IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 30, 2018, accepted November 30, 2018, date of publication December 10, 2018, date of current version January 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2885934

Machine-Learning-Based Parallel Genetic Algorithms for Multi-Objective Optimization in Ultra-Reliable Low-Latency WSNs

YUCHAO CHANG[1,2], XIAOBING YUAN[1,2], BAOQING LI[1,2], DUSIT NIYATO[3], (Fellow, IEEE), AND NAOFAL AL-DHAHIR[4], (Fellow, IEEE)

[1]Science and Technology on Microsystem Laboratory, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798
[4]Department of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson, TX 75080, USA

Corresponding author: Xiaobing Yuan (sinowsn@mail.sim.ac.cn)

The work of Y. Chang was supported by the University of Chinese Academy of Sciences Joint Ph.D. Training Program. The work of N. Al-Dhahir was supported by the NPRP under Grant 8-627-2-260 from the Qatar National Research Fund (a member of the Qatar Foundation).

ABSTRACT Different from conventional wireless sensor networks (WSNs), ultra-reliable and low-latency WSNs (uRLLWSNs), being an important application of 5G networks, must meet more stringent performance requirements. In this paper, we propose a novel algorithm to improve uRLLWSNs' performance by applying machine learning techniques and genetic algorithms. Using the K-means clustering algorithm to construct a 2-tier network topology, the proposed algorithm designs the fetal dataset, denoted by the population, and develops a clustering method of energy conversion to prevent overloaded cluster heads. A multi-objective optimization model is formulated to simultaneously satisfy multiple optimization objectives including the longest network lifetime and the highest network connectivity and reliability. Under this model, the principal component analysis algorithm is adopted to eliminate the various optimization objectives' dependencies and rank their importance levels. Considering the NP-hardness of wireless network scheduling, the genetic algorithm is used to identify the optimal chromosome for designing a near-optimal clustering network topology. Moreover, we prove the convergence of the proposed algorithm both locally and globally. Simulation results are presented to demonstrate the viability of the proposed algorithm compared to state-of-the-art algorithms at an acceptable computational complexity.

INDEX TERMS Machine learning (ML), genetic algorithms (GAs), multi-objective optimization, near-optimal clustering network topology, ultra-reliable and low-latency wireless sensor networks (uRLLWSNs).

I. INTRODUCTION

The number of networked entities is reaching unprecedented levels, resulting in great challenges for ultra-reliable and low-latency Internet of Things (uRLLIoT) applications. Integrating both enhanced mobile broadband (eMBB) and massive machine-type communications (mMTC), ultra-reliable and low-latency communications (uRLLC) has been identified as a key 5G feature by the International Telecommunication Union (ITU) [1]–[4]. Being a crucial part of uRLLIoT, ultra-reliable and low-latency wireless sensor networks (uRLLWSNs) deserve special investigation. The uRLLWSNs evolution is required not only to allow a near-optimal operation in the monitoring environment, but also to conduct further extensions and enhancements, especially for hazardous scenarios such as volcano monitoring. In such scenarios, it is difficult to replace or recharge the sensor battery while being subject to ultra-reliability and low-latency requirements. Hence, further research on uRLLWSNs is indispensable for simultaneously satisfying multiple optimization objectives such as the longest network lifetime, the highest network connectivity and reliability, and so forth. One effective approach to enhance uRLLWSNs is to design efficient routing schemes based on the clustering network topology [5]–[10].

Reviewing the literature, the LEACH (low-energy adaptive clustering hierarchy) protocol in [11] optimizes the energy consumption by dynamically creating clusters. Although

TABLE 1. The key strengths and weaknesses of the various algorithms.

Algorithms	Strength	Weakness
LEACH	Dynamic and simple CHs slection	Unevenly distributed CHs
HEED	Dynamic and fast CHs slection	Suitable to small-scale WSNs and unreliable clustering
LELE	Dynamic CHs selection	Unevenly distributed CHs
DORAHP	Suitable to decentralized multi-hop networks	Suitable to small-scale WSN and high network latency
EEUC	Suitable to multi-hop WSNs	High network latency
GASONeC	Low network latency	High computation complexity
HHCA	Balance the communication load on sensors	Lacks analysis of some important network metrics
ULGAT	Suitable to large-scale WSNs	Over-textitasizing single-objective optimization

LEACH selects sensors as cluster headers (CHs) in the autonomous and decentralized mode, it designs unevenly distributed CHs [12]. The HEED (hybrid energy-efficient distributed clustering) algorithm in [13] adjusts the transmit power levels of a node to improve energy efficiency by considering its residual energy and the number of its neighbors. Although HEED can improve the clustering speed, it excludes some sensors from joining any clusters due to the clustering competition [14]. The LELE (leader election with load balancing energy) algorithm in [15] selects CHs by considering the node residual energy and transmission distance, but suffers from the issue of unevenly distributed CHs [14]. DORAHP, the distributed joint optimization routing algorithm based on the analytic hierarchy process [16], in [17] selects the next hop based on the key criteria of energy, distance, and the number of neighbors. It is suitable for small-scale WSNs and suffers from high network latency due to multiple forwarding operations [18]. The EEUC (energy-efficient unequal clustering) algorithm in [19] utilizes the distance between sensors and the base station (BS) to select CHs, but is similar to DORAHP in suffering from the network latency problem. The GASONeC (genetic algorithm-based self-organizing network clustering) method in [12] provides a framework to optimize clusters. However, its computational complexity is significantly high due to more evolutionary generations [14]. The HHCA (hybrid hierarchical clustering approach) in [20] optimizes network topology to balance the communication load and to increase the network lifetime by designing a three-layer hierarchy, but it lacks analysis of some important network metrics such as network coverage and reliability. To improve the ultra-dense WSN energy efficiency, our prior work in [18] utilizes both unsupervised learning and genetic algorithms, named ULGAT, to identify a near-optimal network topology. However, it is a single-objective optimization and only suitable for the WSN scenario with the deterministic deployment of sensed objects, resulting in over-textitasizing the importance of energy efficiency. The key strengths and weaknesses of the various algorithms are presented in TABLE 1.

To study uRLLWSNs with many stringent requirements, we propose a multi-objective optimization algorithm by applying machine learning techniques and genetic algorithms to identify a near-optimal clustering network topology. The algorithm satisfies multiple optimization objectives including the longest network lifetime and the highest network connectivity and reliability. The proposed algorithm, called MLPGA, has the following main advantages

1) Formulating the plane network as a two-dimensional (2D) graph, the proposed algorithm utilizes the popular K-means clustering algorithm of machine learning to design the 2-tier network topology for encoding sensors as the chromosome. Various chromosomes for diverse network topologies construct the population, from which the optimal chromosome is identified to design a near-optimal network topology for routing scheduling.
2) The proposed algorithm develops a clustering method of energy conversion to transform the CH's communication energy consumption into virtual CMs. Integrating both virtual CMs and real CMs, a clustering network topology without overloaded CHs is designed. Based on the multiple investigated network objectives, a fair optimization model is developed to identify the optimal chromosome for designing a near-optimal clustering network topology without overloaded CHs. Using the principal component analysis (PCA) algorithm of machine learning, the proposed algorithm eliminates dependencies between the multiple optimization objectives and ranks their importance levels to construct the fitness function for evaluating different chromosomes. The minimal schema is defined as the convergence condition of identifying the optimal chromosome in this optimization model. In addition, considering the NP-hardness of nonlinear multi-objective optimization, a genetic algorithm is adopted to learn the optimal chromosome by using selection, crossover, and mutation procedures.
3) The convergence property of the proposed algorithm is proved both locally and globally. Simulation results demonstrate that the proposed algorithm has an acceptable complexity while outperforming state-of-the-art algorithms in terms of the network lifetime, the number of alive sensors, the energy consumption, and the network connectivity and reliability. In addition, we analyze the effect of PCA on improving the network performance.

The remainder of this paper is organized as follows. Section II describes the network model and the proposed algorithm. The convergence property of the proposed algorithm is proved in Section III. Simulation results are presented in Section IV and the paper is concluded in Sections V.

II. MODEL AND PROPOSED ALGORITHM

In this section, we encode sensors as the chromosome of genetic algorithms and develop a clustering method of energy conversion to prevent overloaded CHs. In addition, a multi-objective optimization model satisfying the stringent uRLLWSNs requirements is formulated to identify the optimal chromosome for designing a near-optimal clustering network topology by using the genetic algorithm. Based on machine learning principles [21], the proposed MLPGA is composed of the dataset, the cost function, and the optimization model and procedures. In addition, MLPGA is a centralized scheme using a static optimization mode that prelearns a near-optimal clustering network topology without overloaded CHs.

A. NETWORK MODEL AND ENERGY MODEL

We formulate the plane network as a 2D graph $\mathcal{G}(\mathbb{V}, \mathbb{E})$, where $\mathbb{V} = \{v_0, v_1, \ldots, v_n, \ldots, v_N\}$ represents the set of sensors and $\mathbb{E}$ represents the set of communication links between sensors [22]. The routing scheme is defined as the process of learning the optimum routing path from sensor $v_n, n = 1, \ldots, N$, to the gateway or BS v_0. We make the following assumptions:

1) Communication links for neighbor sensors are set up by the broadcast that includes the basic sensor information such as the current energy and location of the sensor [22];
2) Sensors can adjust the amount of transmission power using power control according to the distance between the transmitter and the receiver [23];
3) Sensors are equipped with the Global Position System (GPS) to be location-aware, and are identical in terms of hardware, software, and energy storage while the BS can be manually maintained [17].

Sensors are denoted as $\boldsymbol{s} = [s_1, \ldots, s_n, \ldots, s_N]^T$, termed a chromosome, where s_n represents the CH when $s_n = 1$ and s_n represents the CM when $s_n = 0$. Massive chromosomes can establish the population that is the fetal dataset for the multi-objective optimization model. We represent a population of the m chromosomes as follows

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{s}_1^T \\ \vdots \\ \boldsymbol{s}_m^T \\ \vdots \\ \boldsymbol{s}_M^T \end{bmatrix} = \begin{bmatrix} [s_{1,1} & \ldots & s_{1,n} & \ldots & s_{1,N}] \\ & & \vdots & & \\ [s_{m,1} & \ldots & s_{m,n} & \ldots & s_{m,N}] \\ & & \vdots & & \\ [s_{M,1} & \ldots & s_{M,n} & \ldots & s_{M,N}] \end{bmatrix}. \tag{1}$$

The CH directly communicates with the BS whereas each CM joins one cluster. In general, the CM joins the nearest cluster, being prone to construct an unreasonable clustering network topology due to some overloaded CHs. CHs close to the BS consume less communication energy while those distant from the BS consume more communication energy leading to their premature death. To design the network topology without overloaded CHs, a clustering method of energy conversion is designed by considering both the CH's transmission energy consumption and the distance. In this algorithm, each CH's transmission energy consumption is transformed into some virtual CMs. For the WSN energy model, in addition to the fixed energy consumption associated with data acquisition and processing, two key system parameters are the distance and the message size for WSN routing scheduling. According to these two parameters, Heinzelman *et al.* [24] proposed a simplified energy loss model, where the energy consumptions for transmitting and receiving the l-bit message over a transmission distance d are, respectively, given by

$$E_{tx}(l, d) = \begin{cases} lE_{elec} + l\epsilon_{fs}d^2, & d < d_0 \\ lE_{elec} + l\epsilon_{mp}d^4, & d \geq d_0, \end{cases} \tag{2a}$$

$$E_{rx}(l) = lE_{elec}, \tag{2b}$$

where E_{elec} is the energy consumption due to data acquisition and processing, while the amplifier energy, given by $\epsilon_{fs}d^2$ or $\epsilon_{mp}d^4$, depends on the distance d and the specified bit-error rate, and $d_0 = \sqrt{\epsilon_{fs}/\epsilon_{mp}}$ is the threshold distance. Typical values for these parameters are given in TABLE 3. For example, the number of virtual CMs for CH $v_{c,p}$ is given by

$$n_p = E_{tx}(l, d(v_{c.p}, v_0))/E_{rx}(l), \tag{3}$$

where $d(v_{c,p}, v_0)$ is the distance from CH $v_{c,p}$ to the BS v_0. According to the sum of the real CMs and virtual CMs, the average number of CMs for each cluster is computed. Each CM joins a nearer cluster whose number of CMs must be not greater than the average number of CMs. The clustering method of energy conversion is presented in Algorithm 1.

B. MULTI-OBJECTIVE OPTIMIZATION MODEL

In general, the network coverage, the network connectivity and reliability, and the network latency are major performance metrics of uRLLWSNs [25]–[27]. Furthermore, the energy efficiency of sensors is equally important because it is difficult to recharge or replace the sensor battery [28]. For some monitoring applications such as seismic wave, sensors conduct the detection task periodically, for example every hour. The network lifetime is defined as the number of transmission rounds from the network birth until sensor deaths lead to network coverage failure [18]. A sensed object is covered by the minimum number of sensors, needed for network coverage. Hence, it is denoted as the critical sensed object [29]. Assume that sensed object o_τ is the critical sensed object and its collection of sensors is denoted by $\mathbb{V}_\tau = \left\{v_{\tau,1}, \ldots, v_{\tau,k}, \ldots, v_{\tau,K}\right\}$. All sensors in collection $\mathbb{V}_\tau$

Algorithm 1 The Clustering Method of Energy Conversion

```
Require: The collection of CHs $\boldsymbol{V}_c = \{v_{c,1}, \ldots, v_{c,P}\}$, the
    collection of real CMs $\boldsymbol{V}_m = \{v_{m,1}, \ldots, v_{m,Q}\}$.
Ensure: The clustering network topology of balanced load.

for $p = 1$ to $P$ do
    Compute the number of virtual CMs $n_p$.
end for
Compute the average number of CMs for each cluster:
$\bar{n} = (Q + n_1 + \cdots + n_p + \cdots n_P)/P$.
for $q = 1$ to $Q$ do
    Order CHs according to the distances between CM
    $v_{m,q}$ and CHs in the ascend mode and compute their
    respective number of CMs: $n'_1, \ldots, n'_p, \ldots n'_P$.
    for $p = 1$ to $P$ do
        if $n'_p < \bar{n}$ then
            CM $v_{m,q}$ joins the $p^{th}$ cluster: $\boldsymbol{V}_p = \boldsymbol{V}_p \cup \{v_{m,q}\}$.
        end if
    end for
end for
return The clustering network topology
    $\boldsymbol{V}_c = \bigcup_{p=1}^{P} \boldsymbol{V}_p$,
    $\boldsymbol{V}_p = \{v_{p,1} \ldots, v_{p,k}, \ldots\}$,
    $\boldsymbol{V}_s \cap \boldsymbol{V}_t = \emptyset, s, t \in \{1, \ldots, p, \ldots P\}, s \neq t$.
```

die due to energy exhaustion, resulting in network coverage failure [25]. Hence, the network lifetime can be expressed as

$$T_{net} \propto \max_{v_{\tau,k} \in \mathbb{V}_\tau} \varepsilon_{\tau,k}, \tag{4}$$

where $\varepsilon_{\tau,k}$ is the residual energy of sensor $v_{\tau,k}$.

To capture the probability of not having a failure within the time interval $(0, t_n)$, the sensor reliability is described by the sensor tolerance that is modeled as a Poisson Distribution [25], [30]. Hence, the network reliability is denoted as

$$R(\mathbb{V}) = \sum\nolimits_{n=1}^{N} R(v_n) = \sum\nolimits_{n=1}^{N} exp(-\theta t_n), \tag{5}$$

where the failure rate of the sensor is a constant, denoted by θ [31]. In typical routing protocols, the fault tolerance is viewed as an ability to maintain the network operation without any interruption [25]. When sensor v_n is encoded as a CH $v_{c,p}$ or CM $v_{m,q}$ based on Algorithm 1, the lifetime of sensor v_n is, respectively, approximated as

$$t_n = \frac{\varepsilon_n}{E_{bro}(l') + E_{tx}\left(l, d(v_{c,p}, v_0)\right) + \eta E_{rx}(l)}, \tag{6}$$

$$t_n = \frac{\varepsilon_n}{E_{bro}(l') + E_{tx}\left(l, d(v_{m,q}, v_{c,p})\right)}, \tag{7}$$

where $d(v_{c,p}, v_0)$ is the distance from CH $v_{c,p}$ to the BS v_0 and parameter η is the number of received data messages, when sensor v_n is encoded as CH $v_{c,p}$. $d(v_{m,q}, v_{c,p})$ is the distance from sensor $v_{m,q}$ to its CH $v_{c,p}$ when sensor v_n is encoded as CM $v_{m,q}$. In addition, $E_{bro}(l')$ represents the broadcast energy consumption and is given by

$$E_{bro}(l') = E_{tx}(l', R_{c,n}) + \gamma E_{rx}(l'), \tag{8}$$

where the sizes of the broadcast message and the data message are l' and l, respectively. Parameter γ is the number of received broadcast messages and is determined by the communication range $R_{c,n}$. Based on the definition of $R_{c,n}$ in [23], the communication range is given by

$$R_{c,n} = \left[1 - \frac{d_{\max} - d(v_n, v_0)}{\mu(d_{\max} - d_{\min})}\right] R_{\max}, \tag{9}$$

where $d_{\max}$ and $d_{\min}$ are, respectively, the maximum and minimum distances between sensors and the BS while $d(v_n, v_0)$ represents the distance between sensor v_n and the BS v_0. Parameter μ is a predefined constant that can be adjusted according to the environment. $R_{\max}$ is the maximum distance from the BS to the monitoring field.

In uRLLWSNs, each sensor is within the communication range of one or more sensors to form a connected network. Hence, maintaining the network connectivity is important for guaranteeing that the messages are indeed propagated to the BS. Given that the network connectivity is closely related to the network coverage [32], the network connectivity is expressed as

$$H(\mathbb{V}) = \sum\nolimits_{n=1}^{N} (h(v_n)), \tag{10a}$$

$$h(v_n) = 1 - exp\left(-\left(R_{c,n} - R_{s,n}\right)\right), \tag{10b}$$

where $R_{c,n} - R_{s,n} > 0$ must be satisfied to achieve the network connectivity, and $R_{s,n}$ is the sensing range of sensor v_n. We set $R_{c,n} = 2R_{s,n}$ to achieve satisfactory performance based on the analysis in [32].

In the monitoring field, each sensed object is observed by at least one sensor. Given that the sensed objects are deployed differently in diverse WSNs' applications, we adopt a general deployment scenario where massive sensed objects, denoted by $\mathbb{O} = \{o_1, \ldots, o_j, \ldots, o_J\}$, are distributed randomly. Adopting the simple binary sensor coverage model in [33], the network coverage is defined as [25]

$$C(\mathbb{O}) = \frac{1}{J}\sum\nolimits_{j=1}^{J} c(o_j), \tag{11a}$$

$$c(o_j) = \begin{cases} 1, & \text{if } \exists v_n \in \mathbb{V},\ d\left(v_n, o_j\right) \le R_{s,n}, \\ 0, & \text{otherwise}, \end{cases} \tag{11b}$$

where $d(v_n, o_j)$ is the Euclidean distance between sensor v_n and sensed object o_j.

For the transmission between sensor v_n and the BS v_0, the network latency is defined as the time elapsed between the departure of a sensed message from sensor v_n to the BS v_0 [25] as follows

$$\begin{aligned} D(v_n, v_0) &= \left(T_q + T_p + T_d\right) \times N(v_n, v_0) \\ &= c \times N(v_n, v_0) \\ &\propto N(v_n, v_0), \end{aligned} \tag{12}$$

where T_q, T_p, and T_d are the queue delay, the propagation delay, and the transmission delay, respectively. The sum of the various delays can be approximated as a constant, denoted by $c = \left(T_q + T_p + T_d\right)$.

For the WSN optimization problem, it is common to define a single network metric as the optimization objective and treat the other network metrics as optimization constraints, which artificially over-textitasizes the importance of one metric [25]. Hence, a fair optimization model is constructed to simultaneously satisfy multiple optimization objectives including the longest network lifetime and the highest network reliability and connectivity, which is formulated as

$$\max\ \{T_{net},\, R(\mathbb{V}),\, H(\mathbb{V})\}; \tag{13a}$$

$$s.t.\ \varepsilon_n \geq E_n, \quad \forall v_n \in \mathbb{V}; \tag{13b}$$

$$C(\mathbb{O}) = 1; \tag{13c}$$

$$D(v_i, v_0) \in \{1, 2\}. \tag{13d}$$

For this optimization model, Equation (13a) is an optimization objective function, Equation (13b) is the least residual energy limit for guaranteeing the sensor's regular functions, and Equation (13d) is the network latency condition. Most importantly, Equation (13c) is the network coverage limit.

C. OPTIMIZATION PROCEDURE

References [29] and [34] prove that it is NP-hard to learn an optimal routing scheduling in WSNs using classical gradient- or Hessian-based algorithms, especially for the multi-objective optimization model. However, the bio-mimetic heuristics-based strategy, for example genetic algorithms, has been widely used to solve NP-hard problems. The reason is that it is capable of obtaining an optimal solution to the optimization problem characterized by the non-differentiable nonlinear objective function [12], [25], [29], [35].

In the genetic-based algorithm optimization procedures, the fitness function, named the cost function in machine learning, is an evaluation function for chromosomes. The objective function in Equation (13a) is a preferred alternative for designing the fitness function due to its efficient assessment of chromosomes. However, the nonlinear dependencies between the multiple optimization objectives may result in ranking unreasonable importance levels of the optimization objectives, which is insufficient for learning an optimal network scheduling strategy. Fortunately, the PCA algorithm based on the singular value decomposition (SVD) is capable of eliminating dependencies [21]. For chromosome $\boldsymbol{s}_m$, denote the various optimization objectives' values of Equation (13a) by $\boldsymbol{x}_m = \begin{bmatrix} x_{m,1} & x_{m,2} & x_{m,3} \end{bmatrix}^T$. Hence, an evaluation dataset for population $\boldsymbol{S}$ is given by

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_m^T \\ \vdots \\ \boldsymbol{x}_M^T \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} x_{m,1} & x_{m,2} & x_{m,3} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} x_{M,1} & x_{M,2} & x_{M,3} \end{bmatrix} \end{bmatrix}, \tag{14}$$

where dataset $\boldsymbol{X}$ is an $M \times 3$ matrix of real-valued data with rank equal to 3 according to Equations (1) and (13a).

When using the PCA algorithm, it is necessary to ensure that samples of dataset $\boldsymbol{X}$ have zero means [21]. The sample means vector of dataset $\boldsymbol{X}$ is approximated by

$$\bar{\boldsymbol{x}} = \frac{1}{M}\left[\sum_i^M x_{i,1} \quad \sum_i^M x_{i,2} \quad \sum_i^M x_{i,3}\right]^T. \tag{15}$$

Hence, an improved dataset for dataset $\boldsymbol{X}$ is given by

$$\overline{\boldsymbol{X}} = \left[(\boldsymbol{x}_1 - \bar{\boldsymbol{x}}) \quad \cdots \quad (\boldsymbol{x}_m - \bar{\boldsymbol{x}}) \quad \cdots \quad (\boldsymbol{x}_M - \bar{\boldsymbol{x}})\right]^T. \tag{16}$$

Dataset $\overline{\boldsymbol{X}}$ is also an $M \times 3$ matrix of real-valued data with rank equal to 3 and is decomposed as $\overline{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. According to the SVD definition in [36], matrix $\boldsymbol{U}$ is $(M \times M)$ orthonormal matrix including the left-singular vectors, matrix $\boldsymbol{\Sigma}$ is $(M \times 3)$ diagonal matrix, and matrix $\boldsymbol{V}$ is (3×3) orthonormal matrix including the right-singular vectors. The m^{th} sample of dataset $\overline{\boldsymbol{X}}$ is projected to sample z_m via the linear transformation $\boldsymbol{V}$ as follows

$$z_m = (\boldsymbol{x}_m - \bar{\boldsymbol{x}}_{\boldsymbol{m}})^T \boldsymbol{V}, \tag{17}$$

resulting in a projection dataset denoted by $\boldsymbol{Z} = \overline{\boldsymbol{X}}\boldsymbol{V}$. Hence, the covariance variance of dataset $\boldsymbol{Z}$ is given by

$$\begin{aligned} \text{Cov}[\boldsymbol{Z}] &= \frac{1}{M-1}\boldsymbol{Z}^T\boldsymbol{Z} \\ &= \frac{1}{M-1}\left(\overline{\boldsymbol{X}}\boldsymbol{V}\right)^T\left(\overline{\boldsymbol{X}}\boldsymbol{V}\right) \\ &= \frac{1}{M-1}\boldsymbol{V}^T\overline{\boldsymbol{X}}^T\overline{\boldsymbol{X}}\boldsymbol{V} \\ &= \frac{1}{M-1}\boldsymbol{V}^T\left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right)^T\left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right)\boldsymbol{V} \\ &= \frac{1}{M-1}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{V} \\ &= \frac{1}{M-1}\boldsymbol{\Sigma}^2, \end{aligned} \tag{18}$$

where $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$ and $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$. Using the linear transformation $\boldsymbol{V}$, dataset $\overline{\boldsymbol{X}}$ is transformed into dataset $\boldsymbol{Z}$ with the diagonal covariance matrix denoted by $\boldsymbol{\Sigma}^2$. According to the PCA principle in [37], the diagonal elements of matrix $\boldsymbol{\Sigma}$ are eigenvalues of matrix $\boldsymbol{Z}$ and their respective ratios of $\sum_{j=1}^3 \Sigma_{j,j}$ are used to rank the importance levels of multiple variances in sample z_m. Hence, the fitness function can be formulated as

$$f_m = z_m^T\left(\frac{diag(\boldsymbol{\Sigma})}{\sum_{i=1}^3 \Sigma_{i,i}}\right) = \frac{1}{\sum_{i=1}^3 \Sigma_{i,i}}\left(\sum_{i=1}^3\left(z_{m,i}\Sigma_{i,i}\right)\right). \tag{19}$$

Therefore, a fitness vector is formed as follows

$$\boldsymbol{f} = \left[f_1 \quad \cdots \quad f_m \quad \cdots \quad f_M\right]^T. \tag{20}$$

Using the search mechanism of genetic algorithms, MLPGA learns the optimal chromosome to design a near-optimal clustering network topology. First, the selection process is accomplished by using the roulette wheel selection, in which each chromosome is given a probability of being copied into the next generation. For example, chromosome $\boldsymbol{s}_m$ is selected with the following likelihood

$$\varphi(\boldsymbol{s}_m) = \frac{f_m}{\sum_{i=1}^M f_i}. \tag{21}$$

Algorithm 2 Roulette Wheel Selection

```
Require: $f, \boldsymbol{S}$;
Ensure: The selection population $\boldsymbol{S}_c$
Generate $\boldsymbol{r} = \left[\, r_0 \; r_1 \; \cdots \; r_m \; \cdots \; r_M \,\right]$,
    where $r_0 = 0$, $r_m \in U(0, 1)$, $q_0$=0;
for $m = 1$ to $M$ do
    Compute probability mass function:
        $q_m = \sum_{k=1}^{m} \varphi(\boldsymbol{s}_k)$;
    for $j = 1$ to $M$ do
        if $q_{m-1} < r_j \leq q_m$ then
            Add the chromosome: $\boldsymbol{S}_c \cup \{\boldsymbol{s}_m\} \Rightarrow \boldsymbol{S}_c$;
        end if
    end for
end for
return $\boldsymbol{S}_c$
```

The roulette wheel selection process [38] is presented in Algorithm 2, while TABLE 2 presents an example about generating a population of the 10 chromosomes as follows

$$\boldsymbol{S}_c = \mathfrak{R}(\boldsymbol{S} \,|\, \boldsymbol{f}) = \{\boldsymbol{s}'_i | i = 1, 2, \ldots, 10\}. \tag{22}$$

In the population $\boldsymbol{S}_c$, each chromosome is randomly selected once for the crossover process. Two selected chromosomes, for example $\boldsymbol{s}'_i$ and $\boldsymbol{s}'_j$, generate two offsprings by exchanging their corresponding genes based on the crossover rate α [39], which is illustrated in Fig. 1 and denoted as

$$\mathfrak{C}(\boldsymbol{s}'_i, \boldsymbol{s}'_j \mid \alpha) \Rightarrow \{\boldsymbol{s}''_i,\ \boldsymbol{s}''_j\}. \tag{23}$$

Different from the crossover process, the mutation process involves altering the values at several randomly selected genes with a lower mutation rate β [39]. The crossover chromosomes, denoted by $\boldsymbol{s}''_i$ and $\boldsymbol{s}''_j$, evolve into the mutation chromosomes as follows

$$\mathfrak{M}(\boldsymbol{s}''_i \mid \beta) \Rightarrow \boldsymbol{s}'''_i, \quad \mathfrak{M}(\boldsymbol{s}''_j \mid \beta) \Rightarrow \boldsymbol{s}'''_j. \tag{24}$$

To identify the optimal chromosome, the diversity of the n^{th} gene for population $\boldsymbol{S}$ is defined as

$$d_n\left(\boldsymbol{S}\right) = \sum_{i=1}^{M} s_{i,n}. \tag{25}$$

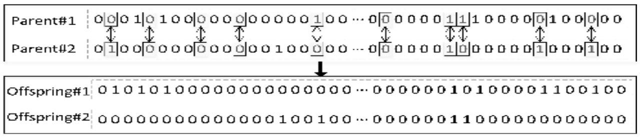

FIGURE 1. Illustration for the chromosome crossover.

If satisfying $d_n\left(\boldsymbol{S}\right) = M$, the n^{th} gene is certain and is added into a minimal schema $\Gamma\left(\boldsymbol{S}\right)$, which is expressed as

$$\Gamma\left(\boldsymbol{S}\right) \cup \{n\} \Rightarrow \Gamma\left(\boldsymbol{S}\right), \tag{26a}$$

$$\Gamma\left(\boldsymbol{S}\right) = \{n \mid d_n\left(S\right) = M\}. \tag{26b}$$

The minimal schema is developed according to the principles of unsupervised learning of machine learning that classifies the samples set into different groups by investigating their similarity [22]. To identify the optimal chromosome, the successive processes (given by selection, crossover, and mutation) continue and repeat until the convergence condition is satisfied, denoted by $\|\Gamma\left(\boldsymbol{S}\right)\| = C$, which is described in Algorithm 3.

III. PROOF OF MLPGA CONVERGENCE

In this section, MLPGA is proved to converge to the optimal chromosome under the above-described optimization model. Our proof is divided into the following two steps:

1) Based on the fitness values of chromosomes, the selection process converges to the locally optimal chromosome with the largest fitness value;
2) The crossover and mutation processes make it possible to access arbitrary chromosomes, which helps to converge to the globally optimal chromosome.

A. LOCAL CONVERGENCE FOR SELECTION

Theorem 1 (Convergence to the Locally Optimal Chromosome via the Selection Process): Denote the initial population by $\boldsymbol{S}_0 = [\, \boldsymbol{s}_1 \, \cdots \, \boldsymbol{s}_m \, \cdots \, \boldsymbol{s}_M \,]^T$ and its probability distribution by $\boldsymbol{P}_0 = [\, \varphi\left(\boldsymbol{s}_1\right) \cdots \varphi(\boldsymbol{s}_m) \cdots \varphi\left(\boldsymbol{s}_M\right)]^T$. After performing k selections on $\boldsymbol{S}_0$, the selection population converges to the population denoted by $\boldsymbol{S}' = \{\boldsymbol{s}_m | \boldsymbol{s}_m = \arg\max_{\boldsymbol{s}_k \in \boldsymbol{S}_0} f_k\}$, in which

TABLE 2. An example for selection population of 10 chromosomes.

No.	Sequence including 100 genes for chromosomes	Fitness value	Likelihood	q_m	r_m	Selected times
1	00010100000000000100· · ·00000011100000100000	0.102994	10.29%	0.1029	0.8530	1
2	00000000000000000001· · ·00000000100000000000	0.089853	8.98%	0.1928	0.6220	0
3	00100000010000000000· · ·00000000000001000000	0.102865	10.28%	0.2957	0.3509	2
4	01000100000000000100· · ·00000011100000100000	0.114033	11.40%	0.4097	0.5132	2
5	00000000000110000000· · ·01000000000000000010	0.094476	9.44%	0.5042	0.4018	0
6	01000000000000100000· · ·00000010000001000100	0.084337	8.43%	0.5885	0.0759	1
7	00000100000000000100· · ·00000011100000100000	0.123162	12.31%	0.7117	0.2399	2
8	00100000010000000000· · ·00000000000001000000	0.082700	8.27%	0.7944	0.6933	0
9	00000000000010010000· · ·00010000000000000000	0.115114	11.51%	0.9095	0.8639	2
10	00000000000000100000· · ·00000010000001000100	0.090461	9.04%	1.0000	0.2399	0

Algorithm 3 Machine-Learning-Based Parallel Genetic Algorithms for Multi-Objective Optimization (MLPGA)

```
Require: The model parameters, $\mathcal{G}(\mathbb{V}, \mathbb{E})$, $N$, $C$, $E_{elec}$, $\epsilon_{fs}$, $\epsilon_{mp}$, $l$, $l'$, $\alpha$, $\beta$, $\tau = 0$, $\widetilde{S} = \emptyset$, $\Gamma(\widetilde{S}) = \emptyset$;
Ensure: The minimal schema $\Gamma(\boldsymbol{S})$;
Generate a population of $M$ chromosomes: $\boldsymbol{S} = \begin{bmatrix} s_1 & \cdots & s_m & \cdots & s_M \end{bmatrix}^T$;
for $m = 1$ to $M$ do
    $\forall s_m \in \boldsymbol{S}$, construct the sample $(\boldsymbol{x}_m - \bar{\boldsymbol{x}})$, establish dataset $\overline{\boldsymbol{X}}$;
    Translate dataset $\overline{\boldsymbol{X}}$ into dataset $\boldsymbol{Z}$ using PCA, and compute the fitness vector $\boldsymbol{f}$;
end for
while $\tau < C$ do
    Generate a selection population $\boldsymbol{S}_c$ of $M$ chromosomes using Algorithm 2:
        $\boldsymbol{S}_c = \mathfrak{R}(\boldsymbol{S}|\boldsymbol{f}) = \{\boldsymbol{s}'_i | i = 1, 2, \ldots, M\}$;
    for $m = 1$ to $(\frac{M}{2})$ do
        Randomly select chromosomes $\boldsymbol{s}'_i$ and $\boldsymbol{s}'_j$ from $\boldsymbol{S}_c$ once to perform crossover with the crossover rate $\alpha$:
            $\mathfrak{C}(\boldsymbol{s}'_i, \boldsymbol{s}'_j \mid \alpha) \Rightarrow \boldsymbol{s}''_i,\ \boldsymbol{s}''_j$;
        Perform mutation on $\boldsymbol{s}''_i$ and $\boldsymbol{s}''_j$ with the mutation rate $\beta$:
            $\mathfrak{M}(\boldsymbol{s}''_i \mid \beta) \Rightarrow \boldsymbol{s}'''_i, \quad \mathfrak{M}(\boldsymbol{s}''_j \mid \beta) \Rightarrow \boldsymbol{s}'''_j$;
        Add $\boldsymbol{s}'''_i$ and $\boldsymbol{s}'''_j$ to the new population $\widetilde{\boldsymbol{S}}$: $\widetilde{\boldsymbol{S}} \cup \{\boldsymbol{s}'''_i, \boldsymbol{s}'''_j\} \Rightarrow \widetilde{\boldsymbol{S}}$;
    end for
    for $n = 1$ to $N$ do
        Evaluate the diversity of the $n^{th}$ gene of $\widetilde{\boldsymbol{S}}$ according to (25) and calculate $d_n(\widetilde{\boldsymbol{S}})$;
        if $d_n(\widetilde{\boldsymbol{S}}) == M$ then
            $\Gamma(\widetilde{\boldsymbol{S}}) \cup \{n\} \Rightarrow \Gamma(\widetilde{\boldsymbol{S}})$ and $\Gamma\left(\widetilde{\boldsymbol{S}}\right) = \left\{n \mid d_n(\widetilde{\boldsymbol{S}}) = M\right\}$;
        end if
    end for
    Evaluate the size of the minimal schema $\Gamma(\widetilde{\boldsymbol{S}})$: $\tau = \|\Gamma(\widetilde{\boldsymbol{S}})\|$;
    Prepare for the next iteration by assigning the population, $\boldsymbol{S} = \widetilde{\boldsymbol{S}}$;
end while
return $\Gamma(\boldsymbol{S})$;
```

the fitness values of chromosomes are much larger than those of chromosomes outside selection population $\boldsymbol{S}'$. The probability of local convergence is given by

$$\widetilde{\varphi}(s_m) = \lim_{k\to\infty} P\{\mathfrak{R}^{(k)}\left(\boldsymbol{S}_0|\boldsymbol{f}_0\right) = s_m|\boldsymbol{P}_0\} = \begin{cases} \dfrac{\varphi(s_m)}{\sum\limits_{s_i\in\boldsymbol{S}_0}\varphi\left(s_i\right)}, & s_m \in \boldsymbol{S}' \\ 0, & s_m \notin \boldsymbol{S}', \end{cases} \tag{27}$$

and $\exists\Delta > 0$ that is subject to

$$\sum_{s_m\in\boldsymbol{S}'}\left|P\{\mathfrak{R}^{(k)}(\boldsymbol{S}_0|\boldsymbol{f}_0) = s_m|\boldsymbol{P}_0\} - \widetilde{\varphi}(s_m)\right| \le (\lambda^{-k} * \Delta), \tag{28}$$

where $\lambda = \min\{\frac{\max f_0}{f(s_m)}|\varphi\left(s_m\right) > 0, s_m \notin \boldsymbol{S}_0\}$.

Proof: Define the basic functions $\delta_{ij} = \begin{cases} 1, i = j \\ 0, i \ne j \end{cases}$ and $F_{ij} = \delta_{ij}f_if_j$, then we obtain a matrix $\boldsymbol{F}_0 = \left(F_{ij}\right)_{(M\times M)}$.

When performing the first selection procedure based on the fitness values, the transform is given by

$$\widehat{\boldsymbol{F}}\left(\boldsymbol{P}_0\right) = \left(\boldsymbol{F}_0\boldsymbol{P}_0\right)\Big/\left(\boldsymbol{1}^T\left(\boldsymbol{F}_0\boldsymbol{P}_0\right)\right), \tag{29}$$

where $\forall s_m \in \boldsymbol{S}_0$ satisfies the following sub-transform

$$\widehat{\boldsymbol{F}}\left(\boldsymbol{P}_0\right)_m = \left(f_m * \varphi\left(s_m\right)\right)\Big/\sum_{s_k\in\boldsymbol{S}_0}\left(f_k\varphi\left(s_k\right)\right). \tag{30}$$

Assume that the selection process has been performed for k times, then the transform can be described as follows

$$\begin{aligned} \widehat{\boldsymbol{F}}^{(k)}\left(\boldsymbol{P}_0\right)_m &= P\{\mathfrak{R}^{(k)}\left(\boldsymbol{S}_0|\boldsymbol{f}_0\right) = s_m^{(k)}|\boldsymbol{P}_0\} \\ &= P\{\mathfrak{R}\left(\boldsymbol{S}_{n-1}|\boldsymbol{f}_{n-1}\right) = s_m^{(k)}|\widehat{\boldsymbol{F}}^{(n-1)}\left(\boldsymbol{P}_0\right)\} \\ &= \widehat{\boldsymbol{F}}(\widehat{\boldsymbol{F}}^{(n-1)}(\boldsymbol{P}_0))_m. \end{aligned} \tag{31}$$

Therefore, the transform for $(k+1)$ selections is given by

$$\begin{aligned} &\widehat{\boldsymbol{F}}^{(k+1)}\left(\boldsymbol{P}_0\right)_m \\ &= \widehat{\boldsymbol{F}}(\widehat{\boldsymbol{F}}^{(k)}\left(\boldsymbol{P}_0\right))_m \\ &= \left(f_m\frac{f_m^k * \varphi(s_m)}{\sum\limits_{s_i\in\boldsymbol{S}_0}(f_i^k * \varphi(s_i))}\right)\Bigg/\left(\sum_{s_j\in\boldsymbol{S}_0}(f_j^k\frac{f_j^k * \varphi(s_j)}{\sum\limits_{s_i\in\boldsymbol{S}_0}(f_i^k * \varphi(s_i))})\right) \\ &= \left(f_m^{k+1} * \varphi(s_m)\right)\Bigg/\left(\sum_{s_j\in\boldsymbol{S}_0}\left(f_m^{k+1} * \varphi(s_j)\right)\right). \end{aligned} \tag{32}$$

Hence, by using mathematical induction, the transform for the selection procedure satisfies the following relation

$$\widehat{\boldsymbol{F}}^{(k)}\left(\boldsymbol{P}_0\right)_m = \left(f_m^k * \varphi(\boldsymbol{s}_m)\right) \Big/ \sum_{\boldsymbol{s}_i \in \boldsymbol{S}_0} \left(f_i^k * \varphi\left(\boldsymbol{s}_i\right)\right). \tag{33}$$

We obtain the transition probability for k selections as follows

$$\begin{aligned}
&\lim_{k\to\infty} P\{\mathfrak{R}^{(k)}\left(\boldsymbol{S}_0|\boldsymbol{f}_0\right) = \boldsymbol{s}_m|\boldsymbol{P}_0\} \\
&\quad = \lim_{k\to\infty}\left(f_m^k * \varphi\left(\boldsymbol{s}_m\right) \Big/ \big(\sum_{\boldsymbol{s}_i\in\boldsymbol{S}_0} (f_i^k * \varphi(\boldsymbol{s}_i))\big)\right) \\
&\quad = \lim_{k\to\infty}\left(\varphi(\boldsymbol{s}_m) \Bigg/ \sum_{\boldsymbol{s}_i\in\boldsymbol{S}_0}\left(\frac{f_i^k}{f_m^k} * \varphi(\boldsymbol{s}_i)\right)\right) \\
&\quad = \varphi\left(\boldsymbol{s}_m\right) \Bigg/ \left(\sum_{\boldsymbol{s}_i\in\boldsymbol{S}_0}\left(\lim_{k\to\infty}\left(\frac{f_i^k}{f_m^k} * \varphi(\boldsymbol{s}_i)\right)\right)\right).
\end{aligned} \tag{34}$$

When satisfying $\boldsymbol{s}_m \in \boldsymbol{S}'$, $\lim\limits_{k\to\infty}(\frac{f(\boldsymbol{s}_i)}{f(\boldsymbol{s}_m)})^k = 0$ if and only if $\boldsymbol{s}_i \notin \boldsymbol{S}'$. When satisfying $\boldsymbol{s}_m \notin \boldsymbol{S}'$, $\lim\limits_{k\to\infty}(\frac{f(\boldsymbol{s}_i)}{f(\boldsymbol{s}_m)})^k = \infty$ if and only if $f\left(\boldsymbol{s}_i\right) > f\left(\boldsymbol{s}_m\right)$. Hence, Equation (27) is proved. Next, by using mathematical deduction, Equation (28) is proved as follows

$$\begin{aligned}
&\sum_{\boldsymbol{s}_m\in\boldsymbol{S}'}\left|P\{\mathfrak{R}^{(k)}\left(\boldsymbol{S}_0|\boldsymbol{f}_0\right) = \boldsymbol{s}_m|\boldsymbol{P}_0\} - \widetilde{\varphi}\left(\boldsymbol{s}_m\right)\right| \\
&\quad = \sum_{\boldsymbol{s}_m\in\boldsymbol{S}'}\left|\frac{\left(\max\boldsymbol{f}_0\right)^k * \varphi\left(\boldsymbol{s}_m\right)}{\sum\limits_{\boldsymbol{s}_i\in\boldsymbol{S}_0}(f_i^k * \varphi\left(\boldsymbol{s}_i\right))} - \frac{\varphi\left(\boldsymbol{s}_m\right)}{\sum\limits_{\boldsymbol{s}_i\in\boldsymbol{S}_0}\varphi\left(\boldsymbol{s}_i\right)}\right| \\
&\quad = \sum_{\boldsymbol{s}_m\in\boldsymbol{S}'}\left|\frac{\varphi\left(\boldsymbol{s}_m\right)}{\sum\limits_{\boldsymbol{s}_i\in\boldsymbol{S}_0}\left((\frac{f_i}{\max\boldsymbol{f}_0})^k * \varphi\left(\boldsymbol{s}_i\right)\right)} - \frac{\varphi\left(\boldsymbol{s}_m\right)}{\sum\limits_{\boldsymbol{s}_i\in\boldsymbol{S}_0}\varphi\left(\boldsymbol{s}_i\right)}\right| \\
&\quad \le \sum_{\boldsymbol{s}_m\in\boldsymbol{S}'}\left(\left|\frac{\sum\limits_{\boldsymbol{s}_i\notin\boldsymbol{S}_0}\left(\left(\frac{f_i}{\max\boldsymbol{f}_0}\right)^k * \varphi\left(\boldsymbol{s}_i\right)\right)}{\sum\limits_{\boldsymbol{s}_i\in\boldsymbol{S}_0}\varphi\left(\boldsymbol{s}_i\right) + \sum\limits_{\boldsymbol{s}_i\notin\boldsymbol{S}_0}\left(\left(\frac{f_i}{\max\boldsymbol{f}_0}\right)^k * \varphi\left(\boldsymbol{s}_i\right)\right)}\right| \right. \\
&\qquad\qquad \left. \frac{\varphi\left(\boldsymbol{s}_m\right)}{\sum\limits_{\boldsymbol{s}_i\in\boldsymbol{S}_0}\varphi\left(\boldsymbol{s}_i\right)}\right) \\
&\quad \le \sum_{\boldsymbol{s}_m\in\boldsymbol{S}'}\left(\frac{\varphi\left(\boldsymbol{s}_m\right)}{\sum\limits_{\boldsymbol{s}_i\in\boldsymbol{S}_0}\varphi\left(\boldsymbol{s}_i\right)}\left|\frac{\lambda^{-k} * \sum\limits_{\boldsymbol{s}_i\notin\boldsymbol{S}_0}\varphi\left(\boldsymbol{s}_i\right)}{\sum\limits_{\boldsymbol{s}_i\in\boldsymbol{S}_0}\varphi\left(\boldsymbol{s}_i\right)}\right|\right) \\
&\quad \le \lambda^{-k} * \left(\frac{\sum\limits_{\boldsymbol{s}_i\notin\boldsymbol{S}_0}\varphi\left(\boldsymbol{s}_i\right)}{\sum\limits_{\boldsymbol{s}_i\in\boldsymbol{S}_0}\varphi\left(\boldsymbol{s}_i\right)}\right).
\end{aligned} \tag{35}$$

Therefore, the convergence to the locally optimal chromosome based on the selection procedure is proved.

B. GLOBAL CONVERGENCE FOR CROSSOVER AND MUTATION

Theorem 2 (Convergence to the Globally Optimal Chromosome via Crossover and Mutation): Assume that the whole chromosome space and its probability distribution are, respectively, denoted by $\mathbb{S} = \{0, 1\}^N = \left\{\boldsymbol{s}_m | 1 \le m \le 2^N, m \in N^+\right\}$ and $\boldsymbol{P} = \{\varphi\left(\boldsymbol{s}_m\right) | \boldsymbol{s}_m \in \mathbb{S}\}$. After k iterations of crossover and mutation, convergence to the global optimal chromosome $\boldsymbol{s}_*$ is achieved with the following probability

$$\begin{aligned}
&P\{\mathfrak{T}^{(k)}(\boldsymbol{S}'|\zeta) = \boldsymbol{s}_*|\boldsymbol{P}\} \\
&\quad = \sum_{\boldsymbol{s}_t\in\mathbb{S}}[\varphi(\boldsymbol{s}_t)\prod_{n=1}^{N}(0.5 + (\delta_{s_{*,n}s_{t,n}} - 0.5)(1-2\zeta)^k)],
\end{aligned} \tag{36a}$$

$$P\{\mathfrak{T}^{(k)}(\boldsymbol{S}'|\zeta) = \boldsymbol{s}_*|\boldsymbol{P}\} > 0, \tag{36b}$$

where the operator $\mathfrak{T}(\boldsymbol{S}'|\zeta) = (\mathfrak{C}(\boldsymbol{S}'|\alpha)\odot\mathfrak{M}(\boldsymbol{S}'|\beta))$ is the composed operation of crossover and mutation and its transition probability is denoted by $\zeta = \alpha\beta$.

Proof: The mathematical induction method is adopted to prove the global convergence for crossover and mutation.

The n^{th} gene $s_{i,n}$ of $\boldsymbol{s}_i$ is obtained using one iteration of crossover and mutation with the probability

$$\begin{aligned}
P\{\mathfrak{T}(\boldsymbol{S}'|\zeta) = s_{i,n}\} &= \begin{cases} 1-\zeta, & s_{t,n} = s_{i,n} \\ \zeta, & s_{t,n} \ne s_{i,n} \end{cases} \\
&= \zeta + \delta_{s_{t,n}s_{i,n}}\left(1-2\zeta\right),
\end{aligned} \tag{37}$$

where $\boldsymbol{s}_t \in \boldsymbol{S}'$. Then, chromosome $\boldsymbol{s}_i$ is obtained with the following probability

$$\begin{aligned}
&P\{\mathfrak{T}(\boldsymbol{S}'|\zeta) = \boldsymbol{s}_i|\boldsymbol{P}\} \\
&\quad = \sum_{\boldsymbol{s}_t\in\boldsymbol{S}'}(\varphi(\boldsymbol{s}_t)\prod_{n=1}^{N}P\{\mathfrak{T}(\boldsymbol{S}'|\zeta) = s_{i,n}\}) \\
&\quad = \sum_{\boldsymbol{s}_t\in\boldsymbol{S}'}(\varphi\left(\boldsymbol{s}_t\right)\prod_{n=1}^{N}(\zeta + \delta_{s_{t,n}s_{i,n}}\left(1-2\zeta\right))) \\
&\quad = \sum_{\boldsymbol{s}_t\in\boldsymbol{S}'}(\varphi\left(\boldsymbol{s}_t\right)\prod_{n=1}^{N}(0.5 + (\delta_{s_{t,n}s_{i,n}} - 0.5)\left(1-2\zeta\right))).
\end{aligned} \tag{38}$$

Hence, Equation (36a) is satisfied when $k = 1$. Assume that Equation (36a) is true for $k (k > 1)$ iterations of crossover and mutation and its corresponding probability is expressed as

$$P_k = P\{\mathfrak{T}^{(k)}(\boldsymbol{S}'|\zeta) = \boldsymbol{s}_i|\boldsymbol{P}\}. \tag{39}$$

Then, for $(k+1)$ iterations, we obtain

$$\begin{aligned}
&P\{\mathfrak{T}^{(k+1)}(\boldsymbol{S}'|\zeta) = \boldsymbol{s}_*|\boldsymbol{P}\} \\
&\quad = \sum_{\boldsymbol{s}_t\in\boldsymbol{S}^k}\left[P\{\mathfrak{T}(\boldsymbol{S}^k|\zeta) = \boldsymbol{s}_*\} * P_k\right] \\
&\quad = \sum_{\boldsymbol{s}_t\in\boldsymbol{S}^k}(\varphi(\boldsymbol{s}_t)\prod_{n=1}^{N}(\zeta + \delta_{s_{t,n}s_{*,n}}(1-2\zeta)) * P_k),
\end{aligned} \tag{40}$$

$$\begin{aligned}&\sum\nolimits_{s_{t,n}} \left[(\zeta + \delta_{s_{t,n}s_{*,n}}(1-2\zeta)) \ *(0.5 + (\delta_{s_{t,n}s_{i,n}} - 0.5)(1-2\zeta)^k)\right]\\
&\quad= \delta_{s_{i,n}s_{*,n}} \left[(1-\zeta)(0.5+0.5(1-2\zeta)^k) \ +\zeta(0.5-0.5(1-2\zeta)^k)\right]\\
&\qquad+ (1-\delta_{s_{i,n}s_{*,n}})\left[(1-\zeta)(0.5-0.5(1-2\zeta)^k) \ +\zeta(0.5+0.5(1-2\zeta)^k)\right]\\
&\quad= \delta_{s_{i,n}s_{*,n}} * (0.5+0.5(1-2\zeta)^{k+1}) + (1-\delta_{s_{i,n}s_{*,n}}) * (0.5-0.5(1-2\zeta)^{k+1})\\
&\quad= 0.5 + (\delta_{s_{i,n}s_{*,n}} - 0.5)(1-2\zeta)^{k+1} > 0. \qquad (41)\end{aligned}$$

where $\boldsymbol{S}^k$ is the selection population that is generated after k iterations of crossover and mutation. We analyze $s_{t,n}, s_{i,n}, s_{*,n}$ under the two different conditions below

1) When satisfying $s_{i,n} = s_{*,n}$, there are two conditions, denoted by $s_{i,n} = s_{t,n}$ and $s_{i,n} \neq s_{t,n}$;
2) When satisfying $s_{i,n} \neq s_{*,n}$, there are two conditions, denoted by $s_{*,n} = s_{t,n}$ and $s_{*,n} \neq s_{t,n}, s_{i,n} = s_{t,n}$.

Then, Equation (41) is obtained.

Therefore, according to the above reasoning, Equation (36a) and Equation (36b) are proved, which, in turn, proves the convergence of the globally optimal chromosome via the crossover and mutation processes.

IV. PERFORMANCE EVALUATION

In this section, simulations are performed to demonstrate the viability of the proposed algorithm compared to state-of-the-art algorithms. We analyze many important network metrics including the network lifetime, the number of alive sensors, the energy consumption, and the network connectivity and reliability. Considering the stringent resource-limited uRLLWSNs environment, the computational complexity is also analyzed.

A. EXPERIMENT SETTING

In our experiments, the monitoring field is set as a square and the BS is located outside the monitoring field. Fig. 2 shows an application instance with a random deployment of 100 sensors and 64 sensed objects, where the symbols "○" and "★" represent the sensor and the sensed object, respectively. The network parameters are listed in TABLE 3 [13], [25], [40]. In addition, to achieve a high confidence level, we perform 100 trials per simulation instance and plot the figures by averaging simulation results.

TABLE 3. The network parameters.

Properties	Values
Monitoring field's width	$100m$
Coordinate of the BS	$(50m, 175m)$
Proportion of CHs	10%
Initial sensor energy	$0.05J$
(α, β, θ)	$(0.8, 0.01, 0.4 \times 10^{-5}H^{-1})$
(l', l)	(25bytes, 64bytes)
E_{elec}	50nJ/bit
$(\epsilon_{fs}, \epsilon_{mp})$	(10pJ/bit/m^2, 0.0013pJ/bit/m^4)

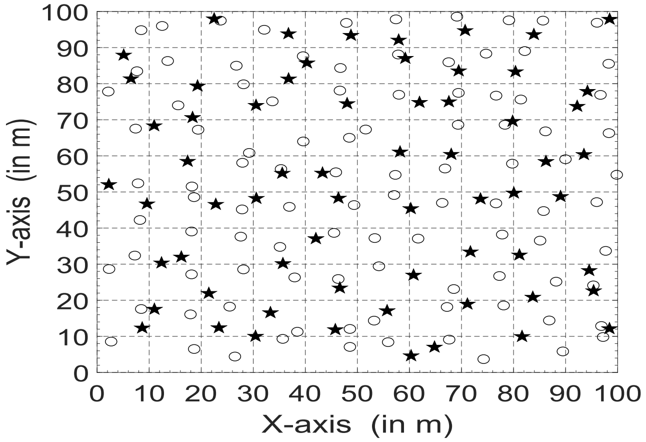

FIGURE 2. A deployment instance with 100 sensors and 64 sensed objects.

MLPGA is similar to HEED and HHCA in using the clustering strategy, and is an improvement of ULGAT in using the machine learning and genetic algorithms techniques. Hence, the network performance of MLPGA is compared to those of HEED, HHCA, and ULGAT. To verify the effect of the PCA algorithm, MLPAG without PCA (denoted henceforth by MLGANP) is also compared to MLPGA. The fitness function of MLGANP is defined as follows

$$f_m = \frac{1}{3}\left|\boldsymbol{x}_m - \bar{\boldsymbol{x}}\right| = \frac{1}{3}\sum_{i=1}^{3}\left|x_{m,i} - \bar{x}_i\right|. \qquad (42)$$

B. NUMBER OF ALIVE SENSORS AND NETWORK LIFETIME

In this subsection, the number of alive sensors and the network lifetime are evaluated when uRLLWSNs cannot fulfill its mission due to network coverage failure. These two performance metrics are affected by multiple critical parameters including the number of sensors N, the communication range parameter μ, and the number of sensed objects J. Parameter N corresponds to the network scale, while parameters μ and J have an effect on the amount of sensed messages and energy consumption. More explicitly, Fig. 3 and Fig. 6 show the number of alive sensors and the network lifetime versus the number of sensors N, respectively. It can be seen that MLPGA can fulfill the network mission for a smaller number of alive sensors than those of the other algorithms and outperforms those algorithms in prolonging the network

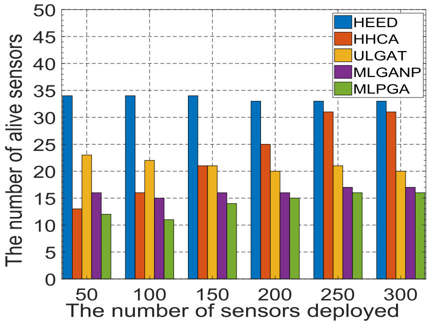

FIGURE 3. The number of alive sensors versus the number of sensors deployed.

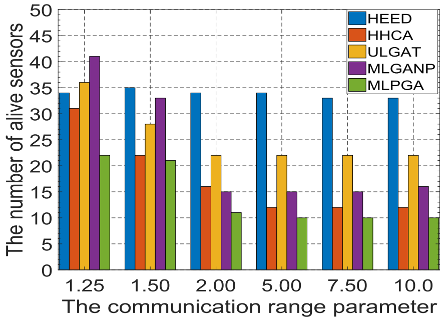

FIGURE 4. The number of alive sensors versus communication range parameter.

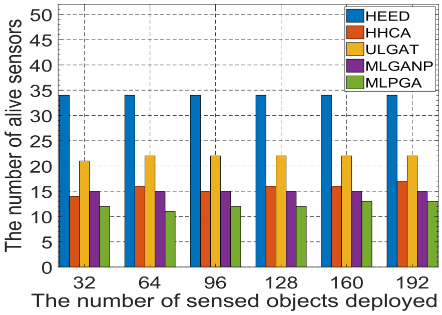

FIGURE 5. The number of alive sensors versus the number of objects deployed.

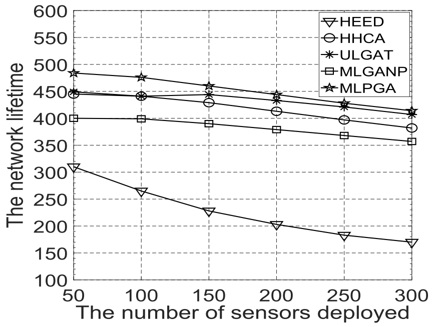

FIGURE 6. The network lifetime versus the number of sensors deployed.

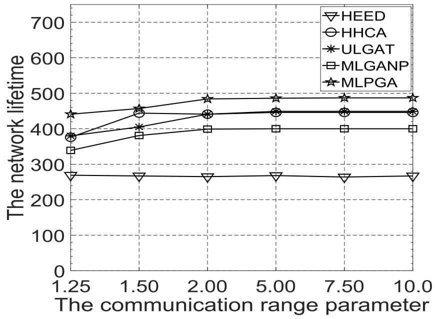

FIGURE 7. The network lifetime versus the communication range parameter.

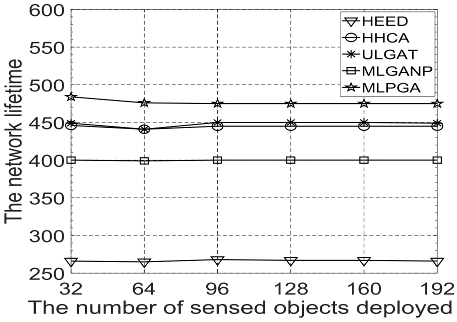

FIGURE 8. The network lifetime versus the number of sensed objects deployed.

lifetime. Fig. 4 and Fig. 7 present the number of alive sensors and the network lifetime versus the communication range parameter μ, respectively. MLPGA also demonstrates better performance in terms of the number of alive sensors and the network lifetime. Fig. 5 and Fig. 8 show the number of alive sensors and the network lifetime versus the number of sensed objects J, respectively. It can be seen that the number of sensed objectives J has a slight impact on both the number of alive sensors and the network lifetime. The reason is that the network coverage is affected by the deployed sensors and their sensing range, not the number of sensed objects. Considering 100 trials' simulations, it is necessary to analyze the standard deviation of one or more network metrics. The standard deviation of the network lifetime is shown in TABLE 4. The standard deviations of MLGANP and MLPGA are smaller than those of the other algorithms, implying that MLPGA and MLGANP are more suitable for the arbitrary deployment scenario.

Different from HEED, HHCA, and ULGAT, the proposed algorithms, called MLPGA and MLGANP, design a near-optimal clustering network topology, in which a clustering method of energy conversion is developed to prevent overloaded CHs. The transmitting energy consumption of each CH is transformed into virtual CMs to participate in the network clustering process with real CMs, which balances the sensors' communication load and avoids premature death of sensors due to energy exhaustion. In addition, the PCA algorithm enhances assessment of the network topology by eliminating dependencies between optimization objectives and ranking their importance levels. Hence, the MLPGA algorithm not only fulfills the network mission with a smaller number of alive sensors, but also prolongs the network lifetime.

TABLE 4. Standard deviation of the network lifetime for 100 trials' simulations.

Property classifications	Parameter values	HEED	HHCA	ULGAT	MLGANP	MLPGA
Various numbers of sensors deployed	50	35	11	14	14	6
	100	36	9	17	11	5
	150	41	8	15	13	8
	200	45	13	13	17	10
	250	49	18	14	21	13
	300	51	23	16	25	16
Various values of the parameter μ	1.25	24	15	6	18	4
	1.50	24	19	5	15	5
	2.00	36	20	17	11	5
	5.00	17	20	18	10	4
	7.50	18	20	18	10	5
	10.0	17	20	18	10	5
Various numbers of sensed objects	32	38	9	17	12	4
	64	36	10	17	11	5
	96	37	11	16	13	4
	128	37	11	16	12	5
	160	38	9	15	12	4
	192	38	12	15	12	4

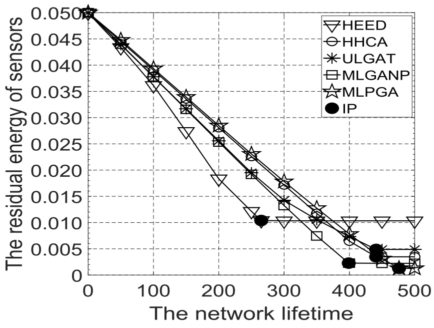

FIGURE 9. **The network lifetime versus the residual energy of sensors.**

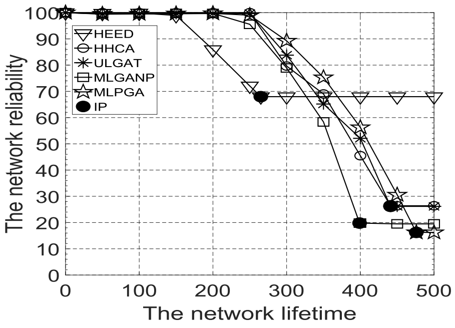

FIGURE 11. **The network lifetime versus the network reliability.**

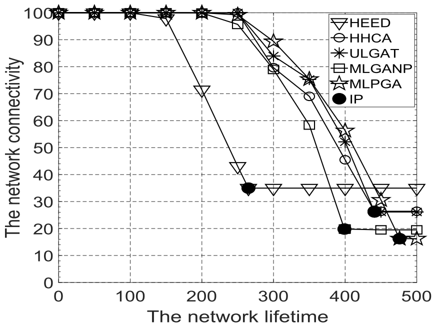

FIGURE 10. **The network lifetime versus the network connectivity.**

C. ENERGY CONSUMPTION, NETWORK CONNECTIVITY AND RELIABILITY

In addition to the network lifetime, it is important to analyze the other network metrics such as the energy consumption, the network connectivity and reliability. Assuming the following typical values for the key parameters $N = 100$, $\mu = 2$, and $J = 64$, the network performance versus the transmission round are, respectively, shown in Fig. 9, Fig. 10, and Fig. 11, where "•" is the inflection point (IP) whose abscissa represents the network termination. The horizontal line after IP indicates that the network performance remains unchanged, because the network does not function properly. Fig. 9 implies that MLPGA achieves better energy efficiency due to its higher average residual energy. From Fig. 10 and Fig. 11, it can be seen that MLPGA outperforms the other algorithms on the network connectivity and reliability.

The MLPGA algorithm formulates the uRLLWSNs problem into a fair multi-objective optimization model by defining the network lifetime and the network connectivity and reliability as optimization objectives, where the energy efficiency is the foundation of these optimization objectives. In this model, the PCA algorithm and the genetic algorithm are used to identify a near-optimal clustering network topology. Hence, MLPGA improves simultaneously the network lifetime and the network connectivity and reliability compared

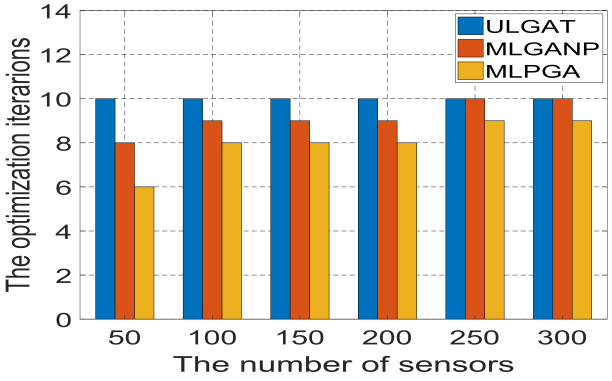

FIGURE 12. Iterations for various numbers of sensors.

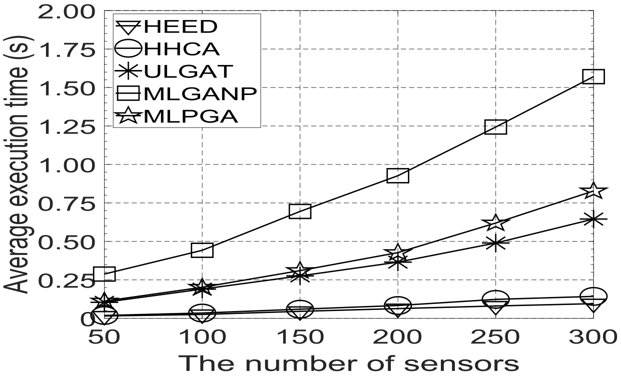

FIGURE 14. The execution time for various numbers of sensors.

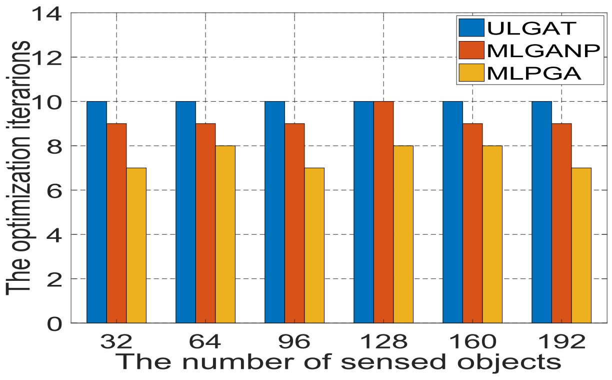

FIGURE 13. Iterations for various numbers of objects.

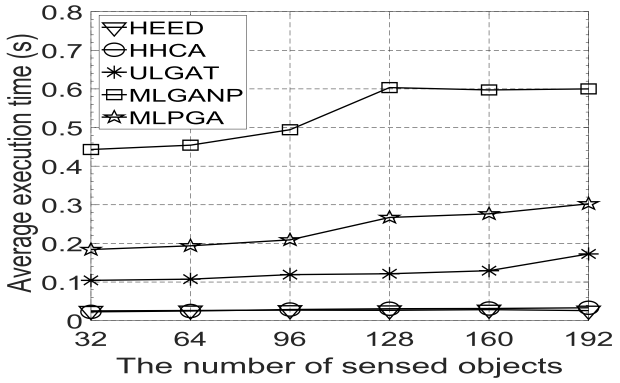

FIGURE 15. The execution time for various numbers of $\mathbb{O}$.

to HEED, HHCA, ULGAT, and MLGANP.

D. DISCUSSION ON COMPUTATIONAL COMPLEXITY

The bio-mimetic algorithms including ULGAT, MLGANP, and MLPGA involve optimization iterations due to using genetic algorithms. For various numbers of sensors and sensed objects with random deployments, optimization iterations per transmission round are shown in Fig. 12 and Fig. 13. It can be seen that MLPGA always performs a smaller number of optimization iterations than ULGAT and MLGANP and the PCA algorithm contributes to reducing the number of optimization iterations. It is necessary to further quantify the complexity with MATLAB 2017b on a Core I7-CPU@2.80GHz computer. Fig. 14 and Fig. 15 present the average execution time per transmission round for various numbers of sensors and sensed objects, respectively. These figures demonstrate that the execution time almost linearly increases with an increasing number of sensors except for HHCA and HEED. However, the number of sensed objects has a slight effect on the execution time, which implies that the number of sensed objects is not the key factor in the execution time. The reason is that the sensed objects' monitoring is related to the network coverage while not affecting the process of identifying a near-optimal clustering network topology. HEED and HHCA are, respectively, the distributed and semi-distributed algorithms, whose clustering operations for all sensors can be performed in parallel. Hence, the execution times of HEED and HHCA are lower compared to those of MLPGA, MLGANP, and ULGAT. Different from HEED and HHCA, the other algorithms need more execution time due to both their centralized scheme and using the genetic algorithm. The execution time of MLPGA is slightly larger than that of ULGAT due to processing of the multi-objective optimization model, while more optimization iterations for MLGANP cost more execution time than MLPGA. However, the computational complexity of MLPGA is still acceptable for monitoring applications such as volcano monitoring, because it is less than 1s for small to medium scale uRLLWSNs.

V. CONCLUSION

In this paper, we utilize machine learning techniques and genetic algorithms to develop the MLPGA algorithm, which identifies the optimal chromosome to design a near-optimum clustering network topology. This network topology provides an efficient communication architecture for ultra-reliable and low-latency wireless sensor networks, to simultaneously satisfy the multiple network objectives including a longer network lifetime and a higher network connectivity and reliability. More explicitly, the proposed algorithm utilizes the popular K-means clustering algorithm of machine learning to design a 2-tier network topology that is modeled into a chromosome, and develops a clustering method of energy conversion to prevent overloaded CHs. To identify the optimal chromosome, the proposed algorithm constructs a multi-objective optimization model according to the critical network metrics and performs this optimization using the genetic algorithm. In this model, the minimal schema of the population is defined as the convergence condition, and the principal component analysis algorithm is adopted to transform the multi-objective function of the optimization model into the fitness function by eliminating dependencies between the multiple optimization objectives and ranking importance levels of various optimization objectives. The proposed algorithm is proved to converge to the optimal chromosome both locally and globally. Simulation results demonstrate that the principal component analysis algorithm improves the network performance and the proposed algorithm is superior to state-of-the-art algorithms including HEED, HHCA, ULGAT, and MLGANP at a comparable complexity.

Interesting future research topics include improving the proposed MLPGA by investigating other performance metrics such as bit error rate and investigating multi-hop routing scheduling under uRLLWSN's stringent requirements.

ACKNOWLEDGMENT

The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] International Telecommunication Union-Recommendations, Standard ITU-R M.2083-0, Sep. 2015, *IMT Vision C Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond*.

[2] W. Guan, X. Wen, L. Wang, Z. Lu, and Y. Shen, "A service-oriented deployment policy of end-to-end network slicing based on complex network theory," *IEEE Access*, vol. 6, pp. 19691–19701, Apr. 2018.

[3] K. Antonakoglou, X. Xu, E. Steinbach, T. Mahmoodi, and M. Dohler, "Toward haptic communications over the 5G tactile Internet," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3034–3059, 4th Quart., 2018.

[4] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

[5] S. M. N. Alam and Z. J. Haas, "Coverage and connectivity in three-dimensional networks with random node deployment," *Ad Hoc Netw.*, vol. 34, pp. 157–169, Nov. 2015.

[6] M. Abdelhakim, Y. Liang, and T. Li, "Mobile coordinated wireless sensor network: An energy efficient scheme for real-time transmissions," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1663–1675, May 2016.

[7] J. Rodriguez, *Fundamentals of 5G Mobile Networks*. Hoboken, NJ, USA: Wiley, 2015.

[8] A. E. Shafie and N. Al-Dhahir, "A secure multiple-access scheme for rechargeable wireless sensors in the presence of an eavesdropper," *IEEE Commun. Lett.*, vol. 19, no. 6, pp. 945–948, Jun. 2015.

[9] G. Han, J. Jiang, N. Bao, L. Wan, and M. Guizani, "Routing protocols for underwater wireless sensor networks," *IEEE Commun. Mag.*, vol. 53, no. 11, pp. 72–78, Nov. 2015.

[10] H. Peng, S. Si, M. K. Awad, N. Zhang, H. Zhao, and X. S. Shen, "Toward energy-efficient and robust large-scale WSNs: A scale-free network approach," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 4035–4047, Dec. 2016.

[11] 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. (HICSS)*, Jan. 2000, p. 10.

[12] X. Yuan, M. Elhoseny, H. K. El-Minir, and A. M. Riad, "A genetic algorithm-based, dynamic clustering method towards improved WSN longevity," *J. Netw. Syst. Manage.*, vol. 25, no. 1, pp. 21–46, Jan. 2017.

[13] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 366–379, Oct. 2004.

[14] Y. Chang, H. Tang, Y. Cheng, Q. Zhao, B. Li, and B. Yuan, "Dynamic hierarchical energy-efficient method based on combinatorial optimization for wireless sensor networks," *Sensors*, vol. 17, no. 7, pp. 1665–1679, Jul. 2017.

[15] M. M. Shirmohammadi, K. Faez, and M. Chhardoli, "LELE: Leader election with load balancing energy in wireless sensor network," in *Proc. IEEE Int. Conf. Commun. Mobile Comput.*, vol. 2, Jan. 2009, pp. 106–110.

[16] T. L. Saaty, *The Analytic Hierarchy Process*. New York, NY, USA: McGraw-Hill, 1980.

[17] Y. Chang, H. Tang, B. Li, and X. Yuan, "Distributed joint optimization routing algorithm based on the analytic hierarchy process for wireless sensor networks," *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2718–2721, Dec. 2017.

[18] Y. Chang, X. Yuan, B. Li, D. Niyato, and N. Al-Dhahir, "A joint unsupervised learning and genetic algorithm approach for topology control in energy-efficient ultra-dense wireless sensor networks," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2370–2373, Nov. 2018.

[19] C. Li, M. Ye, G. Chen, and J. Wu, "An energy-efficient unequal clustering mechanism for wireless sensor networks," in *Proc. IEEE Int. Conf. Mobile Adhoc Sensor Syst. Conf. (MASS)*, Nov. 2005, p. 8.

[20] J.-S. Lee and T.-Y. Kao, "An improved three-layer low-energy adaptive clustering hierarchy for wireless sensor networks," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 951–958, Dec. 2016.

[21] I. Goodfellow, Y. Bengio, A. Courville, and F. Bach, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[22] M. A. Alsheikh, S. Lin, D. Niyato, and H. P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1996–2018, 4th Quart., 2014.

[23] X. Liu and D. He, "Ant colony optimization with greedy migration mechanism for node deployment in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 39, pp. 310–318, Mar. 2014.

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

[25] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo, "A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 550–586, 1st Quart., 2017.

[26] M. R. Senouci, A. Mellouk, L. Oukhellou, and A. Aissani, "An evidence-based sensor coverage model," *IEEE Commun. Lett.*, vol. 16, no. 9, pp. 1462–1465, Sep. 2012.

[27] M. R. Senouci, A. Mellouk, N. Aitsaadi, and L. Oukhellou, "Fusion-based surveillance WSN deployment using Dempster–Shafer theory," *J. Netw. Comput. Appl.*, vol. 64, pp. 154–166, Apr. 2016.

[28] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 4, pp. 609–619, Aug. 2004.

[29] X.-Y. Zhang, J. Zhang, Y.-J. Gong, Z.-H. Zhan, W.-N. Chen, and Y. Li, "Kuhn–Munkres parallel genetic algorithm for the set cover problem and its application to large-scale wireless sensor networks," *IEEE Trans. Electromagn. Compat.*, vol. 20, no. 5, pp. 695–710, Oct. 2016.

[30] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[31] G. Hoblos, M. Staroswiecki, and A. Aitouche, "Optimal design of fault tolerant sensor networks," in *Proc. IEEE Int. Conf. Control Appl.*, Anchorage, AK, USA, Sep. 2000, pp. 467–472.

[32] A. Syarif, I. Benyahia, A. Abouaissa, L. Idoumghar, R. F. Sari, and P. Lorenz, "Evolutionary multi-objective based approach for wireless sensor network deployment," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 1831–1836.

[33] B. Wang, *Coverage Control in Sensor Networks*. New York, NY, USA: Springer-Verlag, 2010.

[34] Z. Lu, W. W. Li, and M. Pan, "Maximum lifetime scheduling for target coverage and data collection in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 2, pp. 714–727, Feb. 2015.

[35] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley, 1989.

[36] T. H. Michael, *Scientific Computing: An Introductory Survey*, 2nd ed. New York, NY, USA: McGraw-Hill, 2002.

[37] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 1986.

[38] D. B. Fogel, "An introduction to simulated evolutionary optimization," *IEEE Trans. Neural Netw.*, vol. 5, no. 1, pp. 3–14, Jan. 1994.

[39] W. Zhang and Y. leung, *Mathematical Foundation of Genetic Algorithms*. Xi'an, China: Xi'an Jiaotong Univ., 2003.

[40] X. Feng, J. Zhang, C. Ren, and T. Guan, "An unequal clustering algorithm concerned with time-delay for Internet of Things," *IEEE Access*, vol. 6, pp. 33895–33909, 2018.

YUCHAO CHANG is currently pursuing the Ph.D. degree with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, and also with the University of Chinese Academy of Sciences. He joins the Prof. Naofal Al-Dhahir Group, The University of Texas at Dallas, as a visiting Ph.D. student, from 2018 to 2019. His research interests include wireless communications and machine learning applications.

XIAOBING YUAN received the Ph.D. degree from the Changchun Institute of Optics, Fine Mechanics, and Physics, Chinese Academy of Sciences, Changchun, China, in 2000. He is currently a Professor with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences. He is the co-inventor of 100 issued Chinese patents and the co-author of 50 papers. His research interests include wireless sensor networks and information transmission.

BAOQING LI received the Ph.D. degree from the State Key Laboratory of Transducer Technology, Shanghai Institute of Metallurgy, Chinese Academy of Sciences, Shanghai, China, in 2000. He is currently a Professor with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences. He is the co-inventor of 100 issued Chinese patents and the co-author of 50 papers. His research interests include signal processing and wireless sensor networks.

DUSIT NIYATO (M'09–SM'15–F'17) received the B.Eng. degree from the King Mongkuts Institute of Technology Ladkrabang, Thailand, in 1999, and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Canada, in 2008. He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include the Internet of Things and network resource pricing.

NAOFAL AL-DHAHIR (F'08) received the Ph.D. degree from the Department of Electrical Engineering, Stanford University. From 1994 to 2003, he was a Principal Member of the Technical Staff at the GE Research and AT&T Shannon Laboratory. He is the Erik Jonsson Distinguished Professor with The University of Texas at Dallas. He is the co-inventor of 41 issued U.S. patents, co-author of over 400 papers, and co-recipient of four IEEE best paper awards. He is the Editor-in-Chief of the IEEE Transactions on Communications.