*Journal of Network Intelligence* ©2017 ISSN 2414-8105(Online)
**Taiwan Ubiquitous Information** Volume **2**, Number **3**, August **2017**

# An Optimal Deployment Wireless Sensor Network Based on Compact Differential Evolution

Trong-The Nguyen, Thi-Kien Dao, Trinh-Dong Nguyen, Truong-Giang Ngo
Department of Information Technology,
Haiphong Private University,
No. 35 Danlap Rd.,Hai-Phong, Vietnam
vnthe@hpu.edu.vn, jvnkien@gmail.com, dongnt@hpu.edu.vn, giangnt@hpu.edu.vn

Shu-Chuan Chu
School of Computer Science, Engineering and Mathematics
Flinders University, Australia
jan.chu@flinders.edu.au

Received March 2017; revised September 2017

Abstract. *This paper proposes a compact Differential Evolution (namely cED) for optimizing the deployment Wireless Sensor Network (WSN). The optimal scheme for deploying WSN should be a light and efficient algorithm because WSN limitations of size, memory, battery power, and computation. The proposed cDE uses a probabilistic model to generate candidate solutions for locating the promising area in search space. The solution of the population-based algorithm is expressed its distributed probability and is responding the order-one behavior for DE. So that cDE is a light and efficient tool that is suitable for deploying WSN. Simulation results are compared with the original and the other methods in the literature e.g. LEACH, LEACH-C, and HEED shows that the proposed method is the better performance regarding residual energy, nodes alive, and received items to save the energy of nodes.*

**Keywords:** Compact differential evolution, Deployment Wireless sensor network, Optimization.

1. **Introduction.** Wireless sensor networks (WSN) is an emerging, promising technology, and it is an essential infrastructure of Internet of Things (IoT) to collect relevant information in the target environment [1][2]. The applications of WSN have widely applied in a variety of fields of industry, traffic control, healthcare, and home automation [3][4]. However, the sensor nodes are limited on computation capability and storage capacity of computing unit, in communication range and radio quality of communication unit, in sensing coverage and accuracy of sensing unit, and in the available energy of power units [5]. Because the limited memory and the power constraints, WSNs fully functional network must be maintained and stable by the good design system employment. The problems arise from the insufficient memory of computational devices to store various candidate solutions for optimization applications. The required solutions to a complex optimization problem even though in limited hardware conditions have arisen from some applications [6].

The compact algorithm is a promising answer to these challenges [7]. An efficient compromise is used to present solutions of search space for the advantages of population-based algorithms without requirements of storing actual populations of solutions. Compact algorithm simulates the behavior of population-based algorithms by employing the replacement a population of solutions with its probabilistic representation [8]. The representation of candidate solutions is considered based on learning and probabilistic sampling models. A probabilistic model is built for the selected solutions. New solutions are generated by sampling the developed probabilistic model. The replacement strategy is used to incorporate new solutions into the virtual population. In the compact algorithms, the number of parameters stored in the memory is smaller than their corresponding algorithms of the population-based structures and the requirement devices capacious memory is less for a run. The scaling architecture and managing strategies are encouraged in designing and operating the extensive network with energy constrained and without recharged battery nodes. Therefore, optimal deployment WSN becomes an important factor for maintaining the stable lifetime of sensors. Clustering has proven to be an effective approach for organizing the network into a connected hierarchy to balance the load and prolong the network lifetime [9]. Clustering in WSNs involves grouping nodes into clusters and selecting a cluster head (CH). Thus, the collection of cluster heads in the system forms a connected dominating set. The most appropriate in the design and deployment and moderately relevant in clustering of the sensor network is the applied intelligence methods.

Differential evolution (DE)[10] is an intelligent algorithm that is a modern and powerful evolutionary optimizer for the continuous parameter spaces in recently. It also is an effective population-based intelligent optimization algorithm. The advantages of DE are as follows [11]. It is a straightforward algorithm whose implementation requires only a few lines of code in any standard programming language. DE needs very few control parameters e.g. the scale factor, the crossover rate, and the population size. The feature of DE makes it easy to use for the practitioners. Nevertheless, it has not considered the saving variable memory, so the optimal performance will not get high in optimal deployment problem in WSN. Moreover, challenges to the optimization applications have arisen from the limited hardware resource, whose condition due to the cost, storage and space restricted. Sensor nodes in WSN are an example of these limited hardware devices.

The objective of this paper is to get the optimal deployment WSN based on the saving variable memory such as compact DE. In the restricted hardware devices due to cost and space as WSN, each sensor node is connected to the only one closest cluster head after being decided CHs. The clustering evaluation model is formulated based on minimizing the average dissipated energy, and standard deviation of residual energy. The proposed method performance is compared with the previous clustering design protocols of low-energy adaptive clustering hierarchy (LEACH), LEACH-centralized (LEACH-C) [12] [13], and hybrid energy-efficient distributed clustering (HEED)[14].

The rest of the organized paper is as follows. Section 2 describes the related work with some theoretical formulations related to optimal deployment WSN and DE. Section 3 deals with compact DE for WSN in detail. The obtained simulation results from proposed method and the compared performance discussed in Section 4. The conclusion provided in Section 5.

## 2. **Related Work.**

### 2.1. **Deployment Evaluation Model for WSN.**

Applied developers concern the various design issues of WSN. Energy awareness is a critical design issue in WSN. Clustering is the most modern energy efficient technique and provides various advantages like energy efficiency, lifetime, scalability, and less delay; but it leads to hot spot problem. To overcome this issue, unequally clustering is proposed as a proven clustering of a practical approach for organizing the network into a connected hierarchy to balance the load and prolong the network lifetime In unequal clustering, the cluster size varies proportionally to the distance to BS. The dense deployment and unattended nature of WSN make it quite difficult to recharge node batteries. Therefore, energy efficiency is a major design goal in these networks. Clustering has been shown to improve network lifetime, a primary metric for evaluating the performance of a sensor network. Periodic re-clustering is necessary to heal disconnected regions and distribute energy consumption across all nodes [9].

$E_t(i)_{dissiapte}$ is the energy dissipated in the cluster head node $i(i \in CH)$ or non-cluster head node $n(i \in non - CH)$ during a single round $t$ where $l$ is the number of bits in each data message, $d_{(ntoBS)}$ is the distance from the cluster head node to the BS in Eq. (1). Each cluster head dissipates energy receiving signals from the nodes, aggregating the signals, and transmitting the aggregate signal to the base station BS which is a Sink. Since the BS is far from the nodes, presumably the energy dissipation follows the multi-path model ($d^4$ power loss). Each non-cluster head node only needs to transmit its data to the cluster head once during a round. Presumably the distance to the cluster head is small, so the energy dissipation follows Friss free-space model ($d^2$ power loss). $d_{(ntoCH)}$ is distance from the node to the cluster head. mn is the actual number of sensor nodes including cluster node that is connected to cluster head node CH. There are the communication energy parameters which are the initial energy for the nodes $E_j$, radio electronics dissipates for receiving and transmitting units $E_{elec}$, amplifier energy parameters $E_{mp}$ and $E_{fs}$, energy of data aggregation $E_{DA}$, number of bits in each data message $l = 1024bit$ [13].

$$E_t(i)_{dissipate} = \begin{cases} (m_n - 1)\, lE_{elec} + m_n lE_{DA} + lE_{elec} + lE_{mp} d^4_{i\ toBS}, & i \in CH \\ lE_{elec} + lE_{fs} d^2_{i\ to\ CH}, & i \in non - CH \end{cases} \tag{1}$$

where $i$ is node $i; t$ is round $t$. The average dissipated energy for round $t$ :

$$\mu\left(E_{dissipate}\right) = \left(\sum_{i\ \in N} E_t(i)_{dissipate}\right)/N \tag{2}$$

The residual energy for the round $(t + 1)$ :

$$E_{t+1}(i)_{residual} = E_t(i)_{residual} - E_t\ (i)_{dissipate} \tag{3}$$

The average residual energy for round t :

$$\mu(E_{residual}) = \sum_{n \in N} \frac{E_t(n)_{residual}}{N} \tag{4}$$

The standard deviation of residual energy:

$$\delta(E_{residual}) = SQRT\left(\sum_{n \in N} \frac{\left(\mu(E_{residual}) - E(i)_{residual}\right)^2}{N}\right) \tag{5}$$

The average dissipated energy,$\mu(E_{dissipate}$ is minimized to save the energy using Eq. (2) and standard deviation of residual energy, $\delta(E_{dissipate}$ is minimized to balance the energy load of nodes using Eqs. (3)-(5) in WSN. The average residual energy and number of received items is optimized to prolong the lifetime of sensor networks by applying this clustering evaluation model for measuring the performance.

2.2. **Differential Evolution Algorithm.** Differential evolution (DE) [9] which use biology-inspired operations e.g. crossover, mutation, and selection on a population to optimize an objective function over iterations. The performed evolution of DE has four operations included initialization, mutation, crossover, and selection.

Step1 *Initialization:* an initial population of $N$ agents is generated randomly. Each agent is a candidate solution containing $D$ dimension of unknown parameters. The population evolves through successive generations

$$\begin{array}{l} x_{j,i,0} = x_{j,min} + rand_j\,(0,1) \times (x_{j,max} - x_{j,min}) \\ j = 1, 2, ..D,\;\; i = 1, 2, ..N; rand_j\,(0,1)\,\tilde{}\,U\,(0,1) \end{array} \tag{6}$$

where $x_{j,i,0}$ is a vector indicates an agent in a population belonging to a current generation G. $x_{i,G} = [x_{1,i,G}, x_{2,i,G}, ..x_{D,i,D}]\,,\; i = 1, ..N; G = 1, ..G_{max}$ . All agents in a population are generated by enforcing the constraint of boundaries in which $x_{min} \leq x_{i,G} \leq x_{max}$ , where $x_{min}$ is set to $x_{1,min}, ..x_{D,min}$ and $x_{max}$ is set to $x_{1,max}, ..x_{D,max}$.

Step2 *Mutation:* after the initialization, DE runs a mutation to explore the search space. Some mutation strategies denoted as $DE/x/y/z$. It specifies the DE mutation strategies by indicating the vector $/x/$ to be perturbed, the number $/y/$ of difference vectors used to perturb $/x/$, and the type $/z/$ of crossover. In this paper, the original DE is considered. A vector $v_{i,G}$ is computed by each vector $x_{i,G}$

$$v_{i,G} = x_{r_{i1}G} + F \times (x_{r_{i2}G} + x_{r_{i3}G}) \tag{7}$$

where $F \in (0, 2)$ is a factor of scaling variable to speed up convergence of the DE; the indexes $r_{i,1}, r_{i,2} and r_{i,3}$ are mutually exclusive integers randomly selected from the interval $[1, N]$.

Step 3 *Crossover:* a crossover operation recombines agents to a new solution. It can make up increasing the diversity in the population but including successful solutions from the previous generation. Usually, DE adopts exponential or binomial crossover schemes. Here, the binomial crossover is used. It changes components that are chosen randomly from $\{1, 2, ..D\}$ and makes the number of parameters inherited from the mutant obey a nearly binomial distribution. A new candidate solution is calculated as given.

$$u_{j,i,G} = \begin{cases} v_{j,i,G} \; if \;\; rand_{j,i}\,(0,1) \leq CR \; or \; j = j_{rand} \\ \qquad\qquad x_{j,i,G} \; otherwise \end{cases} \tag{8}$$

where $u_{j,i,G}$ is new a trial vector that $x_{i,G}$ assumed as $[u_{1,i,G}, u_{2,i,G}, ...u_{D,i,G}]$, with $u_{i,G} \neq x_{i,G}$; $CR$ is crossover rate with $CR \in (0,1)\,; rand_{j,i}\,(0,1)\,\tilde{}\,U\,(0,1)\,; j_{rand} \in \{1, 2, \ldots, D\}$ The constant $0 < CR < 1$ obviously affects the amount of crossover operations. Usually, $0.6 < CR < 1$ is a good value for fast convergence.

Step 4 *Selection:* the population size constant is kept in consecutive generations. This operation determines if the vector $v_{i,G}$ or the vector $u_{i,G}$survives in the next generation. The selection operation works by the following relations.

$$x_{i,G+1} = \begin{cases} u_{i,G} \; if \;\; f\,(u_{i,G}) \leq f\,(x_{i,G}) \\ x_{i,G} \; if \;\; f\,(u_{i,G}) > f\,(x_{i,G}) \end{cases} \tag{9}$$

where $f(.)$ is the objective function to be optimized. If the value given by $u_{i,G}$ is lower than the value of $x_{i,G}$ then $u_{i,G}$ replaces $x_{i,G}$in the next generation, otherwise $x_{i,G}$ is kept. Therefore, the population can improve or be the same in optimization of the the the $f(.)$ but it never becomes worst. After selection, the algorithm goes back to iterate Step 2. *Mutation, crossover, and selection* are applied until a certain condition i.e. maximization of the number of generations $G_{max}$ or minimization of the objective function stops iterations.

3. **Compact DE for Optimal Deployment WSN.** Compact algorithms simulate the behavior of population-based algorithms by sampling probabilistic models for a population of solutions. An actual population of the solution is processed as a virtual population by encoding its probabilistic representation. Compact DE is constructed based on the framework of DE.

3.1. **Compact Differential Evolution.** The compact method process for evolution algorithm is to simulate the behavior of DE with a much smaller occupied memory. The population of solutions of the DE is described as a virtual population by encoding within a data structure, namely Perturbation Vector (PV). PV is the probabilistic model of a population of solutions. The compact DE maintains a real-valued prototype vector that represents the probability of each component being expressed in a candidate solution. As the DE progresses, agents fight with their competitors and their number in the population can go up or down depending on whether the DE makes good or bad decisions. These decisions are made implicitly by the DE when selection takes place. This is achieved by maintaining a vector that specifies the probability of including each component in a solution in new candidate solutions. Candidate solutions are probabilistically generated from the vector, and the elements in the better solution are used to make small changes to the probabilities in the vector.

The distribution of the individual in the hypothetical agents must be described by a probability density function (PDF)[15]. PDF is defined on the normalized interval from -1 to +1. The distribution of the each individual could be assumed as Gaussian PDF with mean $\mu$ and standard deviation $\delta$ [7]. A minimization problem is considered in an m-dimensional hyper-rectangle in Normalization of two truncated Gaussian curves. The parameters assume without loss of generality, to be normalized so that each search interval is arranged [-1,+1]. Therefore PV is a vector of $m \times 2$ matrix specifying the two parameters of the PDF of each design variable of mean and standard deviation. It is defined as $\mu^t, \delta^t$ The $\mu$and $\delta$ values are a Gaussian (PDF) truncated within the interval [-1, +1], respectively. The amplitude of the PDF is normalized in order to keep its area equal to 1. The apex t is time steps. The initialization of the virtual population is generated for each design variable $i, \mu_i^1 = 0$ and $\delta_i^1 = k$ where k is set as a large positive constant$(e.g., k = 10)$. The PDF height normalization is obtained approximately sufficient in well the uniform distribution with a wide shape. The generating for a candidate individual is produced from PV$(\mu_i, \delta_i)$. The value of mean $\mu$ and standard deviation $\delta$ in PV are associated equation of a truncated Gaussian PDF [16]. The codomain of CDF is arranged from 0 to 1.

***Update*** step of the compact DE has a constant size of $1/n$. While the DE needs to store n bits for each gene position, the compact DE only needs to keep the proportion of ones, a finite set of $n + 1$ numbers that can be stored with $\log_2 (n + 1)$ bits. While in many problems device memory is not a concern, it can be easily imagined large problems that need huge population sizes. In such cases, cutting down the memory requirement from n to logn results in significant savings. In the compact DE, the size of the update step is the thing that is analogous to the population size. The winner is indicated the vector that scores a better tness value and loser is indicated the individual losing. The update rule for each of its elements is $\mu_i^t, \delta_i^t => \mu_i^{t+1}, \delta_i^{t+1}$ :

$$\mu_i^{t+1} = \mu_i^t + \frac{1}{n}\left(winner_i - loser_i\right) \tag{10}$$

where n is virtual population size. Regarding $\delta$values, the update rule of each element is given by:

$$\delta_i^{t+1}\text{=SQRT(}\left(\left(\delta_i^t\right)^2 + \left(\mu_i^t\right)^2 - \left(\mu_i^{t+1}\right)^2 + \frac{1}{n}\left(winner_i^2 - loser_i^2\right)\right) \tag{11}$$

In elitist compact schemes, at each moment of the optimization process, the solution displaying the best performance is retained in a separate memory slot. If a new candidate solution is computed, the tness based comparison between it and the elite is carried out. cDE employs a probabilistic model to represent the solution set. Moreover, only an agent is used in the whole algorithm. Thus, a modest memory space is well suited for the embedded equipment with limited hardware.

***Crossover*** in the DE is to combine bits and pieces from fit solutions. A repeated application of most commonly used crossover operators eventually leads to a de-correlation of the population's agents. In this de-correlated state, the population is more compactly represented as a probability vector. Thus the generation of individuals from this vector can be seen as a shortcut to the eventual aim of crossover.

***Selection*** gives more copies to better individuals. However, it does not always do so for better agents. This is because agents are always evaluated within the context of a larger individual. Suppose individual a competes with individual b. When these two individuals compete, individual $a$ will win. At the level of the gene, however, a decision error is made on the second position. The role of the population is to buffer against a finite number of such decision errors. Imagine the following selection scheme: pick two individuals randomly from the population, and keep two copies of the better one. This scheme is equivalent to a steady-state binary tournament selection. In a population of size, the proportion of the winning alleles will increase by $1/n$. An update rule increasing a gene's proportion by $1/n$ simulates a small step in the action of a DE with a population of size $n$.

3.2. **Solution Model of Deployment WSN.** Because the limited memory and the power constraints of sensor nodes, a prolonging the lifetime is a core demand in designing and deploying sensor networks [17]. A crucial factor to extend the lifetime of WSNs is to reduce the energy consumption of its entire network. Power consumption of WSNs is affected directly by the clustering criterion problem. The clustering formation optimal problem would be solved by the cDE application. Equation (2) is the average dissipated energy, $\mu(E_{dissipate}$ which should be minimized to save the energy. Equations (3)-(5) are given the standard deviation of residual energy, $\delta(E_{residual})$ which should be minimized to balance the energy load of nodes using in WSN. The average residual energy and number of received items is optimized to prolong the lifetime of sensor networks by applying this clustering evaluation model for measuring the performance.

3.3. **Objective function.** The heuristic clustering methods are employed by evaluating the objective function. The objective function has also evaluated the performance of wireless sensor network. This feature is composed of the average dissipated energy for round t in Equation (2) and standard deviation of residual energy in Eq.(5). It is the binary integer programming model with a binary decision variable $x_i = 1$ of cluster head and $x_i = 0$ of the sensor for each node $i$ in WSN. Each sensor node member is connected to the only one closest cluster head after the cluster heads are decided. Eq. (1) is used to get $E_t(n)_{dissipate}$ with cluster head node ($i \in CH, x_i = 1$) or non-cluster head node ($i \in non - CH, x_i = 0$).

$$Minimize \quad \mathrm{w}_1 \times \mu(E_{dissipate}) + \mathrm{w}_2 \times \delta(E_{residual}) \tag{12}$$

where $w_1$ and $w_2$ are the weight of the average dissipated energy and standard deviation of residual energy. $w_1 + w_2$ is set to 1. The residual energy of round 11 - 12 and dissipated the energy of round 11 are shown in Table 1 for ten node network example. We can get the $\mu(E_{dissipate} = 0 : 000113, \mu(E_{residual}) = 0.49746, \delta(E_{residual} = 0 : 000157$. The objective value of Equation (9) is 0.000265 (with $w_1 and w_2$ are set to 0.5) for 10 nodes network.

Table 1. Residual and dissipated energy of 11 -12 round of 10 node network example

| No. of node | $E_{11}$ (n)residual | $E_{11}$ (n)dissipate | $E_{12}$ (n)residual |
|---|---|---|---|
| 1 | 0.497465 | 0.000080 | 0.497385 |
| 2 | 0.497564 | 0.000055 | 0.497509 |
| 3 | 0.497496 | 0.000060 | 0.497436 |
| 4 | 0.497616 | 0.000214 | 0.497402 |
| 5 | 0.497552 | 0.000078 | 0.497474 |
| 6 | 0.497548 | 0.000078 | 0.497470 |
| 7 | 0.497430 | 0.000374 | 0.497056 |
| 8 | 0.497127 | 0.000058 | 0.497069 |
| 9 | 0.497501 | 0.000073 | 0.497428 |
| 10 | 0.497388 | 0.000064 | 0.497324 |

3.4. **The Expressed Agents.** A wireless sensor network is modeled a graphs G with n nodes in the distributed randomly of desired areas. Table 2 shows the position of nodes in network areas. Each node can communicate with others by using $r$ transmission range. Node i can receive the signal of node $j$ if node $i$ is in the transmission range $r$ of node $j$. Table III indicates the attribution of existing CHs in the sensor networks.

Table 2. A sample of expressing the positions of the sensor nodes

| Index | $\mathrm{Node_i}$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | .. | n |
|---|---|---|---|---|---|---|---|---|---|---|
| x | 05 | 65 | 95 | 100 | 75 | 60 | 45 | 40 | .. | 10 |
| y | 01 | 5 | 10 | 30 | 15 | 20 | 45 | 60 | .. | 80 |

Table 3. The attribution of existing cluster head (CHs) if flag =1, set Node is set to cluster head (CH), otherwise Node configured to member node (SN)

| Index | $\mathrm{Node_i}$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | .. | n |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Cluster head | | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | .. | 0 |

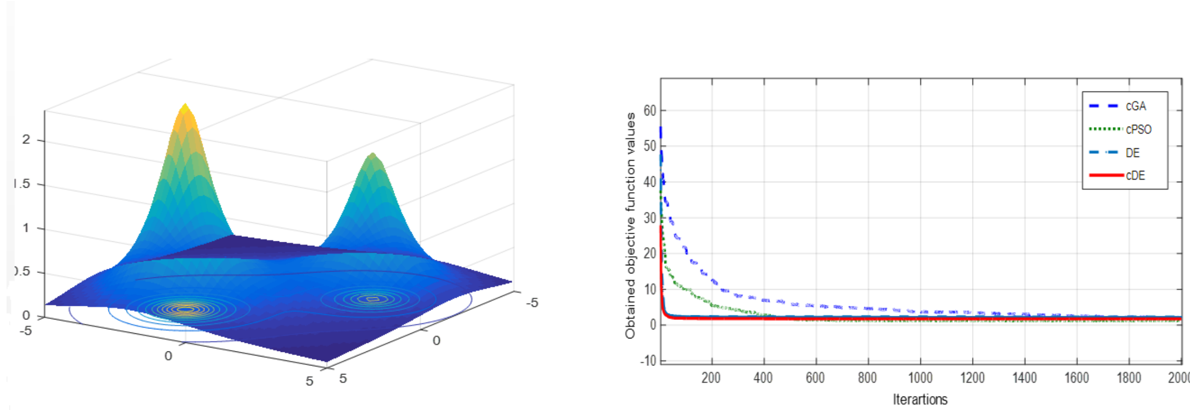



FIGURE 1. Comparison convergence for optimizing clustering formation in WSNs of the proposed approach cDE with the other methods, e.g., the original approach DE, cGA, and cPSO

3.5. **Optimal clustering formation.** The details steps of cDE for optimizing the clustering formation are given below:
Step 1: initialized PV as a probabilistic model vector,
Step 2: generated agent a by PV; assigned Fmin to fitness(a),
Step 3: generated agent b by PV;
Step 4: competed a, b: winner, loser = compete(a,b),
Step 5: updated PV according to Eqs. (7) and (8).
Step 6: Accepted a new solution as Eq. (9) if the improved solution is ok ($F_{new} less than F_{min}$), and assign the minimum function $F_{min} to F_{new}$
Step 7: if it is not meet the termination condition, go to step 3.
Step 8: output the best agent

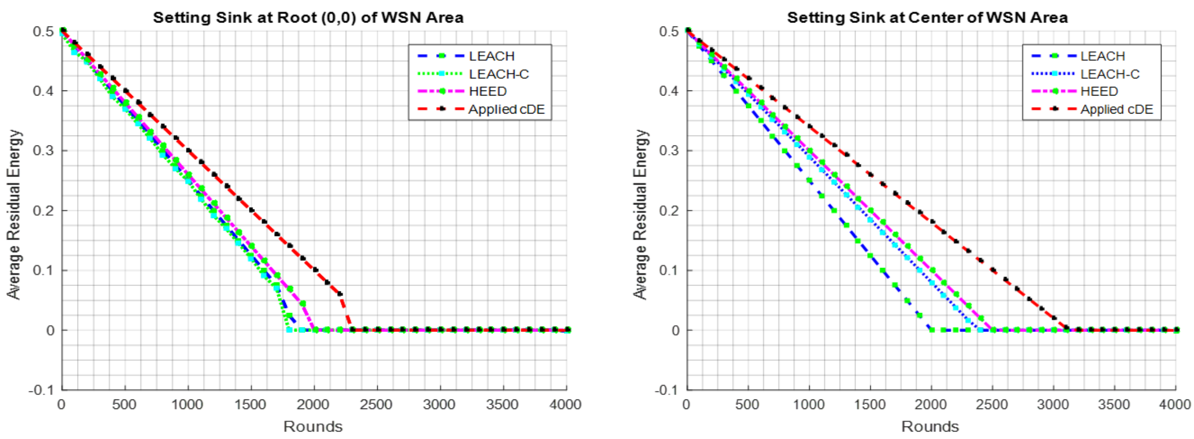
4. **Experimental Results.** Simulation of the network with N-node ($N = 100, 200..$) are distributed in a 2-Dimensional problem space [0:100,0:100]. There are N deployed nodes in the target network for establishing a n x n grid space test platform, where nodes were randomly distributed between ($x = 0, y = 0$) and ($x = n, y = n$) with the Sink with $N$ set to 100, 200, 300 and 400 node networks with $Sink(0,0)$ and Sink (Center of nodes). The objective function is in Equation (9) repeated the generations of 2000 by different random seeds with 25 runs. The initial values of communication energy parameters, $E_j = 0 : 5J, E_{elec} = 50nJbit, E_{mp} = 0 : 0013pJ/bit/m^4, E_{fs} = 10pJ/bit/m^2, E_{DA} = 5nJ/bit/signal, l = 1024bit$ [13]. The final result obtained by taking the average of the outcomes from all runs. The results are compared with that its running in the original DE, compact genetic algorithm (cGA) [7], compact particle swarm optimization (cPSO)[18].

Figure 1 compares the proposed method of clustering formation base on cDE for WSN with that applied in original DE, cGA, and cPSO. Apparently, the proposed method curve is faster than those obtained in the other methods regarding convergence.

The test case with various grid sizes is considered as effective of solution search rate. That mean the number of sensor nodes was increased or decreased accordingly to different N values were verified to evaluate the effectiveness, timeliness, and reliability. Table 4 illustrates the cases of $N$ from 100 to 400 are respectively. It means to increase the density of node in a cluster; the chances for nodes become cluster head will be higher. The number of cluster-heads is about 10.0% of the total number of nodes; the percentage may vary if nodes are unevenly distributed. On average, the distance is reduced by 80% as compared

TABLE 4. Variant $N$ values were verified to evaluate the effectiveness of convergence and solution search rate

| Number of Nodes | Convergence after generation | Cluster Head (%) | Distance decreased | Solution Search rate |
|---|---|---|---|---|
| 100 | 180 | 10.0% | 77.85% | 100% |
| 200 | 350 | 10.0% | 81.20% | 100% |
| 300 | 450 | 10.0% | 82.10% | 100% |
| 400 | 640 | 11.2% | 83.20% | 99.2% |




a) The scenario of setting Sink at (0,0) of the network zone

b) The scenario of setting Sink at the center of the network zone

FIGURE 2. Comparison of consumed average residual energy of the proposed cDE for 100 nodes with LEACH, LEACH-C, HEED methods

with the distance of direct transmission. This percentage increases slightly as node count increases because of the larger the number of nodes and denser node distribution results in more efficient cluster optimization. As expected, in an application where nodes are densely distributed, reducing the number of heads tends to increase the solution quality significantly.

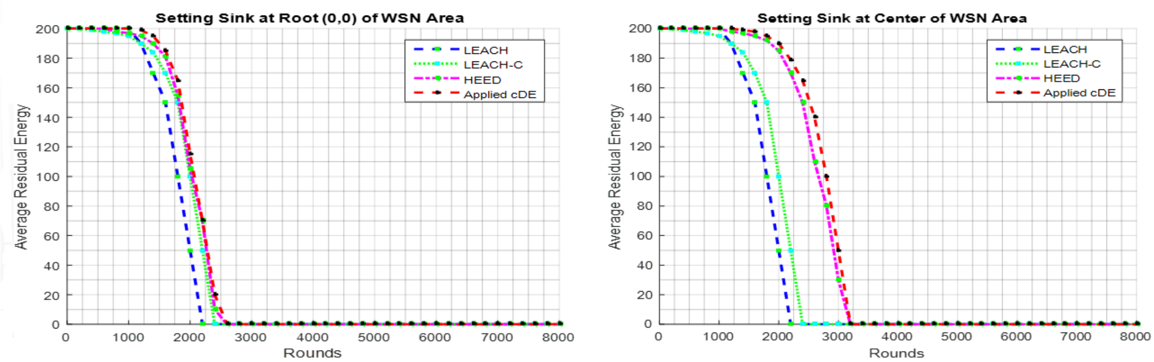
The other experiments, the performances of proposed method can be compared to previous methods (LEACH, LEACH-C [12][13], and HEED [14]), as illustrated in Tables 5, 6, and 7, in which FND and LND are the first node die and last node die respectively. The energy consumption in the network is focused on by the cluster heads. Figs. 2, 3, and 4 compare the network average residual energy of performance measures for 100 node networks of LEACH, LEACH-C, HEED and the proposed cDE methods with two cases of $Sinkin(0,0)$ and in the center. Obviously, the average residual energy consumption of proposed method of cDE optimized is better than those obtained from LEACH, LEACH-C, HEED, for the 100 node network with $Sink(0,0)$ and from LEACH, LEACH-C for Sink in center respectively.

Figure 3 shows the proposed cDE method performance for 200 node networks regarding a number of nodes alive in comparison with those obtained from LEACH, LEACH-C, HEED methods. Obviously, the figure of the proposed method is better than those obtained from LEACH, LEACH-C in both cases of Sink (0, 0) and Sink in the center.

Fig.4 compares the network average number of received items for 200 node networks of LEACH, LEACH-C, HEED and the proposed cDE methods. Obviously, the number of received items of the proposed method outperforms the other methods.

TABLE 5. Comparisons of performance using different methods for 100 node networks

| Number of nodes | Clustering methods | Round of FND | Round of LND | Total data |
|---|---|---|---|---|
| 100 (0,0) | LEACH | 3503 | 3901 | 383431 |
| | LEACH-C | 4141 | 4271 | 414947 |
| | HEED | 3683 | 4431 | 432563 |
| | **Proposed cDE** | **4329** | **4445** | **439379** |
| 100 (center) | LEACH | 3585 | 4181 | 404223 |
| | LEACH-C | 4308 | 4333 | 428428 |
| | HEED | 4611 | 6799 | 669817 |
| | **Proposed cDE** | **4611** | **6617** | **654978** |




a) Comparison the number of nodes alive for 200 network nodes of the proposed cDE with LEACH, LEACH-C, HEED methods with Sink in root (0, 0)

b) Comparison the number of nodes alive for 200 network nodes of the proposed cDE with LEACH, LEACH-C, HEED methods with Sink in center ($x_{max}/2$, $y_{max}/2$)

FIGURE 3. Comparison the number of nodes alive for a configured network with 200 node

TABLE 6. Comparisons of performance using different methods for 200 node networks

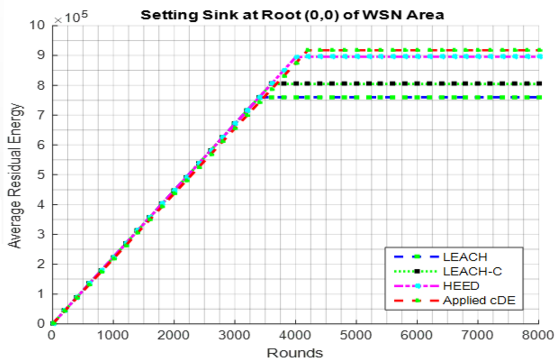
| Number of nodes | Clustering methods | Round of FND | Round of LND | Total data |
|---|---|---|---|---|
| 200 (0,0) | LEACH | 1030 | 3100 | 519376 |
| | LEACH-C | 2119 | 4517 | 589751 |
| | HEED | 855 | 2198 | 381256 |
| | **Proposed cDE** | **2904** | **3336** | **600930** |
| 200 (center) | LEACH | 3508 | 3700 | 734792 |
| | LEACH-C | 3011 | 3632 | 718482 |
| | HEED | 3285 | 4444 | 854271 |
| | **Proposed cDE** | **4302** | **4310** | **857349** |

The clustering method is better if the round of the first node dies (FND), a round of last node dies-(LND) and total data messages received are bigger than the others. Observed from Tables 5, 6 and 7, the performance of proposed method is better than other methods.

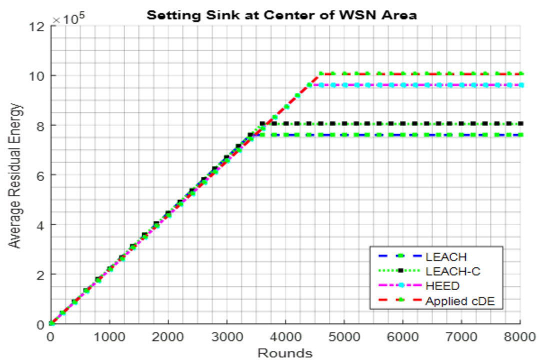
5. **Conclusions.** In this paper, we proposed a solution to the optimization deployment wireless sensor networks (WSN) with hierarchical clustering formation based on compact Differential Evolution (cDE). To prolong life time network, the saving variable memory

TABLE 7. Comparisons of performance using different methods for 400 node networks

| Number of nodes | Clustering methods | Round of FND | Round of LND | Total data |
|---|---|---|---|---|
| 400 (0,0) | LEACH | 46 | 2640 | 395594 |
| | LEACH-C | 317 | 2858 | 554136 |
| | HEED | 94 | 1407 | 302591 |
| | **Proposed cDE** | **560** | **6681** | **560474** |
| 400 (center) | LEACH | 888 | 3540 | 1041461 |
| | LEACH-C | 1390 | 2763 | 1040614 |
| | HEED | 843 | 2485 | 799151 |
| | **Proposed cDE** | **2202** | **3204** | **1103129** |




a) Compared the received data items for 400 nodes of the proposed cDE with LEACH, LEACH-C, HEED and methods with Sink in root (0,0)




b) Compared the received data items for 400 nodes of the proposed cDE with LEACH, LEACH-C, HEED and methods with Sink Center

FIGURE 4. Comparison the number of received data items for 400 node networks of the draft cDE with LEACH, LEACH-C, HEED methods

such as in the cDE is targeted to deal with the optimization of limited hardware devices like WSN. The objective function was proposed based on clustering evaluation model with a binary decision variable to minimize the average dissipated energy and standard deviation of residual energy of the limited hardware devices such as sensor nodes in WSN. In this proposed method, we represented the population as a probability distribution over the set of solutions and operationally equivalent to the order-one behavior of the DE. Generated new candidate solutions by learning explicit probabilistic models of promising solutions found so far and sampling the built models are used to optimize. The best clustering solution with proposed objective function is to deal with saving and balancing the energy consumption in WSN. Simulation results compared with the original, and the other methods in the literature as LEACH, LEACH-C, and HEED show that the proposed algorithm provides the effective way of using a limited memory. The proposed approach also got the better performance regarding the average residual energy, the number of nodes alive, and the number of received items to save the energy of nodes.

## REFERENCES

[1] J.Yick, B.Mukherjee, and D.Ghosal, Wireless sensor network survey, *Comput. Networks,* vol. 52, no. 12, pp. 2292-2330, 2008.

[2] T.-T.Nguyen, T.-K.Dao, M.-F.Horng, and C.-S.Shieh, An Energy-based Cluster Head Selection Algorithm to Support Long-lifetime in Wireless Sensor Networks, *journal of Netw. Intell.,* vol. 1, no. 1, pp. 23-37, 2016.

[3] C. F.Garca-hernndez, P. H.Ibargengoytia-gonzlez, J.Garca-hernndez, and J. aPrez-daz, Wireless Sensor Networks and Applications: a Survey, *journal of Computuer Science,* vol. 7, no. 3, pp. 264-273, 2007.

[4] T.-T.Nguyen, J.-S.Pan, S.-C.Chu, J. F.Roddick, and T.-K.Dao, Optimization Localization in Wireless Sensor Network Based on Multi-Objective Firefly Algorithm, *journal of Network Intellegent,* vol. 1, no. 4, p. 130-138, 2016.

[5] R.V.Kulkarni, A.Frster, and G. K.Venayagamoorthy, Computational intelligence in wireless sensor networks: A survey, *IEEE Commun. Surv. Tutorials,* vol. 13, no. 1, pp. 68-96, 2011.

[6] T. K. Dao, T.S. Pan, and T. T. Nguyen, A Compact Articial Bee Colony Optimization for Topology Control Scheme in Wireless Sensor Networks, *journal of Inf. Hiding Multimed. Signal Process.,* vol. 6, no. 2, pp. 297-310, 2015.

[7] G. R.Harik, F. G.Lobo, and D. E.Goldberg, The compact genetic algorithm, *IEEE Trans. Evol. Comput.,* vol. 3, no. 4, pp. 287-297, 1999.

[8] F.Neri, G.Iacca, and E.Mininno, Compact Optimization, *in Intelligent Systems Reference Library,* vol. 38, I.Zelinka, V.Snael, and A.Abraham, Eds.Springer Berlin Heidelberg, 2013, pp. 337-364.

[9] O.Younis, M.Krunz, and S.Ramasubramanian, Node clustering in wireless sensor networks: Recent developments and deployment challenges, *IEEE Netw.,* vol. 20, no. 3, pp. 20-25, 2006.

[10] R.Storn and K.Price, Differential evolutiona simple and efficient heuristic for global optimization over continuous spaces, *J. Glob. Optim.,* vol. 11, no. 4, p. 341-359., 1997.

[11] K.Price, R. M.Storn, and J. A.Lampinen, *Differential Evolution: A Practical Approach to Global Optimization.* Springer Science & Business Media., 2006.

[12] W. R.Heinzelman, A.Chand rakasan, and H.Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.* , vol. 0, no. c, pp. 3005-3014, 2000.

[13] W. B.Heinzelman, A. P.Chandrakasan, and H.Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, *IEEE Trans. Wirel. Commun.,* vol. 1, no. 4, pp. 660-670, 2002.

[14] O.Younis, Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach HEED , *third Annu. Conf. IEEE,* pp. 1-12, 2004.

[15] S.Grosskinsky, A.Stacey, and G.Grimmett, Probability And Measure - Billingsley.pdf. *John Wiley and Sons, New York, Toronto, London,* pp. 1-66, 2006.

[16] W. J.Cody, Rational Chebyshev approximations for the error function, *Math. Comput.,* vol. 23, no. 107, pp. 631-631, 1969.

[17] T.Arampatzis, J.Lygeros, and S.Manesis, A Survey of Applications of Wireless Sensors and Wireless Sensor Networks, *Proc. 2005 IEEE Int. Symp. on, Mediterrean Conf. Control Autom. Intell. Control.* 2005., pp. 719-724, 2005.

[18] G.Iacca, Compact Particle Swarm Optimization Compact Particle Swarm Optimization, *journal of Inf. Sci. (Ny).* , vol. 239, no. October 2015, pp. 96-121, 2013.