represents population by random values from 0 to 1.
Repair the infeasible solutions that do not has K number of CHs.
For each row of X, calculates the fitness values.
bestX = the row in X which has the corresponding to best fitness value
for $t = 1$**to** t_{max} **do**
  if $(t\%G == 0||t == 1)$ **then**
    All the fitness values taken in the ascending order.
    Divide X into number of rows referring to three categories rooster, hens, and chicks.
  end if
  for each row y in X **do**
    if y represents a rooster **then**
      Using Eq.3, update y values.
    end if
    if y represents a hen **then**
      Using Eq.4, update y values.
    end if
    if y represents a chick **then**
      Using Eq.7, update y values
    end if
    Transform x to its binary representation b using equation 13
    for j=$N_r + 1$ **to** *popsize* **step** 2 **do**
      Generate x_{c1},x_{c2} from X_j,X_{j+1} by one point cross over
      set $X_j = x_{c1}$,$X_{j+1} = x_{c2}$
    end for
    for $j = N_r + 1$ **to** *popsize* **do**
      Select a random number r from 0 to 1.
      if $r <$ mutation rate **then**
        Generate random integer $r1$, $(1 \leq r1 \leq n)$
        if $X(r1) == 1$ **then**
          set $X(r1) = 0$
        else
          set $X(r1) = 1$
        end if
      end if
    end for
    Repair the infeasible solutions that do not has K number of CHs.
    Update the fitness values for each row of X.
    if a new solution is better than the previous one, update bestX with the new best solution .
  end for
end for

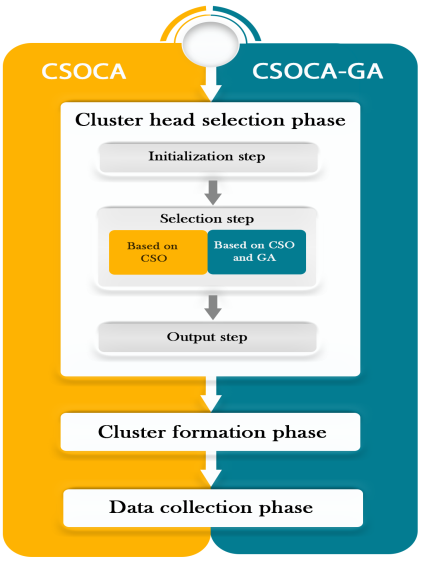

FIGURE 2. Diagram of COSCA and CSOCA-GA.

Heads Election) modified by adapting GA crossover and mutation processes into CSOCA to increase the population diversity(the proposed algorithm of this phase can be describe in Algorithm 2).

1) CSOCA-GA: CLUSTER HEADS ELECTION PHASE

This phase consists of three steps: The initialization step, the Selection step, and the Output step. In the initialization step, CSO parameters, number of CHs, mutation rate and crossover method are set. In the output step, the individual with best fitness value is used to get elected CHs. Finally, in the selection step, the individuals that represent roosters (individuals have the best fitness values) moved to the subsequent generation. While, individuals that represent hens, and chicks are subjected to crossover and mutation processes as follows:

2) CROSSOVER

The crossover process is used to generate the next population by taking pairs of parents in the current population then combining these pairs to create a new individual for the new population. We use a One-point crossover in which randomly selects a crossover point, and the values after that point are swapped between the two parents.

3) MUTATION

Interchanges in the values of a randomly selected individual are the basic operations for mutation operator [43]. The mutation probability is the most important factor in the mutation process since it defines the frequency of mutation of each part in the individual. The new individuals are generated directly after the crossover process if there is no mutation. The mutation operation in CSOCA-GA can be described as follows:

- for each individual x_i
- Select a random number r from 0 to 1.
- **If** $r \leq$ mutation rate, **then**
  - Select random integer r_1 $(1 \leq r_1 \leq n)$
  - transform $x_i(r_1)$ to binary form (b) using equation13
  - If $(b = 1)$, set $x_i(r_1) = 0$; Otherwise, set $x_i(r_1) = 1$.
- **Else**, stop;

VI. SIMULATION RESULTS

In this section, the CSO algorithm's parameters are analyzed. Then CSOCA and CSOCA-GA are evaluated in terms of network lifetime and consumed energy against other algorithms. All experiments are conducted using MATLAB R2016b. Besides, we consider 100 nodes are randomly scattered in the region of size 100 $\times$ 100 meters square with BS at the corner, and with simulation setting provided in Table 2.

TABLE 2. Simulation setting.

Parameter	Value
Network area size	100 $\times$ 100
Nodes	100
Initial energy	$0.5J$
E_{elec}	50 nJ/bit
Free space ϵ_{fs}	10 $pJ/bit/m^2$
Multi-path ϵ_{mp}	0.00013 $pJ/bit/m^4$
d_0	87 m
E_{DA}	5 nJ/bit/signal
Packet size	4000 bits
Percentage of CHs	0.05

A. CSO PARAMETER ANALYSIS

In this section, the effects of the CSO algorithm's parameters on the performance of the proposed algorithms are analyzed, these parameters include population size (*popSize*), number of roasters (N_r), number of hens (N_h) and the update time steps (G).

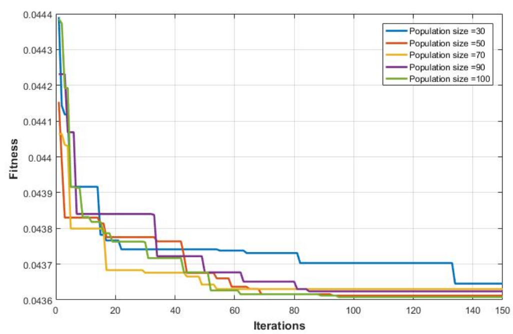

FIGURE 3. The proposed algorithm performance as a function of *popSize*.

To study the effect of *popSize* on the proposed algorithm we set $N_r = 0.3$, $N_h = 0.5$, $G = 10$ and number of chicks $N_c = N - N_r - N_h$ and the results can be shown in Figure 3. In Figure 3, it is obvious that the proposed algorithm obtains

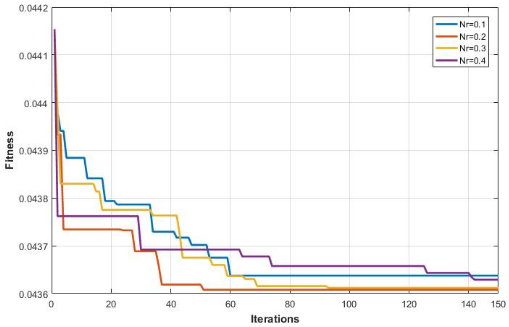

FIGURE 4. Optimization performance of CSOCA versus N_r.

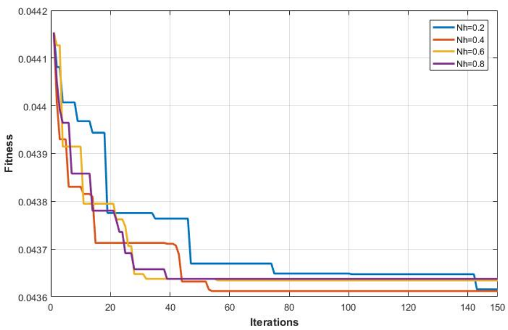

FIGURE 5. Optimization performance of CSOCA versus N_h.

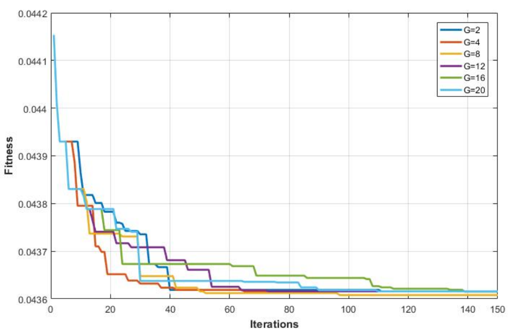

FIGURE 6. Optimization performance of CSOCA versus G.

the best fitness with a *popSize* value from 50 to 100. So we can say that when *popSize* value equal to $n/2$ the proposed algorithm reaches to the best performance.

To study the impact of N_r and N_h on the proposed algorithm performance, we set the following: firstly, we fix $popSize = 50$, $N_h = 0.5$, $G = 10$ and $N_c = N - N_r - N_h$ and then set the value of N_r as $N_r = \{0.1, 0.2, 0.3, 0.4\}$ to evaluate the optimization performance of the proposed algorithm as shown in Figure 4. From Figure 4 we can obtain that the when the $N_r = 0.2$ the proposed algorithm has the best optimization performance.

In order to study the effect of N_h, we fix $popSize = 50$, $N_r = 0.2$, $G = 10$ and $N_c = N - N_r - N_h$ and then set the value of N_h as $N_h = \{0.2, 0.4, 0.6, 0.8\}$ shown in Figure 5.

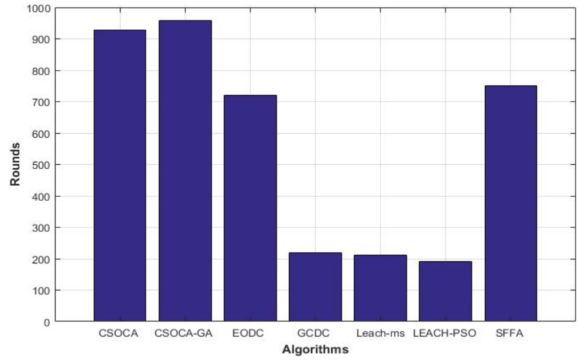

FIGURE 7. Lifetime (first node dies) in CSOCA, CSOCA-GA, EODC, GCDC, LEACHMS, LEACHPSO and SFFA algorithms.

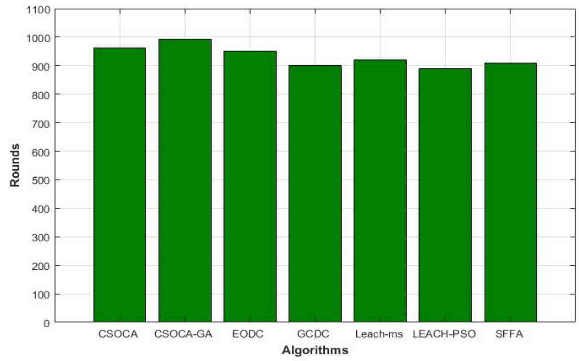

FIGURE 8. Lifetime (half node dies) in CSOCA, CSOCA-GA, EODC, GCDC, LEACHMS, LEACHPSO and SFFA algorithms.

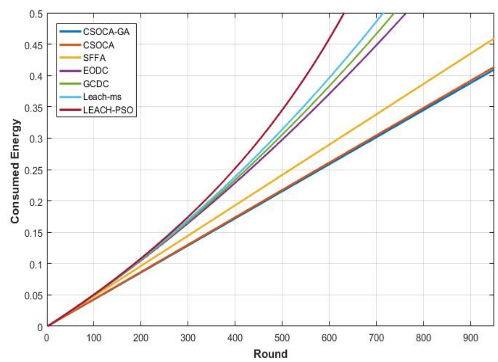

FIGURE 9. Average consumed energy per round for CSOCA, CSOCA-GA, EODC, GCDC, LEACHMS, LEACHPSO and SFFA algorithms.

From Figure 5it is clear that the proposed algorithm has the best optimization performance when the $N_h = 0.4$

Finally, in order to test the impact of G values on the proposed algorithm we fix $popSize = 50$, $N_h = 0.4$, $N_r = 0.2$ and $N_c = N - N_r - N_h$ and then set the value of G as $G = \{2, 8, 12, 16, 20\}$. As obtained from Figure 6, the proposed algorithm has the best optimization performance when $G = 8$.

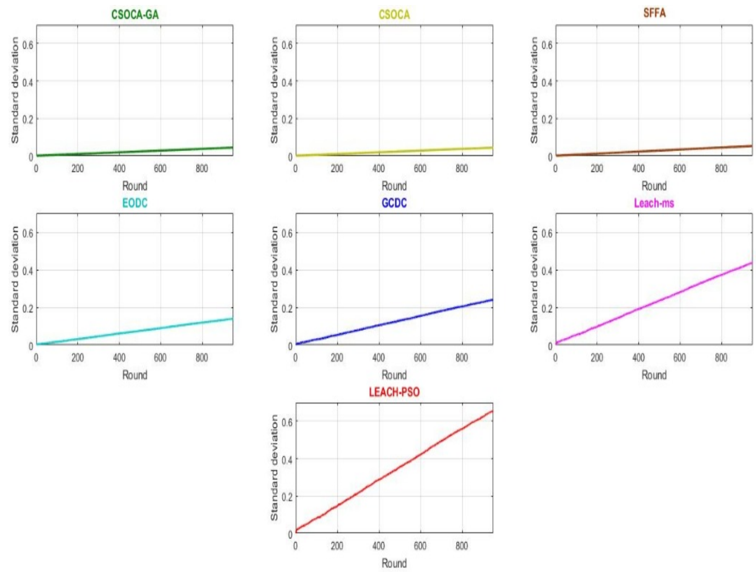

FIGURE 10. Standard deviation for CSOCA, CSOCA-GA, EODC, GCDC, LEACHMS, LEACHPSO and SFFA algorithms.

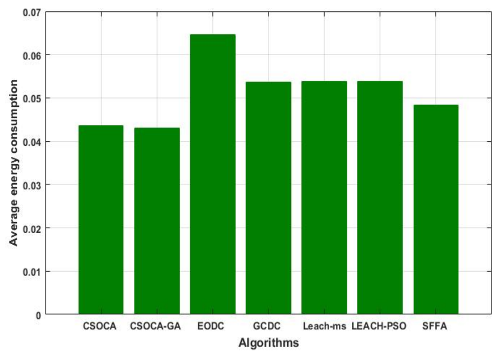

FIGURE 11. Average consumed energy until first node die for CSOCA, CSOCA-GA, EODC, GCDC, LEACHMS, LEACHPSO and SFFA algorithms.

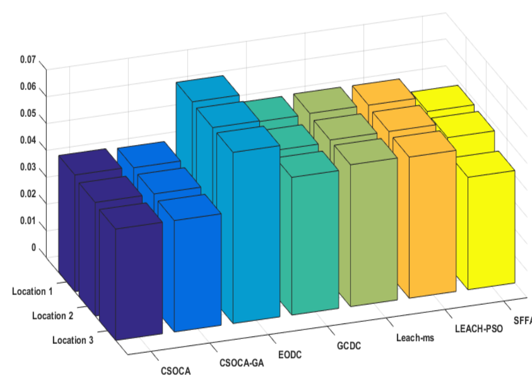

FIGURE 12. The impact of BS placement on average consumed energy until first node die.

B. COSCA AND COSCA-GA EVALUATIONS

For the next tests, we consider one point crossover and mutation rate of 0.05% for CSOCA-GA. The performance results of COSCA, and CSOCA-GA are compared with LEACH-MS [25], LEACH-PSO [24], EODC [32], GCDC [33] and SFFA [31] algorithms.

1) PERFORMANCE METRICS USED

The performance metrics we follow here are described as follows:

1) Network lifetime: the time at which the first node dies.
2) Average of consumed per node at each round: the overall consumed energy per round divided by the total number of nodes.
3) Average consumed energy per round: the average of consumed energy until the first node dies.

2) ANALYSIS OF NETWORK LIFETIME

Figures 7 and 8 show that CSOCA improves the network lifetime in terms of the first node dies compared with EODC, GCDC, LEACHMS, LEACHPSO, and SFFA up to 22%, 76%, 77%, 80%, and 19%, respectively and improves the network lifetime in terms of half nodes die compared with EODC, GCDC, LEACHMS, LEACHPSO, and SFFA up to 32%, 6%, 4%, 7%, and 5%, respectively. While, CSOCA-GA improves the network lifetime in terms of the first node dies compared with EODC, GCDC, LEACHMS, LEACHPSO, and SFFA up to 25%, 77%, 78%, 80 and 22%, respectively and improves the network lifetime in terms of half nodes die compared with EODC, GCDC, LEACHMS, LEACHPSO, and SFFA up to 35%, 9%, 7%, 10%, and 8%, respectively.

It is obvious that the number of live nodes in CSOCA and CSOCA-GA is more than those of EODC, GCDC, LEACHMS, LEACHPSO, and SFFA algorithms. We can

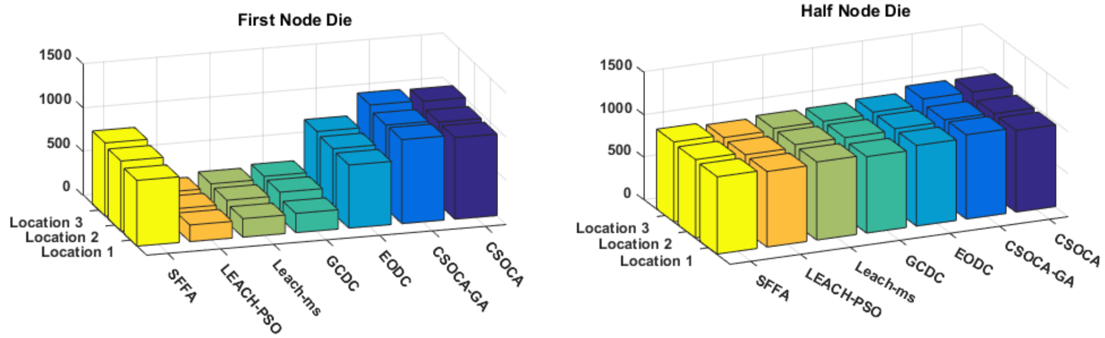

FIGURE 13. The impact of BS placement on the network lifetime.

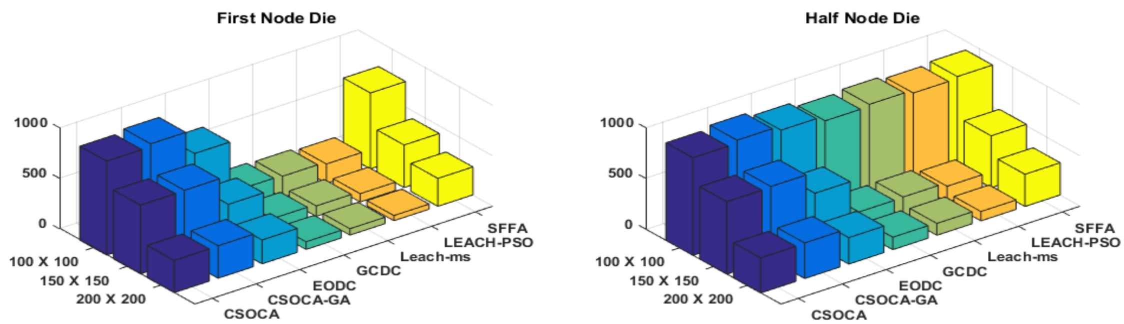

FIGURE 14. The impact of network size on the network lifetime.

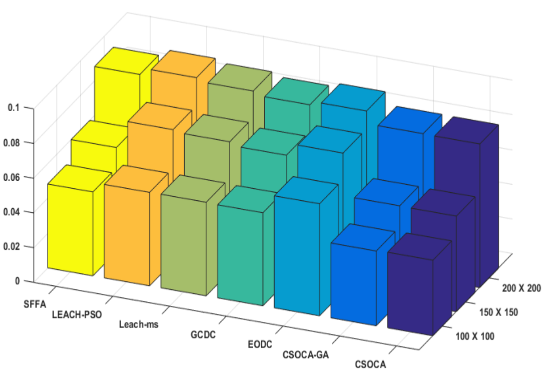

FIGURE 15. The impact of network size on average consumed energy until first node die.

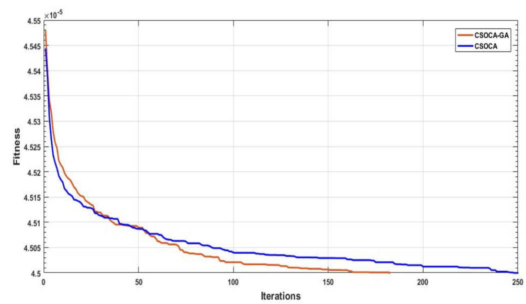

FIGURE 16. Convergence of CSOCA and CSOCA-GA.

observe that CSOCA-GA improves the network lifetime compared with CSOCA up to 3%.

3) ANALYSIS OF ENERGY CONSUMPTION

Here, we calculate the standard deviation of the consumed energy of nodes per round and the average consumed energy of nodes per round. Figure 9 shows the average of consumed energy per round for CSOCA, CSOCA-GA, EODC, GCDC, LEACHMS, LEACHPSO, and SFFA algorithms.

The standard deviation of consumed energy of nodes and the average consumed energy of nodes per round as the number of rounds increases are represented in Figures 10 and 9, respectively. Where the distribution rate of consumed energy of nodes can be represented by the standard deviation of the consumed energy. The consumed energy of each node is uniform if the standard deviation is small. Figure 10 shows the standard deviations for the consumed energy among the nodes of each algorithm. The curves indicate that there is even distribution of consumed energy among the nodes in CSOCA and CSOCA-GA, therefore, the balance of the energy consumption in the network for CSOCA and CSOCA-GA is better than the other algorithms.

Figure 11 shows the average energy consumption per round until the first node die and it can be estimated by total consumed energy until the first die divided by round number at which the first node die. From Figure 11 we can observe that CSOCA and CSOCA-GA evenly distribute the workload

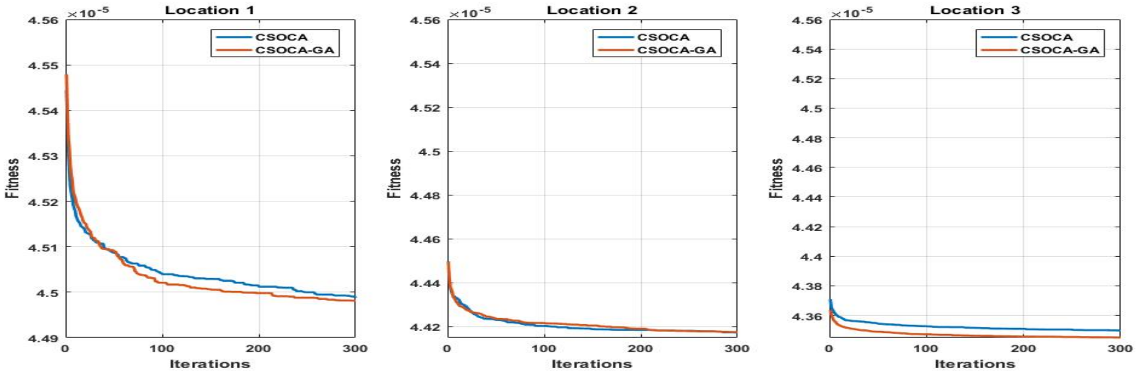

FIGURE 17. The impact of BS placement on the convergence of CSOCA and CSOCA-GA.

among the nodes and divide the workload by round which increases the lifetime of the network which is not achieved by the other algorithms.

C. IMPACT OF BASE STATION PLACEMENT

In this test, our aim is to show the effect of BS placements on the performance of the algorithms. In this test, 100 sensor nodes are deployed randomly in 100 $m\times$ 100 m area. The BS is located at three different placements: the corner, the middle of one edge and the center of the network, for simplicity we refer to these placements as $Location_1$, $Location_2$, and $Location_3$ respectively. The performance is evaluated using the same setting in the previous tests for CSO and COSCA-GA. Figure 13 shows the impact of BS placement on the lifetime (in terms of the first node dies and the half node dies) for CSOCA, CSOCA-GA, EODC, GCDC, LEACHMS, LEACHPSO, and SFFA algorithms. Figure 12 shows the impact of BS placement on the average consumed energy until the first node die for CSOCA, CSOCA-GA, EODC, GCDC, LEACHMS, LEACHPSO, and SFFA algorithms. In Figure 13 and Figure 12 the network lifetime is improved at $Location_3$ as compared to $Location_2$ and $Location_1$, besides, the consumed energy until first node die is minimized for all algorithms this due to the paths from nodes to the BS is decreased.

D. IMPACT OF NETWORK SIZE

In this test, the effect of the network size on the performance of the algorithms is discussed. Where, 100 nodes are randomly deployed in our simulations in the following regions 100 × 100; 150 × 150; 200 × 200; on a two-dimensional plane with BS at the corner and with the same setting in the previous tests for CSO and COSCA-GA.

Figure 14 and Figure 15 show the effect of network size on the network lifetime (in terms of the first node dies and the half node dies) and the average consumed energy until the first node die, respectively, for the algorithms CSOCA, CSOCA-GA, EODC, GCDC, LEACHMS, LEACHPSO, and SFFA.

From Figure 14 and Figure 15 we can notice that for all algorithms as the network size expands the consumed energy is increased and the network lifetime is decreased, this is due to the increased distance between nodes. The similar behavior can also be noticed when the BS is placed at the network boundary, as shown in Figures 13 and 12.It is clear that CSOCA and CSOCA-GA still provide the best network lifetime performance.

E. ANALYSIS OF CSOCA AND CSOCA-GA CONVERGENCE

In this test, our target is to show the convergence rate of the fitness value for CSOCA and CSOCA-GA. All the graph obtained by plotting the average of 50 different runs with different random network topologies and with the same CSO parameters setting ($popSize$ = 100, N_r = 0.2, N_h = 0.4,and G = 8) for CSOCA and CSOCA-GA, and with one point crossover and mutation rate of 0.05% for CSOCA-GA.

Figure 16 shows the relationship between the fitness value and the number of iterations for CSOCA and CSOCA-GA. We can observe that the value of the average fitness values decreases as the number of iterations increases for CSOCA and CSOCA-GA, however, the convergence of CSOCA-GA is faster than CSOCA to the solution(where CSOCAGA convergence at iteration number 160 while CSOCA convergence at iteration number 260), this is due to CSOCA-GA uses crossover and mutation processes in GA to produce offsprings, which increases the population diversity and gives the ability to escape from a local optimum.

Figure 17 shows the effect of BS placement on the convergence of CSOCA and CSOCA-GA. It is clear that the convergence of CSOCA-GA is better than CSOCA at the corner ($Location_1$), middle of one edge ($Location_2$), and center of the network ($Location_3$). It is noticed that the fitness value improved for CSOCA and CSOCA-GA as the BS moved to the network center this is due to the distance to the BS decreases. Moreover, the fitness value for CSOCA-GA is improved at the network center as compared to the other BS locations.

Figure 18 shows the effect of node density on the convergence of CSOCA and CSOCA-GA when the number of nodes

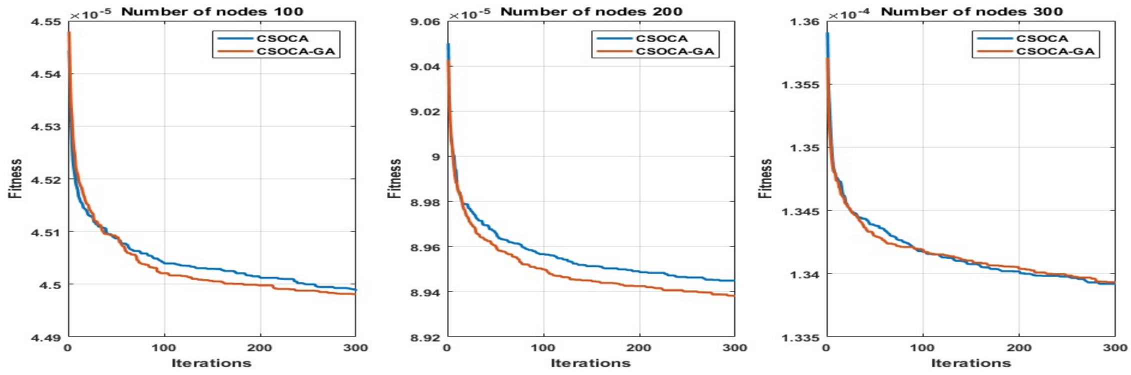

FIGURE 18. The impact of node density on the convergence of CSOCA and CSOCA-GA.

increased from 100 to 300 an increment of 100 nodes. It is noticed that as the number of nodes increased the fitness value increased. Furthermore, as the number of nodes increased the convergence of CSOCA and CSOCA-GA is close to each other.

In summary, even energy consumption between all nodes is achieved by CSOCA and CSOCA-GA, due to a dynamic clustering approach is utilized where various sets of nodes work as CH at each round and thus, extending the network lifetime. Both CSOCA and CSOCA-GA achieve well distribution of CHs; this due to they utilize the CSO algorithm for CHs selection processes, where the CSO algorithm adaptively searches for the solution and has the ability to distribute the search process by adopting the hierarchy order of the population.

VII. CONCLUSION

In order to improve network lifetime, we proposed the Chicken Swarm Optimization based Clustering Algorithm (CSOCA)and CSOCA-GA (which is an improvement of CSOCA by employing GA) where the CSO algorithm is modified to optimize the energy usage in WSNs. CSOCA and CSOCA-GA utilize a hierarchal order concept in which the population divided into three groups, then arranged according to their fitness values in order to select the best nodes that work as CHs in each round. CSOCA-GA employs crossover and mutation processes to increase the population diversity. The fitness function is formed to minimize the total consumed energy and the total number of times of the selected group of nodes to work as CHs. Results show that CSOCA and CSOCA-GA improve network lifetime compared to EODC, GCDC, LEACHMS, LEACHPSO and SFFA algorithms.

REFERENCES

[1] A. Bradai, A. Benslimane, and K. D. Singh, "Dynamic anchor points selection for mobility management in software defined networks," *J. Netw. Comput. Appl.*, vol. 57, pp. 1–11, Nov. 2015.

[2] S. Amri, F. Khelifi, A. Bradai, A. Rachedi, M. L. Kaddachi, and M. Atri, "A new fuzzy logic based node localization mechanism for wireless sensor networks," *Future Gener. Comput. Syst.*, vol. 93, pp. 799–813, Apr. 2019.

[3] Z. Ning, X. Kong, F. Xia, W. Hou, and X. Wang, "Green and sustainable cloud of things: Enabling collaborative edge computing," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 72–78, Jan. 2019.

[4] Z. Ning, X. Hu, Z. Chen, M. Zhou, B. Hu, J. Cheng, and M. S. Obaidat, "A cooperative quality-aware service access system for social Internet of vehicles," *IEEE Internet Things J. 5, no.*, vol. 4, no. 2017, pp. 2506–2517.

[5] Z. Ning, F. Xia, X. Hu, Z. Chen, and M. S. Obaidat, "Social-oriented adaptive transmission in opportunistic Internet of smartphones," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 810–820, Apr. 2017.

[6] Z. Ning, L. Liu, F. Xia, B. Jedari, I. Lee, and W. Zhang, "CAIS: A copy adjustable incentive scheme in community-based socially aware networking," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3406–3419, Apr. 2017.

[7] F. Xia, L. Liu, J. Li, J. Ma, and A. V. Vasilakos, "Socially aware networking: A survey," *IEEE Syst. J.*, vol. 9, no. 3, pp. 904–921, Sep. 2015.

[8] F. Wang and J. Liu, "Networked wireless sensor data collection: Issues, challenges, and approaches," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 673–687, 2011.

[9] H. Karl and A. Willig, *Protocols and Architectures for Wireless SensorNetworks*. Hoboken, NJ, USA: Wiley, 2007.

[10] D. Gong, Y. Yang, and Z. Pan, "Energy-efficient clustering in lossy wireless sensor networks," *J. Parallel Distrib. Comput.*, vol. 73, no. 9, pp. 1323–1336, Sep. 2013.

[11] P. Kuila and P. K. Jana, "A novel differential evolution based clustering algorithm for wireless sensor networks," *Appl. Soft Comput.*, vol. 25, pp. 414–425, Dec. 2014.

[12] X. Meng, X. Gao, and H. Zhang, "A new bio-inspired algorithm: Chicken swarm optimization," in *Advances in Swarm Intelligence. ICSI* (Lecture Notes in Computer Science), vol. 8794, Y. Tan, Y. Shi, and C. A. C. Coello, Eds. Cham, Switzerland: Springer, 2014.

[13] S. Deb, X.-Z. Gao, K. Tammi, K. Kalita, and P. Mahanta, "Recent studies on chicken swarm optimization algorithm: A review (2014–2018)," *Artif. Intell. Rev.*, vol. 53, no. 3, pp. 1737–1765, Mar. 2020.

[14] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, 2000, p. 10.

[15] Y.-L. Chen, N.-C. Wang, and Y.-C. Lin, "Power-efficient gathering in sensor information system architectures with a phase-based coverage algorithm in a wireless sensor network," *Adv. Sci. Lett.*, vol. 9, no. 1, pp. 620–625, Apr. 2012.

[16] J. Wang, Y. Gao, X. Yin, F. Li, and H.-J. Kim, "An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 2018, Dec. 2018, Art. no. 9472075.

[17] J. Wang, Y. Gao, W. Liu, W. Wu, and S.-J. Lim, "An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor networks," *Comput., Mater. Continua*, vol. 58, no. 3, pp. 711–725, 2019.

[18] W. Osamy, A. M. Khedr, A. Aziz, and A. A. El-Sawy, "Cluster-tree routing based entropy scheme for data gathering in wireless sensor networks," *IEEE Access*, vol. 6, pp. 77372–77387, 2018.

[19] J. Wang, Y. Gao, K. Wang, A. K. Sangaiah, and S.-J. Lim, "An affinity propagation-based self-adaptive clustering method for wireless sensor networks," *Sensors*, vol. 19, no. 11, p. 2579, 2019.

[20] J. Wang, Y. Gao, C. Zhou, R. S. Sherratt, and L. Wang, ''Optimal coverage multi-path scheduling scheme with multiple mobile sinks for WSNs,'' *CMC-Comput., Mater. Continua*, vol. 62, no. 2, 695–711, 2020.
[21] J.-L. Liu and C. V. Ravishankar, ''LEACH-GA: Genetic algorithm-based energy-efficient adaptive clustering protocol for wireless sensor networks,'' *Int. J. Mach. Learn. Comput.*, vol. 1, no. 1, p. 79, 2011.
[22] L. Zhao, S. Qu, and Y. Yi, ''A modified cluster-head selection algorithm in wireless sensor networks based on LEACH,'' *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, p. 287, Dec. 2018.
[23] A. Salim, W. Osamy, and A. M. Khedr, ''IBLEACH: Intra-balanced LEACH protocol for wireless sensor networks,'' *Wireless Netw.*, vol. 20, pp. 1515–1525, Jun. 2014, doi: 10.1007/s11276-014-0691-4.
[24] A. Yadav, S. Kumar, and S. Vijendra, ''Network life time analysis of WSNs using particle swarm optimization,'' *Procedia Comput. Sci.*, vol. 132, pp. 805–815, 2018.
[25] T. Shankar and A. P. A. Karthikeyan Sivasankar Rajesh, ''Hybrid approach for optimal cluster head selection in WSN using leach and monkey search algorithms,'' *J. Eng. Sci. Technol.*, vol. 2, no. 12, pp. 506–517, 2017.
[26] W. Osamy, A. Salim, and A. M. Khedr, ''An information entropy based-clustering algorithm for heterogeneous wireless sensor networks,'' *Wireless Netw*, vol. 26, pp. 1869–1886, Nov. 2018, doi: 10.1007/s11276-018-1877-y.
[27] M. Azharuddin and P. K. Jana, ''Particle swarm optimization for maximizing lifetime of wireless sensor networks,'' *Comput. Electr. Eng.*, vol. 51, pp. 26–42, Apr. 2016.
[28] V. Anand and S. Pandey, ''Particle swarm optimization and harmony search based clustering and routing in wireless sensor networks,'' *Int. J. Comput. Intell. Syst.*, vol. 10, no. 1, pp. 1252–1262, 2017.
[29] R. Ranjan and S. Kar, ''A novel approach for finding optimal number of cluster head in wireless sensor network,'' in *Proc. Nat. Conf. Commun. (NCC)*, Jan. 2011, pp. 1–5.
[30] I. S. Akila and R. Venkatesan, ''A fuzzy based energy-aware clustering architecture for cooperative communication in WSN,'' *Comput. J.*, vol. 59, no. 10, pp. 1551–1562, Oct. 2016.
[31] A. Barzin, A. Sadeghieh, H. K. Zareh, and M. Honarvar, ''Hybrid swarm intelligence-based clustering algorithm for energy management in wireless sensor networks,'' *J. Ind. Syst. Eng.*, vol. 12, no. 3, pp. 78–106, 2019.
[32] C. Jothikumar and R. Venkataraman, ''EODC: An energy optimized dynamic clustering protocol for wireless sensor network using PSO approach,'' *Int. J. Comput. Commun. Control*, vol. 14, no. 2, pp. 183–198, 2019.
[33] M. Kheireddine, R. Abdellatif, and G. Ferrari, ''Genetic centralized dynamic clustering in wireless sensor networks,'' in *Proc. IFIP Int. Conf. Comput. Sci. Appl.* Cham, Switzerland: Springer, 2015, pp. 503–511.
[34] G. Sun, Y. H. Liu, S. Liang, Z. Y. Chen, A. M. Wang, Q. A. Ju, and Y. Zhang, ''A sidelobe and energy optimization array node selection algorithm for collaborative beamforming in wireless sensor networks,'' *IEEE Access*, vol. 6, pp. 2515–2530, 2018.
[35] D. Wu, S. Xu, F. Kong, ''Convergence analysis and improvement of the chicken swarm optimization algorithm,'' *IEEE Access*, vol. 4 , pp. 9400–9412, 2016.
[36] Q. Wang and L. Zhu, ''Optimization of wireless sensor networks based on chicken swarm optimization algorithm,'' *AIP Conf.*, vol. 1839, no. 1, 2017, Art. no. 020197.
[37] P. Chen and Y. Mao, ''Wireless sensor network node localization algorithm based on chicken swarm optimization and multi-power mobile anchor,'' in *Proc. 3rd Int. Conf. Mater. Eng., Manuf. Technol. Control*. Paris, France: Atlantis Press, 2016, pp. 1–6.
[38] M. Al Shayokh and S. Y. Shin, ''Bio inspired distributed WSN localization based on chicken swarm optimization,'' *Wireless Pers. Commun.*, vol. 97, no. 4, pp. 5691–5706, Dec. 2017.
[39] A. Aziz, K. Singh, W. Osamy, and A. M. Khedr, ''Optimising compressive sensing matrix using chicken swarm optimisation algorithm,'' *IET Wireless Sensor Syst.*, vol. 9, no. 5, pp. 306–312, Oct. 2019.
[40] A. Aziz, K. Singh, W. Osamy, and A. M. Khedr, ''Effective algorithm for optimizing compressive sensing in IoT and periodic monitoring applications,'' *J. Netw. Comput. Appl.*, vol. 126, pp. 12–28, Jan. 2019.
[41] C. Lv, Q. Wang, W. Yan, and Y. Shen, ''Energy-balanced compressive data gathering in wireless sensor networks,'' *J. Netw. Comput. Appl.*, vol. 61, pp. 102–114, Feb. 2016.
[42] I. Dietrich and F. Dressler, ''On the lifetime of wireless sensor networks,'' *ACM Trans. Sensor Netw.*, vol. 5, no. 1, 2009, Art. no. 5.
[43] S. N. Sivanandam and S. N. Deepa, ''Genetic algorithms,'' in *Introduction to Genetic Algorithms*. Berlin, Germany: Springer, 2008, pp. 15–37.

WALID OSAMY received the B.Sc. (Hons.), M.S., and Ph.D. degrees in computer science from the Faculty of Science, Zagazig University, Egypt, in June 2000, in June 2006, and in September 2010, respectively. He has been involved in the projects of network infrastructure, and management information systems with the Communication Information Technology Center (CITC), Zagazig University, Egypt. From 2010 to 2019, he was an Assistant Professor with the Department of Computer Science, Faculty of Computers and Informatics, Benha University, Egypt, where he has been an Associate Professor with the Department of Computer Science, Faculty of Computers and Artificial Intelligence, since 2019. Since 2015, he has been an Assistant Professor with Qassim University, Buridah, Saudi Arabia. His research interests include computational intelligence and in the field of the IoT (mobile computing and wireless sensor networks (WSNs)).

AHMED A. EL-SAWY received the B.S. and M.S. degrees from Cairo University, in 2000 and 2003, respectively, and the Ph.D. degree in computer science from Suez Canal University, in 2011. He has been a Lecturer with Benha University, since 2012, where he also has been an Associate Professor in computer science, since 2017. From 2004 to 2011, he served as a Lecturer for the Faculty of Computers and Information Systems, Taif University, Saudi Arabia. Since 2017, he served as the Head of the computer science with the Faculty of Computers and Artificial Intelligence, Benha University, Egypt. His research interests include artificial intelligence in computer networks, bioinformatics, chimoinformatics, deep learning, and optimization.

AHMED SALIM received the B.Sc. degree in computer science and the M.Sc. degree in distributed computing from Zagazig University, Egypt, in 2001 and 2006, respectively, and the Ph.D. degree in systems, network, and telecommunication devices from the Bonch-Bruvich University of Telecommunication, Saint-Petersburg, Russia, in 2010. In 2012, he was a Consultant of the Information and Communication Technology Project (ICTP) at Zagazig University. From 2011 to 2019, he was an Assistant Professor with the Department of Mathematics, Faculty of Science, Zagazig University, where he has been an Associate Professor, since 2019. Since 2014, he has been an Assistant Professor with the Department of Mathematics, Faculty of Science and Art, Qassim University, Al-Mithnab, Saudi Arabia. His research interests include decomposable algorithms, computing, the IoT, and wireless sensor networks.