

A Hyper-Heuristic Framework for Lifetime Maximization in Wireless Sensor Networks With A Mobile Sink

Jinghui Zhong, Zhixing Huang, Liang Feng, Wan Du, and Ying Li

Abstract—Maximizing the lifetime of wireless sensor networks (WSNs) is an important and challenging research problem. Properly scheduling the movements of mobile sinks to balance the energy consumption of wireless sensor network is one of the most effective approaches to prolong the lifetime of wireless sensor networks. However, the existing mobile sink scheduling methods either require a great amount of computational time or lack effectiveness in finding high-quality scheduling solutions. To address the above issues, this paper proposes a novel hyperheuristic framework, which can automatically construct high-level heuristics to schedule the sink movements and prolong the network lifetime. In the proposed framework, a set of low-level heuristics are defined as building blocks to construct high-level heuristics and a set of random networks with different features are designed for training. Further, a genetic programming algorithm is adopted to automatically evolve promising high-level heuristics based on the building blocks and the training networks. By using the genetic programming to evolve more effective heuristics and applying these heuristics in a greedy scheme, our proposed hyper-heuristic framework can prolong the network lifetime competitively with other methods, with small time consumption. A series of comprehensive experiments, including both static and dynamic networks, are designed. The simulation results have demonstrated that the proposed method can offer a very promising performance in terms of network lifetime and response time.

Index Terms—Genetic programming, hyper-heuristics, static and dynamic networks, wireless sensor networks (WSNs).

Manuscript received July 11, 2019; revised September 18, 2019; accepted October 17, 2019. This work was supported by the National Natural Science Foundation of China (61602181,61876025), Program for Guangdong Introducing Innovative and Entrepreneurial Teams (2017ZT07X183), Guangdong Natural Science Foundation Research Team (2018B030312003), the Guangdong–Hong Kong Joint Innovation Platform (2018B050502006), and the Fundamental Research Funds for the Central Universities (D2191200). Recommended by Associate Editor Zhen Song. (*Corresponding author: Wan Du*)

Citation: J. H. Zhong, Z. X. Huang, L. Feng, W. Du, and Y. Li, “A hyperheuristic framework for lifetime maximization in wireless sensor networks with a mobile sink,” *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 1, pp. 223–236, Jan. 2020.

J. H. Zhong and Z. X. Huang are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: jinghuizhong@gmail.com; zhi.you.xing@163.com).

L. Feng is with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: liangf@cqu.edu.cn).

W. Du is with the Department of Computer Science and Engineering, School of Engineering, The University of California, Merced USA (e-mail: wdu3@ucmerced.edu).

Y. Li is with the Binxun Technology Co., Ltd, Shenzhen 518000, China (e-mail: liying7820@163.com).

Digital Object Identifier 10.1109/JAS.2019.1911846

I. INTRODUCTION

WIRELESS sensor networks (WSNs) have been deployed for a wide range of applications such as water quality monitoring [1], [2], smart cities [3], wildlife protection [4], and medical care [5]. In real-world applications, the lifetime of wireless sensor networks is one of the most important attributes affecting the performance of the network, since sensors in wireless sensor network are often battery-powered and disposable. Prolonging the network lifetime is a significant and challenging research topic in the wireless sensor network community [6]–[9]. Over the past decades, various approaches have been proposed to maximize the lifetime of wireless sensor networks [10]–[16]. Among them, scheduling the movements of mobile sinks to balance the energy consumption of sensors is one of the most effective strategies [17]. In particular, a mobile sink, which can move to different locations of the sensing region, is the device utilized to collect data from sensors and send data to the base station. The proper movement scheduling of the mobile sinks is essential to the energy consumption of the network and thus able to prolong the network lifetime.

However, to find the optimal movement of the mobile sink is an NP-hard optimization problem [18]. In the literature, various methods have been proposed to solve this problem, including linear programming methods (e.g., the mixed integer linear programming model (MILP) [18]), greedy heuristic methods (e.g., the greedy maximum residual energy (GMRE) [17]), and stochastic search methods (e.g., the ant colony optimization (ACO) [19]). Traditional linear programming methods generally require a large amount of memory and huge computational time, which makes them inapplicable for large-scale wireless sensor networks. The greedy heuristic methods are not able to provide high quality solutions due to the inefficient human-design heuristics. The stochastic search methods, on the other hand, are capable of finding high quality solutions. Nevertheless, they often require a huge amount of computational time, since they contain an iterative search process for optimization during the application. Besides, due to the long response time, both the existing linear programming methods and the existing stochastic search methods are not suitable for dynamic cases whose network settings are changing over time.

Keeping the above in mind, to find high quality solutions efficiently, a novel hyper-heuristic framework (HHF) is proposed in this paper. In the proposed framework, primitives

(i.e., the low-level heuristics and linking functions) for constructing high-level heuristics and training networks for simulating the performance of evolved high-level heuristics are firstly designed, respectively. Then, the genetic programming (GP) based framework is developed to evolve better high-level heuristics off-line, based on the predefined primitives and training networks. In our framework, the high-level heuristics, which consist of basic information of networks and other human-designed functions, are denoted as H_2 s. Lastly, the H_2 with the best performance (i.e., the one can offer the longest average lifetime on training networks) is outputted as the final solution. This outputted H_2 can then be used to schedule the movements of the mobile sink in unseen testing networks.

There are several advantages of the proposed HHF to break the dilemma of solution quality and response time. 1) Because HHF designs heuristics (i.e., H_2 s) automatically based on some low-level heuristics, it avoids the tedious and time-consuming manual designing process of the heuristic rules. 2) Since typical heuristic rules serve as a part of low-level heuristics in the search space, HHF is likely to evolve more efficient and flexible H_2 s than typical ones. 3) Because the proposed HHF is trained off-line before the application and the obtained H_2 s can be used directly in new unseen networks, the obtained H_2 s mitigate the huge time consumption of linear programming methods and ACO, and have a short response time in real-world applications. The short response time also makes HHF suitable for dynamic networks.

This paper has designed a series of comprehensive experiments to analyze the proposed HHF. A large number of static and dynamic networks with different scales have been designed for testing and several state-of-the-art methods are also compared with the proposed method. Besides, different training schemes of HHF, including training on matching and mismatching training networks, have also been investigated to facilitate the practicability of the proposed HHF in real-world applications. The empirical results have demonstrated that HHF can produce feasible high-quality solutions in various scenarios and has a short response time.

This paper extends our early paper [20] with three improvements: 1) This paper extends [20] with more practical formulation of the to-be-solved problem, more comprehensive technical details of the algorithm implementation, and more theoretical analysis; 2) The previous work focused on static networks, while in this work, the proposed method is applied to both static and dynamic networks. The corresponding new experiments on dynamic networks are also conducted to validate the efficacy of the proposed framework; 3) A much more comprehensive experimental study has been conducted to facilitate the practical applications of the proposed method, with more procedure results and more compared methods. The final performance and the procedure results both show the effectiveness and reliability of the proposed method.

The rest of the paper is organized as follows. Section II describes the related works. Section III gives the problem definition. The proposed hyper-heuristic framework is presented in Section IV, which is followed by the empirical studies in Section V and Section VI. Finally, Section VII draws the conclusion.

II. RELATED WORKS

Mobile sink scheduling problem in wireless sensor networks requires finding a proper visiting order and sojourn time of the candidate sink sites, so that the lifetime of the network can be maximized. Over the past decades, various methods have been proposed to solve the mobile sink scheduling problem [8], [21]–[23]. Generally, existing methods can be categorized into two main classes: the centralized algorithms and the distributed algorithms [17]. The main difference between these two types of algorithms is how much information can be utilized in searching the route of sink movement. For the centralized algorithms, all information on the map can be aggregated to the sink (so called “centralized”) so that the sink movement can better approximate the global optima. In contrast, the distributed algorithms can only know the information around it. It means when there are multiple sinks on the map, each sink searches its own route respectively only based on local information.

In the centralized algorithms, the information of the whole system, such as the residual energy and the consumption rate of each sensor, is given. Based on the global information of the system, the centralized algorithms usually regard the sink scheduling problem as mathematical optimization problems (e.g., linear programming problems). For example, Gandham *et al.* [24] proposed to use integer linear programming (ILP) to determine the locations of the base stations (i.e., the sinks), with an objective to minimize the maximum energy consumption of the network in scheduling. However, there is no constraint on the sink movements and no relation between the number of base stations and their locations. Wang *et al.* [25] formulated the mobile sink scheduling problem as an optimization problem of maximizing the sum of sojourn time of the sink at every candidate sink site. They proposed a linear programming (LP) method to solve the problem. Since the LP-based methods discussed above are very time-consuming, Luo *et al.* [26] proposed an enhanced primal-dual algorithm to reduce the computational time. Generally, the centralized algorithms can find the global optimal solution. However, they require the global information of the system in advance, which is often impractical in real-world applications. Besides, the centralized algorithms usually require a large amount of computation time and memory, which makes them inapplicable for large-scale wireless sensor networks.

To overcome the limitations of the centralized algorithms, distributed algorithms have been proposed [27]–[29]. For example, Basagni *et al.* [17] proposed a greedy maximum residual energy (GMRE) heuristic rule to schedule the sink movement, and found approximation solutions of the global optimal solutions in mobile sink scheduling. In the GMRE, the sink prefers to move to candidate sink sites which possess the largest residual energy in the neighborhood. The GMRE is further discussed and applied in various scenarios [30]. Since only optimizing the sink movement is not yet effective enough to prolong the network lifetime, several joint optimization problems with controlled sink movements have also been formulated and solved by distributed algorithms. For instance, Yun *et al.* [31] proposed a distributed algorithm to maximize

the network lifetime and determine the data flow routing simultaneously, in a delay-tolerant wireless sensor network. Although the distributed algorithms have enjoyed a great success in many wireless sensor networks, the performance of the existing distributed algorithms requires to be improved for real-world applications.

In recent years, several researchers have proposed to use evolutionary algorithms to schedule the mobile sink in the wireless sensor networks. For example, Carrabs *et al.* [32] combined the genetic algorithm (GA) with the column generation techniques to maximize the network lifetime. Zhong *et al.* [19] and Xie *et al.* [33] adopted the ant colony optimization (ACO) to solve the lifetime maximization and data aggregation in wireless sensor network, respectively. Recently, based on their ideas, Kumar *et al.* [34] also proposed an ACO based mobile sink framework to solve the sink deployment and path determination with a non-uniform data sensing rate. Since the evolutionary algorithm shows a great potential in prolonging the network lifetime, its corresponding researches on wireless sensor networks are concluded in some surveys [35], [36]. However, all these algorithms have a drawback limiting their practicability in real-world dynamic networks. They usually require a much larger amount of computational time due to their iterative nature and long simulation time.

To relieve the disadvantages in the centralized algorithms and the evolutionary algorithms, and to further improve the performance of the distributed algorithms, a new hyper-heuristic framework is proposed in this paper. The hyper-heuristics is a type of algorithms designing heuristics automatically based on a set of given low-level heuristics and a given problem [37]. Currently, it has been widely and successfully applied in various practices, such as job shop scheduling [38], production scheduling [39], and other kinds of combinatorial optimization problems [40]. Especially, the genetic programming based hyper-heuristics methods is an important branch of hyper-heuristics [41]–[43]. In contrast to the centralized algorithms, the proposed hyper-heuristic method uses the obtained heuristics to schedule sink movements, which can be trained off-line and have a short response time. Meanwhile, unlike existing greedy heuristics that were designed by human experts, the proposed method can construct effective heuristics automatically based on simple low-level heuristics.

III. PROBLEM DEFINITION

In this section, the definition of the mobile sink scheduling problem in wireless sensor networks is introduced. We denote the given wireless sensor network as $M = (S, C, \Psi)$, where S ($|S| = n$) is the set of sensors, C ($|C| = m$) is the set of candidate sink sites, and Ψ is the sensing region of M so that $\{s \in \Psi, c \in \Psi | \forall s \in S, \forall c \in C\}$. n and m are the number of sensors and the number of candidate sink sites, respectively. As in [18], some assumptions of the network are described as follows.

1) The sink moves around among C and collects data from sensors. The sink is supposed to have unlimited energy which can support its running during the whole network lifetime.

2) The sojourn time of the sink at each candidate sink site is the number of rounds the sink stays at the site (each round lasts for a minimum time interval of Δt). The lifetime of a wireless sensor network (denoted as L) is regarded as the sum of rounds from the beginning to the time when at least one sensor runs out of its power. The L is computed by (1)

$$L = \sum_{c=1}^m t_c \quad (1)$$

where t_c is the number of rounds the sink stays at the c th sink site. To describe the location of the sink in the l th round, another variable δ_c^l is also introduced.

$$\delta_c^l = \begin{cases} 1, & \text{if the sink stays at the } c\text{th sink site} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

For wireless sensor networks with a single sink, we have $\sum_c^m \delta_c^l = 1, \forall l$. And according to the definition of δ_c^l , t_c can be calculated by $t_c = \sum_l \delta_c^l$.

3) The sensors are distributed in the sensing region according to a sensor-updating function ℓ . The distribution of sensors in the next round S^{l+1} is defined as $S^{l+1} = \ell(S^l)$. In particular, $S^{l+1} = \ell(S^l) = S^l, \forall l$ indicates that the sensors are static. Sensors sense the physical environment and transmit the data packets to the sink. Every sensor has a same limited initial energy E_i ($i \in S$) and the sensing data rate of each sensor is r .

4) The sink and sensors have a maximum communication range R . Sensor i can transmit data packets to sensor j (or the sink) in one hop if and only if $d_{i,j} \leq R$ where $d_{i,j}$ is the Euclidean distance between sensor i and sensor j (or the sink).

5) The sensors can send data to the sink by a multi-hop transmission tree (H) if their distances to the sink are larger than R . The amount of data sent (or received) by sensor i in the l th round is denoted as p_i^l (or q_i^l). Both p_i^l and q_i^l are determined based on H , which is calculated by the flow augmentation algorithm (FA) proposed in [44]. Therefore, p_i^l and q_i^l can be regarded as the results of FA method whose inputs are $S^l, c : \delta_c^l = 1$, and the residual energy of sensors $G^l = [g_1^l, g_2^l, \dots, g_n^l]$, as shown in (3). The H is reconstructed when the sink moves to another $c \in C$.

$$[p_i^l, q_i^l] = FA(S^l, c : \delta_c^l = 1, G^l). \quad (3)$$

6) For a certain sensor i , the amount of inflow data is always equal to the amount of outflow data. The inflow data is made up of the sensing data generated by itself ($r\Delta t$) and the transmission data from other sensors in the subtree of H ($\|q_i^l\|$). The outflow data ($\|p_i^l\|$) is the transmission data sent from sensor i to next node (i.e., another sensor or the sink) in H .

7) The energy consumption of a sensor only consists of data sending and data reception. The consumption rates of data sending and reception of sensor i in the l th round are et_i^l and er_i^l , respectively. Each component $et_{i,j}^l$ of et_i^l is a function of the distance $d_{i,j}^l$, as shown in (4). The A and B are constant coefficients, and $d_{i,j}^l$ is the Euclidean distance between sensor i and sensor j ($\forall j \in S \cup C$). In our model, er_i^l is a

constant vector.

$$\mathbf{et}_i^l = A \times d_{i,j}^{l,2} + B. \quad (4)$$

Therefore, based on the assumptions 5) and 7), the energy consumption of a sensor during the network lifetime can also be calculated by $\sum_l (\mathbf{et}_i^l \cdot \mathbf{p}_i^l + \mathbf{er}_i^l \cdot \mathbf{q}_i^l)$.

Besides, there are some complements about these assumptions. In assumption 1), C is a finite set. The traveling time t between two different $c \in C$ is ignored because $t \ll \Delta t$. In assumption 5), all sensors are reachable in H . This means that for any $i \in S$, there must be at least one $j \in S$ so that $d_{i,j} \leq R$. The routing overhead p_o generated by the FA method is ignored because p_o is much smaller than the amount of sending data p_i ($\forall i \in S$). In assumption 7), the movements of sensors will not generate energy consumption because in real-world applications, the movements of sensors are usually caused by physical environment factors such as the ocean flow.

Based on the assumptions introduced above, the lifetime maximization problem is modeled as follows.

$$\max L = \sum_{c=1}^m t_c \quad (5)$$

$$\text{s.t. } t_c = \sum_l \delta_c^l, \forall c \in C \quad (6)$$

$$\sum_c \delta_c^l = 1, \forall l \quad (7)$$

$$\|\mathbf{p}_i^l\| - \|\mathbf{q}_i^l\| - r\Delta t = 0, \forall i \in S \quad (8)$$

$$\|\mathbf{p}_i^l\|_0 = 1, \forall i \in S \quad (9)$$

$$E_i - \sum_l (\mathbf{et}_i^l \cdot \mathbf{p}_i^l + \mathbf{er}_i^l \cdot \mathbf{q}_i^l) \geq 0, \forall i \in S. \quad (10)$$

The objective of the problem is to maximize the lifetime of the whole network. Constraints (6) and (7) describe the sojourn time of each candidate sink site and that there is only one mobile sink in M . Constraint (8) introduces the flow conservation mentioned in assumption 6), which means the outgoing flow $\|\mathbf{p}_i^l\|$ is always equal to the sum of the reception flow $\|\mathbf{q}_i^l\|$ and the self-sensing flow $r_i\Delta t$ during a round. Constraint (9) defines that every sensor can have only one parent in the multi-hop transmission tree H . Constraint (10) requires that the residual energy of every sensor must be larger than 0 when the network is alive. Otherwise, the network is dead and the final lifetime is outputted.

IV. PROPOSED FRAMEWORK

In this section, the general framework of the proposed hyper-heuristic framework is introduced at first. Then, the major components of the proposed framework, i.e., the training network design, primitive design, and hyper heuristic discovery, are introduced.

A. General Framework

The heuristic-based scheduling strategy commonly works as

follows: given a minimum sojourn time Δt , the mobile sink periodically makes a decision in each Δt to decide whether to stay at the current sink site or to move to the next sink site based on a heuristic rule. This process is repeated until the network dies. One typical example of the heuristic rule is the GMRE [18]. Based on this scheduling strategy, how to design a suitable heuristic Γ to select sink sites properly is the key point to prolong the network lifetime. In this way, the problem of determining the optimal path of the mobile sink can be transformed into designing a proper heuristic Γ^* that maximizes the network lifetime L , as can be expressed by (11)

$$\Gamma^* = \arg \max_{\Gamma} T(\Gamma) \quad (11)$$

where $T(\Gamma)$ is the scheduling network lifetime based on Γ and Γ is used as the heuristic rule to schedule the mobile sink.

To solve (11), the HHF is specially developed for finding proper heuristics automatically. Fig. 1 illustrates the general framework of the proposed HHF. The HHF contains three parts: training network design, primitive design, and hyper heuristic discovery. Firstly, according to the prior knowledge of the unseen testing networks, a number of training networks and a set of primitives are designed. Then, the GP-based hyper heuristic discovery is performed to design H_{2s} based on the training networks and primitives. By applying the designed H_{2s} to schedule the sink movements on training networks, the fitness of H_{2s} (i.e., the average lifetime of simulated training networks) is obtained. The hyper heuristic discovery and the fitness evaluation of H_{2s} are performed iteratively to evolve better H_{2s} . Finally, the H_2 with the best fitness is outputted as the final solution (i.e., Γ^*) when the termination condition is met. The Γ^* is used to schedule the sink movements in unseen testing networks.

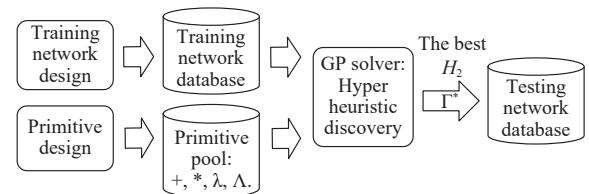


Fig. 1. The proposed hyper-heuristic framework.

B. Training Network Design

Several factors should be considered during the design of training networks, including the number of sensors n , the deployment of sensors, the number of candidate sink sites m , the deployment of candidate sink sites, and the updating rule of sensors ℓ . The deployment of sensors describes the locations of sensors monitoring the physical environment. One of the most common deployment strategies is the uniform deployment strategy. Besides, in this study, the deployment of the candidate sink sites in training and testing networks is obeying the “grid distribution”. The grid distribution means that the candidate sink sites are obtained by dividing the sensing region into grids and the grid centers are considered as the locations of candidate sink sites. By generating the sensors and candidate sink sites obeying uniform distribution and grid distribution, we can obtain a number of random training

networks with different features.

To make the design process clearer, here is an example for training network design. Suppose we want to define a 2-D stationary ($\ell(S^l) = S^l$) training networks with 50 sensors ($n = 50$) and 30 candidate sink sites ($m = 30$) in a square sensing region. The network is formulated as $M = (S, C, \Psi)$, where $S \in \mathbb{R}^{50 \times 2}$ follows the uniform distribution (S : uniform distribution), $C \in \mathbb{R}^{30 \times 2}$ follows the grid distribution (C : grid distribution), and $\Psi = (0, 100) \times (0, 100)$. We can firstly determine the candidate sink sites by “plotting” grids on the $(0, 100) \times (0, 100)$ map. Then we generate 50 sensors by sampling a series of positions obeying the uniform distribution within the map. The final network example is shown in Fig. 2. By this means, various network cases can be generated based on different network formulations.

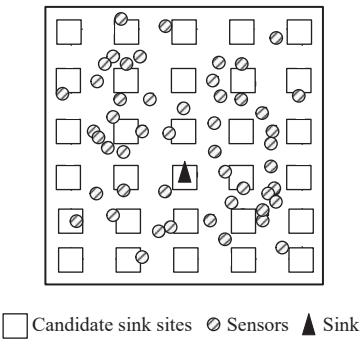


Fig. 2. An example 2-D training network.

C. Primitive Design

The design of primitives (i.e., low-level heuristics and concatenate functions) is problem specific. In this paper, we adopt four common arithmetic operations as linking functions (i.e., $+, -, \times, \div$) and design five typical and general low-level heuristics to construct the hyper-heuristics. The five low-level heuristics are defined as follows.

1) *Minimum Residual Energy* (λ): λ represents the minimum residual energy of sensors which can communicate with the sink in one hop (i.e., the sensors less than R away from the sink). Because the sensors close to the sink often need to transmit a lot of packets from the subtree of H , which makes them run out of energy quickly. By considering λ , the $c \in C$ covering sensors with larger minimum residual energy is more likely to be selected. Specifically, λ is computed as follows.

$$\lambda = \min_i g_i^l, \forall i \in S^l, d(i, c) \leq R \quad (12)$$

where $d(i, c)$ is the Euclidean distance between sensor i and the candidate sink site c .

2) *Maximum Residual Energy* (Λ): Λ represents the maximum residual energy of sensors whose $d(i, c) \leq R$. It is opposite to λ . In fact, Λ is a popular heuristic rule used in the greedy maximum residual energy (GMRE), which was proposed in [17]. The computation of Λ is as follows.

$$\Lambda = \max_i g_i^l, \forall i \in S^l, d(i, c) \leq R. \quad (13)$$

3) *Local Simulated Network Lifetime* (κ): κ implies the sink to select the site which leads to the maximum simulated

network lifetime. Here the simulated network lifetime is calculated by keeping the sink staying at that sink site till the network dies. To relieve the routing overhead p_o , the simulation just considers the sensors less than R away from the sink site. Therefore, the simulated lifetime is the minimum of the quotient between the residual energy and the energy consumption among the sensors whose $d(i, c) \leq R$. This heuristic integrates the residual energy and the consumption rate to imply the potential lifetime of every $c \in C$.

$$\kappa = \min_i \frac{g_i^l}{\mathbf{e} \mathbf{t}_i^l \cdot \mathbf{p}_i^l + \mathbf{e} \mathbf{r}_i^l \cdot \mathbf{q}_i^l} \quad \forall i \in S^l, d(i, c) \leq R. \quad (14)$$

4) *Average Sensor Energy* (μ): μ guides the sink to the sink site that has the maximum surrounding energy density. μ is computed as follows.

$$\mu = \frac{\sum_{i \in \Phi} g_i^l}{\|\Phi\|_0 + 1} \quad \Phi = \{i | \forall i \in S^l, d(i, c) \leq R\}. \quad (15)$$

The “+1” in denominator is used to guarantee that the denominator is always a positive number.

5) *Average Consumption Rate of Sensors* (ν): ν guides the sink to the site with the minimum average consumption rate of sensors surrounding $c \in C$. The lower average consumption rate implies a longer network lifetime in some way. ν is computed as follows.

$$\nu = \frac{\sum_{i \in \Phi} \mathbf{e} \mathbf{t}_i^l \cdot \mathbf{p}_i^l + \mathbf{e} \mathbf{r}_i^l \cdot \mathbf{q}_i^l}{\|\Phi\|_0 + 1} \quad \Phi = \{i | \forall i \in S^l, d(i, c) \leq R\}. \quad (16)$$

D. GP-based Hyper Heuristic Discovery

In this subsection, a GP-based hyper heuristic discovery approach (GPHH) is proposed. The flowchart of the proposed approach is shown in Fig. 3, which consists of four major steps: initialization, reproduction, fitness evaluation, and selection. In the proposed approach, a population of chromosomes are initialized at first. Each chromosome represents a candidate H_2 for scheduling the sink movements. Then, new H_2 s are generated using genetic operators (the implementation of genetic operators is further introduced below). The fitness values of newly generated H_2 s are evaluated by simulating the training networks. Finally, the H_2 s with higher fitness values are selected to form the new population for the next generation. When the termination conditions are met, the best H_2 is considered as the outputted H_2 to schedule the sink movements on unseen testing networks. In this paper, a recently published GP variant named self-learning gene expression programming (SL-GEP) [45] is adopted as the GP solver to evolve H_2 s. It is worth mentioning that other GP variants can also be adopted as the GP solver in the proposed framework. The implementation details of the proposed approach are as follows.

1) *Chromosome Representation*: In SL-GEP, each chromosome is a vector of symbols $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,j}]$. Each symbol $x_{i,j}$ can be a function (e.g., $+$, \times), a low-level

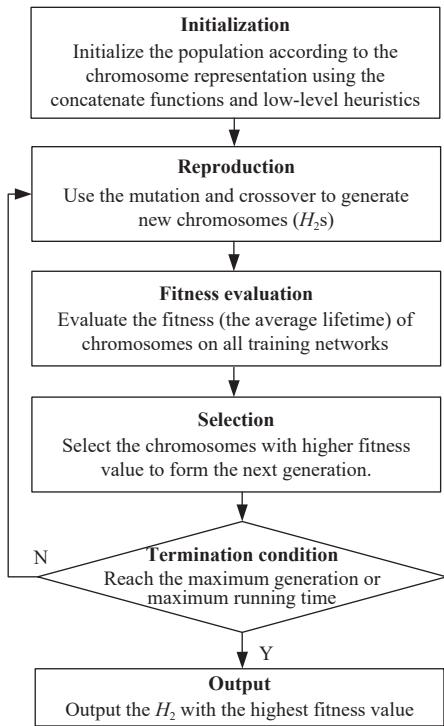


Fig. 3. The flowchart of the hyper heuristic discovery.

heuristic (e.g., λ and Λ), and other symbol types (e.g., the automatically defined function (ADF) mentioned in [45]). Each chromosome can be decoded as a parsing tree to represent a formula (i.e., a H_2) by using a breath-first-search travelling scheme. Fig. 4 shows an example of the chromosomes in SL-GEP, which contains a main program and one ADF. In this example, the chromosome “ $+ \times ADF \Lambda \mu \kappa \lambda - I_1 I_2$ ” is decoded as “ $\Lambda \times \mu + \kappa - \lambda$ ”. More detail introduction of the chromosome representation can be referred in [45].

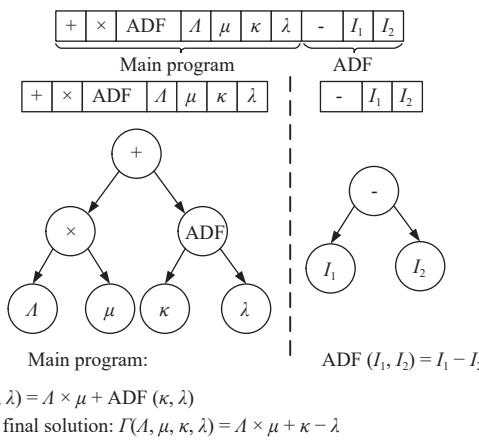


Fig. 4. The parsing tree of an example chromosome.

2) *Reproduction*: The reproduction includes two main operations: mutation and crossover. These two operations are used to change the genes in chromosomes to generate new chromosomes. In mutation operation, the symbols in chromosomes X_i are modified into other values based on the “DE/current-to-best/1” mutation, as introduced in [45]. For

example, the five-value chromosome “ $\Lambda + \mu \kappa \lambda$ ” can give birth to another five-value chromosome “ $\Lambda + \nu \kappa \lambda$ ” by mutating “ μ ” into “ ν ”. In crossover operation, new chromosomes are generated by substituting the segments of one chromosome into another chromosome. For example, two chromosomes “ $+ \times \Lambda \mu \kappa$ ” and “ $\sin + \lambda \nu \Lambda$ ” can generate a new chromosome “ $\sin + \Lambda \mu \kappa$ ” by substituting the first two values from latter chromosome into the former one.

3) *Fitness Evaluation*: To evaluate the fitness value of a newly generated H_2 , a simulation is performed to schedule the sink in the training networks using the given H_2 . The average lifetime of all training networks is then used as the fitness value of the given H_2 . The simulation follows the “sense-think-act” paradigm, as shown in Fig. 5. For example, given a decoded chromosome Γ , the information of the network (e.g., the sensors deployment S^l and the residual energy G^l) is obtained firstly (i.e., sense). Then the Γ is calculated based on the network information to select the next sink site $c : \delta_c^{l+1} = 1$ (i.e., think). After that, the sink is moved to the selected sink site in simulation (i.e., act). For each training network, the given H_2 is applied to schedule the movements of the sink using the “sense-think-act” paradigm repetitively until the death of the training network. After all training networks are simulated using the above method, the average lifetime of all training networks is calculated as the fitness value (τ) of the given H_2

$$\tau = \frac{\sum_t L_t}{N} \quad (17)$$

where L_t is the lifetime of the t th training network and N is the number of training networks. It is worth mentioning that the starting point of the sink for each simulation is set to the candidate sink site which is closest to the middle point of the sensing region.

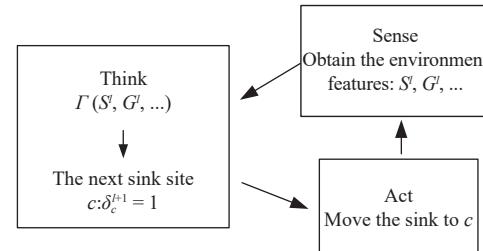


Fig. 5. The “sense-think-act” paradigm.

4) *Selection*: After the fitness values of all chromosomes are obtained, the selection operation is performed. The selection operation implements the “survival of the fittest” principle in nature evolution. In this operation, a certain number of better chromosomes are selected to form the new population. After the selection, the reproduction and selection operations are repeated until the termination conditions (e.g., the maximum generation) are satisfied. It is worth mentioning that the low-level heuristics are also the candidate solutions in the search space of the proposed HHF. Thus, the proposed HHF has potential to find the H_2 s that can perform better than the predefined low-level heuristics.

Overall, to apply the proposed framework more effectively, a careful design of training networks and primitives, and a well tune of GPHH parameters are needed. Among them, designing the training networks and primitives for GPHH is the most important step for application. The training networks should be designed with various features (e.g., the distribution of sensors, candidate sink sites, and the sensor updating rules) to produce a more general H_2 . It should be noticed that more properties of the application can also be considered in the training network design to improve the performance of obtained H_2 s, if these properties are available. At the same time, it is advisable to design effective primitives for GPHH, since the redundancy and ineffectiveness of the information in primitives will degrade the performance of GPHH. It is worth mentioning that the proposed framework can be easily transferred to other WSN applications such as noise filtering [46], [47]. For example, the proposed HHF can be used to automatically design a more robust triggering condition which needs careful beforehand design (i.e., the triggering threshold matrix and filter parameters) mentioned in [47].

V. EXPERIMENTS ON STATIONARY NETWORKS

In this section, to validate the effectiveness of the proposed HHF, the H_2 generated by the HHF is applied to stationary networks which can be categorized into two classes: the simple networks and the complex networks.

A. Experiment Design

The stationary networks are wireless sensor networks whose sensor deployment is unchanged during the network lifetime. In the experiment of stationary networks, the sensors are uniformly deployed in the sensing region. Ten types of networks (indexed from “0” to “9”) are designed for training and testing. There are ten testing wireless sensor networks and twenty training wireless sensor networks in each network type. The parameters of wireless sensor networks are set according to [48]: the sensing rate r is 1 bit/s, the least sojourn time Δt is set as 3600 seconds, and the coefficients A, B in (4) are set as 50 nJ/bit and 100 pJ/bit/m² respectively. The energy consumption for packet reception of every sensor is 50 nJ/bit. The maximum communication range of sensors and the sink R is 30 meters. The flow augmentation algorithm (FA) proposed in [44] is used to construct the multi-hop flow routing tree H . These parameters are also listed in Table I. All experiments are performed by a computer with the Intel Core i7-7700 with 3.6 GHz and four cores. The size of memory of the platform is 16 GB. Besides, there are several different parameters for each network type and they are listed in Table I. For example, the number of sensors is 50 for network “0” but 100 for network “1”. The initial energy is 50 Joule for network “0” to “7”, but 500 Joule for network “8” and “9”. Especially, the network “6” is set on a rectangle (width : $length = 1 : 3$) sensing region (i.e., similar to the health monitoring of bridges or skyscrapers) and the network “7” has a sink-forbidden region $(20, 70) \times (20, 70)$. The network “8” and “9” are two large-scale networks which are used to evaluate the generalization of testing methods.

There are three main parameters in SL-GEP: the population

TABLE I
THE PARAMETERS OF STATIONARY WIRELESS SENSOR NETWORKS

Index	Shared parameters	Settings
0		$n = 50, m = 5 \times 5,$ $\Psi(m) = (0, 100) \times (0, 100), E_i = 50 \text{ J}$
1		$n = 100, m = 5 \times 5,$ $\Psi(m) = (0, 100) \times (0, 100), E_i = 50 \text{ J}$
2		$n = 50, m = 10 \times 10,$ $\Psi(m) = (0, 100) \times (0, 100), E_i = 50 \text{ J}$
3		$n = 100, m = 10 \times 10,$ $\Psi(m) = (0, 100) \times (0, 100), E_i = 50 \text{ J}$
4	$r_i = 1 \text{ (bit/s)}$ $\Delta t = 3600 \text{ (s)}$	$n = 50, m = 20 \times 20,$ $\Psi(m) = (0, 100) \times (0, 100), E_i = 50 \text{ J}$
5	$A = 50 \text{ (nJ/bit)}$ $B = 100 \text{ (pJ/bit/m}^2\text{)}$	$n = 100, m = 20 \times 20,$ $\Psi(m) = (0, 100) \times (0, 100), E_i = 50 \text{ J}$
6	$er_i = 50 \text{ (nJ/bit)}$ $R = 30 \text{ (m)}$ Routing algorithm = FA [44]	$n = 100, m = 10 \times 30,$ $\Psi(m) = (0, 50) \times (0, 150),$ rectangle stationary networks, $E_i = 50 \text{ J}$
7		$n = 100, m = 20 \times 20,$ $\Psi(m) = (0, 100) \times (0, 100),$ the network with a sink-forbidden region. The forbidden region is $(20, 70) \times (20, 70), E_i = 50 \text{ J}$
8		$n = 1000, m = 50 \times 50,$ $\Psi(m) = (0, 200) \times (0, 200),$ large-scale stationary networks, $E_i = 500 \text{ J}$
9		$n = 1000, m = 100 \times 100,$ $\Psi(m) = (0, 200) \times (0, 200),$ large-scale stationary networks, $E_i = 500 \text{ J}$

size, the length of chromosome head, and the length of chromosome tail [45]. Generally speaking, the population size needs a careful adjustment to be appropriate. A large population size would lead to a slow convergence speed while a small one would result in easy premature. Besides, it is advisable for the length of chromosome head and tail to be proportional with the number of primitives so that the chromosomes can contain enough primitives to construct solutions. In our work, the parameters of SL-GEP are set the same as those in [45]. Further, to validate the performance of the HHF, the proposed HHF is compared with seven other methods. The first one is the linear programming method (LP(C-MB)), which is a classic deterministic method for maximizing the network lifetime with constrained sink mobility [49]. In our experiments, the LP(C-MB) is implemented by GNU Linear Programming Kit 4.60 (GLPK-4.60) which can be downloaded from the website¹. Since the LP(C-MB) considers the global information and the whole solution space, it can usually obtain the optimal solution but with a high cost of time and memory. The second one is the ant colony optimization (ACO) [19], a stochastic search algorithm, which has a strong global search ability and has been shown quite effective in solving network lifetime maximization with mobile sink. In this study, the parameters of ACO are set the same as [19]. The third one is the greedy maximum residual energy (GMRE) [17], which is a typical distributed algorithm for the sink movement scheduling. Each round, the sink will move to the candidate sink site which has the sensor with the highest residual energy. The GMRE can only provide a local optimal solution due to its local greedy searching manner. The other four methods are those which

¹<https://www.gnu.org/software/glpk/>

directly use the low-level heuristics introduced in Section IV-C as heuristic rules to guide the sink movements. For these four methods, the sink will move to the candidate sink site with higher heuristic value (lower heuristic value for v) in each Δt based on these heuristic rules. Since the Λ is the same as GMRE, it is omitted in the experiment.

B. The Training Process of HHF

First, the proposed HHF is utilized to learn H_2 s based on training wireless sensor networks. According to the training wireless sensor networks fed to HHF, two versions of HHF, the distinct HHF and the transfer HHF, are performed. In the distinct HHF, the twenty training wireless sensor networks and ten testing wireless sensor networks all come from the same network type. In the transfer HHF, the network type of training wireless sensor networks is different from the one of testing networks. In our experiment, the “5” network type is used as the training network type of the transfer HHF, which means the outputted H_2 of “5” network will be directly used to schedule the sink movements in any other network types in the transfer HHF. It is worth mentioning that the transfer HHF would be more generalized and practical since its prior knowledge of training networks can be not a perfect matching with the testing networks. But the transfer HHF is likely to perform worse than the distinct one because the transfer HHF has less prior knowledge of the training networks. For each training process, the SL-GEP is performed for 500 generations to search for a good H_2 based on twenty training networks from one type.

To investigate the convergence of the proposed method, the convergence curves of networks “1”, “5”, and “8” are taken as examples and they are shown in Fig. 6. The curves of network “1” and “5” both converge to a maximum value, with steep increase at the beginning of evolution and relatively steady increment after 200 generations. Although the curve of network “8” also reaches a maximum, it seems it has not converged yet since network “8” is a large-scale network. The increment of these three fitness curves validates the effectiveness of the GPHH training.

After 500 generations, the H_2 with the highest fitness value

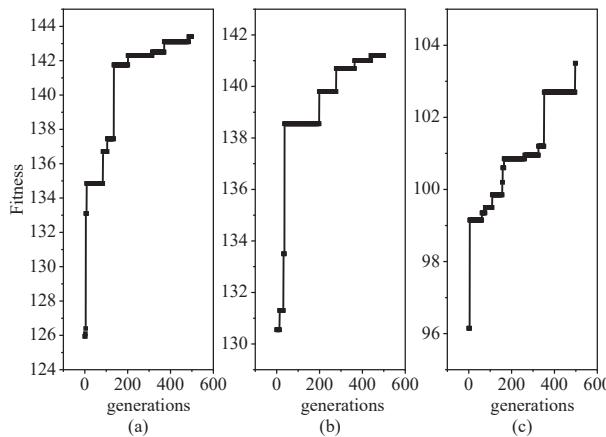


Fig. 6. The convergence curves of networks: (a) network “1”; (b) network “5”; (c) network “8”.

is outputted to guide the sink movement in unseen testing networks. The outputted H_2 s and the training time of the distinct HHF are shown in Table II. It is worth mentioning that, because the training time of networks “8” and “9” is too long (i.e., each generation will take more than 2 days) for our experiment, these two large-scale networks are omitted for the distinct HHF and networks “8” and “9” are only used to evaluate the generalization of other testing methods. It is clear that the time consumption of hyper heuristic discovery is pretty large, especially facing the large-scale network such as networks “8” and “9” which makes the training time unbearable. However, we still believe the HHF has a high practicability in real-world applications. On one hand, because the information of many stationary wireless sensor networks can be determined beforehand, the time-consuming training process can be performed off-line before applications. On the other hand, since the heuristic rules designed by the proposed method can be easily applied to different networks, it is flexible for users to train on small-scale networks and transfer the outputted heuristics to large-scale networks.

TABLE II
THE TRAINING RESULTS OF THE DISTINCT HHF ON STATIONARY WIRELESS SENSOR NETWORKS

Index	Hyper heuristic	Time (s)
0	$(\frac{\lambda + \frac{\lambda}{\Lambda}}{v} + \frac{\frac{\lambda + \frac{\lambda}{\Lambda}}{v} \times (\kappa + \frac{\kappa}{\Lambda})}{\lambda}) \times (\frac{\lambda}{\kappa} - \kappa)$	8251.8
1	$\frac{\mu}{\frac{\Lambda}{v} - 2v + \kappa}$	29 610.3
2	$(\frac{(v + \mu + \Lambda) \times (\kappa + \lambda)}{v \times \mu \times (v \times \mu + 2v)} - \kappa) \times \kappa$	22 105.5
3	$\frac{\lambda^9 \times (v - \kappa)}{v^2 \kappa^3}$	72 941.2
4	$\frac{\Lambda - \mu}{v + \Lambda - \mu}$	79 480.2
5	$\frac{(\frac{\Lambda}{v})^2 - \Lambda - (v^2 - \kappa) \times \frac{\lambda \times \kappa}{v^2}}{v^2}$	212 345
6	$\kappa \times (\frac{\lambda \times (\Lambda - \lambda) + \lambda \times (\kappa - \lambda)}{\Lambda \times (v \times \mu - \Lambda)} - \kappa)$	134 033
7	$\frac{(\kappa - v - \mu) \times \frac{v + \mu}{\kappa} \times (2v + \kappa)}{(\kappa - v) \times \frac{v}{\kappa} + \frac{\mu}{\kappa}}$	103 028

C. Empirical Results

To evaluate the quality of the solutions, two criterion metrics are adopted: the average network lifetime (AL) and the average decision time (ADT). The average network lifetime represents the average of the simulated network lifetime (L) of a certain method on the ten testing wireless sensor networks for each network type. Based on L given in (5), the decision time (DT) which represents the time consumption for each sink movement decision, is computed by (18)

$$DT = \frac{T_{run}}{L} \quad (18)$$

TABLE III
THE RESULTS OF STATIONARY WIRELESS SENSOR NETWORKS

	Index	LP(C-MB)	ACO	GMRE	λ	κ	μ	ν	HHF_d	HHF_t
0	<i>AL</i>	130.33	+	94.20	+	64.1	-	64.8	-	84.6
	<i>ADT(s)</i>	0.0766	-	0.0229	-	0.0010	-	0.0013	-	0.0003
1	<i>AL</i>	N/A		139.9	\approx	87	-	118.4	-	136.6
	<i>ADT(s)</i>	N/A		0.0887	-	0.0005	\approx	0.0005	\approx	0.0005
2	<i>AL</i>	N/A		110.0	\approx	64.9	-	62.9	-	101.8
	<i>ADT(s)</i>	N/A		0.0323	-	0.0008	\approx	0.0008	\approx	0.0007
3	<i>AL</i>	N/A		144.5	\approx	99.0	-	106.4	-	133.0
	<i>ADT(s)</i>	N/A		0.1005	-	0.0005	+	0.0005	+	0.0015
4	<i>AL</i>	N/A		107.4	\approx	65.0	-	71.9	-	101.5
	<i>ADT(s)</i>	N/A		0.0616	-	0.0058	+	0.0060	\approx	0.0026
5	<i>AL</i>	N/A		152.6	+	111.7	-	114.9	-	136.1
	<i>ADT(s)</i>	N/A		0.1375	-	0.0006	+	0.0007	+	0.0043
6	<i>AL</i>	N/A		112.9	\approx	85.6	-	86.4	-	109.7
	<i>ADT(s)</i>	N/A		0.142	-	0.0003	+	0.0008	+	0.0032
7	<i>AL</i>	N/A		137.4	+	84.4	-	111.3	+	125.1
	<i>ADT(s)</i>	N/A		0.134	-	0.0004	+	0.0008	+	0.0033
8	<i>AL</i>	N/A		763.0	\approx	504.9	-	668.6	-	725.6
	<i>ADT(s)</i>	N/A		9.124	-	0.0133	+	0.0127	+	0.2404
9	<i>AL</i>	N/A		761.9	+	485.5	-	641.5	-	727.9
	<i>ADT(s)</i>	N/A		12.10	-	0.0395	+	0.0346	+	0.9094

Symbols \approx , +, and - represent the corresponding method is similar to, significantly better, and worse than the transfer HHF according to the Wilcoxon rank-sum test at $\alpha = 5\%$. The same symbols are also used in Table VI

where T_{run} is the total running time of a wireless sensor network. The *ADT* is the average of *DT* on ten testing network cases. The *AL* measures the quality of solutions of methods while the *ADT* measures the efficiency of methods to obtain solutions.

Table III lists the simulation results on the testing networks, where HHF_d and HHF_t are the distinct version and transfer version of HHF, respectively. It can be observed that the *ALs* of the two HHFs are statistically similar to those of ACO in most cases. Furthermore, the *ADTs* of the HHFs are much shorter than those of ACO, and similar to those of the distributed algorithms. The LP(C-MB) method obtains the highest *AL* in the smallest network type. This is because the LP(C-MB) considers the global information to find solutions. Besides, the sojourn time of each sink site found by LP(C-MB) may not be the integer times of Δt , which makes LP(C-MB) reaches a better approximation of the global optimal solution. However, the LP(C-MB) requires too much memory and computational time, which makes it unsuitable for larger networks. Another centralized algorithm, ACO, has the second best performance in *AL* but a longer *ADT* compared with other distributed algorithms. As for other distributed algorithms, such as GMRE, they cannot obtain a satisfactory solution (i.e., the *AL* of GMRE is only 60% to 70% of the *AL* of ACO). All distributed algorithms have a similar short *ADT*. However, since the H_2 s are the combinations of low-level heuristics and need to be decoded from chromosomes before each round, the *ADTs* of H_2 s are usually lightly longer than

those of other distributed algorithms. The two implementations of HHF have a similar performance with each other in these stationary networks and they are capable of obtaining significantly higher *AL* than other distributed algorithms. The above results demonstrate that the proposed HHF is effective to find promising H_2 s to guide the sink movements.

To further analyze the performance of HHF, the cumulative distribution function (CDF) of the hop number of packets and the average sojourn time of every sink on each network type are given out. Because ACO and GMRE are the popular centralized scheduling algorithm and distributed algorithm, respectively, ACO, GMRE, and two versions of HHF are taken as the examples to be compared. The networks “7” and “8” are taken as examples because they are relatively more complex than other networks (i.e., network “7” with a sink-forbidden region and network “8” is large-scale). The CDF of the hop number of these two networks are shown in Fig. 7. To reduce the energy consumption and improve the transmission efficiency, the hop number of packets is expected to be as few as possible. Therefore, the CDF of the hop number reflects the efficiency of the network and implies the network lifetime in some way. It can be observed that, in network “7”, the curves of two HHFs even perform lightly better than ACO. And in network “8”, the CDF curves of ACO and HHF_t nearly coincide with each other while the curve of GMRE is much lower and broader than the curves of ACO and HHF_t .

The average sojourn time of these four methods on

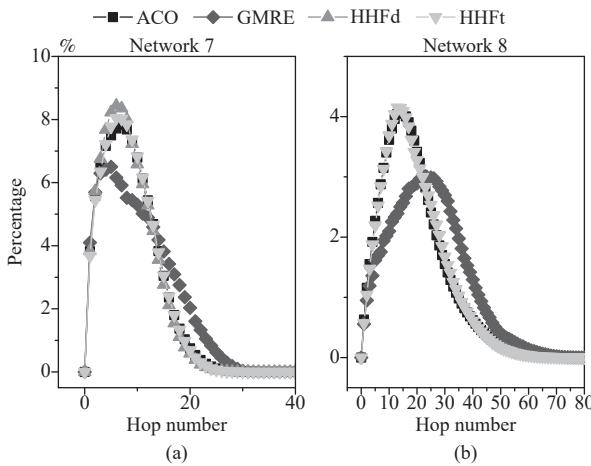


Fig. 7. The CDF of hop number in stationary networks. (a) the percentage vs. hop number on network “7”; (b) the percentage vs. hop number on network “8” (the curve of HHF_d is omitted because of unbearable training time).

networks “7” and “8” are shown in Fig. 8. For the network “7” which has the forbidden region, the sojourn times of sink sites of ACO and the two HHFs are mostly distributed along the margins of the forbidden region, which are the closest places to the central region of the map. This is because, for the square map, the H usually has a smaller hop number of the packets in the whole network when the sink in the central region of the sensing region. However, the sink in GMRE seldom selects the sink sites in the central part of the sensing region (i.e., the margins of forbidden region) and it prefers to stay at the marginal part of the sensing region. Such solutions bring a large hop number of packets in networks (i.e., some packets need to be transmitted through the whole map to reach the sink). Besides, the sojourn time of sink sites of GMRE only lies in a few candidate sink sites. This makes the sensors near these sink sites need to transmit much more packets than others during the network lifetime. This also makes the energy of a part of sensors drop down rapidly.

For the large-scale network “8”, the sojourn time of candidate sink site of ACO is mostly distributed in the center of the sensing region like the Gaussian distribution. The sojourn time in HHF_t looks different with the one of ACO, but they still have some common characteristics. Firstly, in HHF_t , several sink sites in the central region have a large sojourn time to make the networks have a small hop number of packets. Secondly, other sink sites with smaller sojourn times are distributed widely from the center to the margin of the sensing region to make the energy consumption of sensors more balanced. The similarity of the CDF of hop number of packets and the similarity of the average sojourn time in ASO and HHF both validate that the outputted H_2 of the proposed framework is quite efficient to have a competitive performance with ACO in terms of prolonging network lifetime.

VI. EXPERIMENTS ON DYNAMIC NETWORKS

A. Simulation Settings

In this section, three types of dynamic networks are

designed to test the effectiveness of the proposed method. These three types of dynamic networks are as follows: networks containing shaking sensors (e.g., those networks with sensors which detect positions containing noise), networks with migrating sensors (e.g., those used for monitoring the ocean flow), and networks with sensors gathering to a moving point (e.g., those used for target tracking). As same for stationary networks, there are twenty training networks and ten testing networks for each network type. ACO, GMRE, and two versions of HHF are taken as the testing methods to validate the effectiveness of the proposed HHF. The detail experiment settings are listed in Table IV. For network “10”, every sensor will move to another location in the sensing region randomly with different small step sizes every round. For network “11”, the sensors move around the 2-D sensing region obeying the clock-wise direction. For network “12”, there is a random moving point on the sensing region and it moves randomly every round. The sensors in network “12” will gather around the moving point each round. Other parameters are set the same with stationary networks. For these dynamic networks, the mobile sink should make a decision about next sink site within an acceptable response time which is dependent on the least sojourn time (i.e., a response time is acceptable when it is much smaller than the least sojourn time). However, the least sojourn time is problem specific, which means it may be unsuitable to define a fixed acceptable response time for all problems (including the existing benchmark problems in our work and other application problems) to discuss the effectiveness of different algorithms. Nevertheless, the response time of different algorithms should be as small as possible to make themselves more practical in applications.

To adopt ACO into dynamic networks, ACO is performed at the beginning of every round and the second step of the outputted solution (i.e., a sequence of sink sites) is regarded as the next step according to the current network status. Since GMRE and HHF are distributed algorithms, they can determine the next step each time the network is updated. The training process of the distinct HHF in dynamic networks is similar to the one in stationary networks and the sensors are updated using the updating rule ℓ in each round during the training process. The GP algorithm also runs 500 generations on twenty training networks and outputs the best H_2 . It is worth mentioning that the initial networks (i.e., the sensor positions) of these twenty training networks and ten testing networks in dynamic networks are the twenty training networks and ten testing networks of stationary network “5”. The training results of the distinct HHF on dynamic networks are shown in Table V. The training time becomes much longer than stationary networks because the mobility of sensors can also do help to energy balance of sensors, which leads to a larger L in simulation.

B. Empirical Results

The AL and ADT are still used as the criteria of the experiments on dynamic networks. The experimental results are listed in Table VI. Generally speaking, the HHF_d and HHF_t have very competitive performances in comparison with

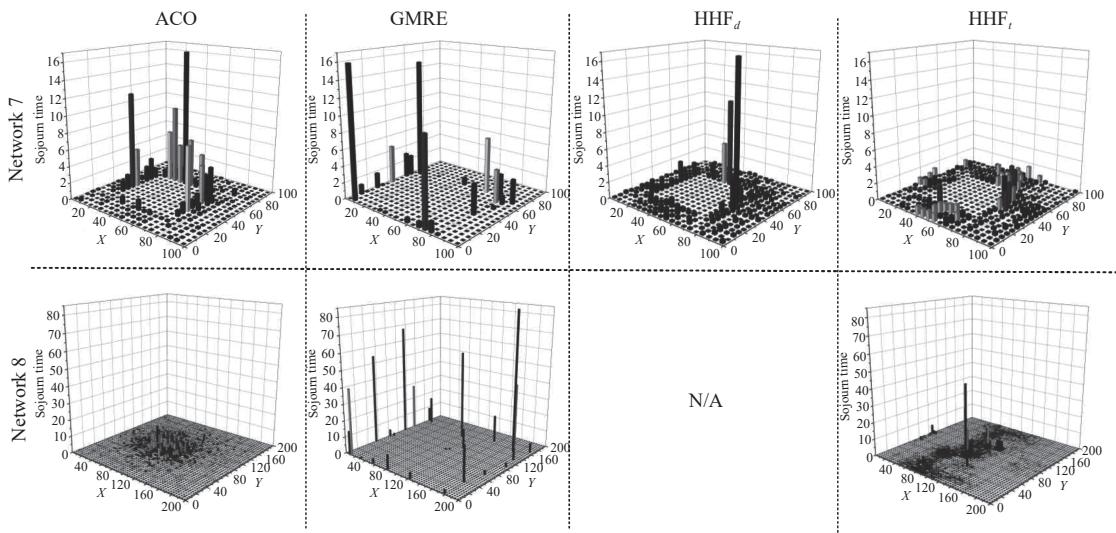
Fig. 8. The average sojourn time in stationary complex network (The result of HHF_d on network 8 is missing because the training time is too long).

TABLE IV
THE PARAMETERS OF DYNAMIC WIRELESS SENSOR NETWORKS

Index	Settings
10	$n = 100, m = 20 \times 20, \Psi(m) = (0, 100) \times (0, 100)$, the network with shaking sensors (i.e., every sensor will move to another place every round with different small step sizes randomly), $E_i = 50 \text{ J}$
11	$n = 100, m = 20 \times 20, \Psi(m) = (0, 100) \times (0, 100)$, the network with migrating sensors (i.e., every sensor will move around the sensing region obeying the clock-wise direction), $E_i = 50 \text{ J}$
12	$n = 100, m = 20 \times 20, \Psi(m) = (0, 100) \times (0, 100)$, the network with sensors aggregating to a moving point (i.e., the target point randomly moves to another place on the sensing region every round. Then the sensors aggregate to a small square whose center is the target point), $E_i = 50 \text{ J}$

TABLE V
THE TRAINING RESULTS OF DISTINCT HHF ON DYNAMIC WIRELESS SENSOR NETWORKS

Index	Heuristic rule	Time (s)
10	$\left(\frac{\nu}{\mu} + \kappa \right) \times \kappa$	307 474
11	$\frac{\Delta}{\nu^2} - \kappa$	271 784
12	$\frac{\Lambda - \kappa - \nu - \lambda}{\kappa}$	482 388

the ACO in most cases according to Table VI. Due to the global searching ability, ACO obtains the longest *AL* among all these three network types. However, the *ADTs* of ACO are much longer than those of distributed algorithms (e.g., nearly 10 000 times of *ADTs* of the distributed algorithms in networks “10” and “12”), which means it is more likely for ACO to become impractical in applications of dynamic networks. It should also be noticed that since the centralized algorithms (e.g., ACO) need a large computation burden, the decision time would be longer when these centralized algorithms are implemented on mobile sinks (e.g., unmanned aerial vehicles) which have less computation power than our experiment platform. GMRE still has a shorter *ADT* but lower

AL performance. The *AL* of GMRE is averagely the 60% to 70% of the one of ACO, which is similar to the one in stationary networks. The distinct HHF performs better than (or at least similar to) the transfer HHF on all testing networks. Especially, in network “12”, the transfer HHF performs much worse than other methods because of the seriously mismatching priori knowledge. By learning the priori knowledge from the training networks, including the updating rule ℓ , the distinct HHF can solve the network “12” with higher *AL* but shorter *ADT*. Based on the results, it can conclude that the competitive performance and short response time mainly benefit from the properly designed training networks and the beforehand training. Without them, the performance of the proposed method would be unsatisfactory, like the HHF_t on network “12”. Also, since the proposed method only utilizes the distinct knowledge of each network type instead of each network case, it is reasonable for HHF_d to have a lightly worse performance than ACO which schedules the sink movement for every network case.

The cumulative distribution function (CDF) of the hop number of packets and the average sojourn time of the sink sites on each network type are also given out. The networks “11” and “12” are taken as the examples. The CDFs of hop number are shown as Fig. 9. In network “11”, the CDF curve of ACO means the most efficient network in the network “11”. HHF_t , HHF_d , and GMRE are ranking two to four respectively, which properly reflects the network efficiency

TABLE VI
THE RESULTS OF DYNAMIC WIRELESS SENSOR NETWORKS

	Index	ACO	GMRE	HHF_d	HHF_t
10	AL	197.8	+	128.9	-
	$ADT(s)$	11.14	-	0.0064	-
11	AL	162.1	\approx	138.6	-
	$ADT(s)$	9.953	-	0.0060	-
12	AL	661.3	+	228.7	+
	$ADT(s)$	21.77	-	0.0061	-

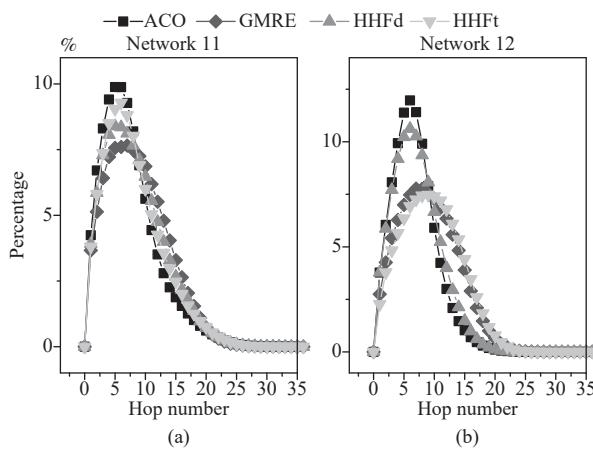


Fig. 9. The CDF of hop number in dynamic networks. (a) the percentage vs. hop number on network “11”; (b) the percentage vs. hop number on network “12”.

and AL s. The curves of network “12” also imply that, with the mismatching prior knowledge, the network efficiency of HHF_t is much lower (i.e., even lower than the one of GMRE). By introducing the correct prior knowledge, the HHF can have a very competitive performance with ACO.

The average sojourn time of these two networks are shown in Fig. 10. For the network “11” whose sensors move around the sensing region in the clock-wise direction, the sensor distribution in every round can be regarded as a new random distribution. Therefore, the average sojourn time in the ACO and the two HHFs are distributed around the central part of the sensing region like the Gaussian distribution as shown in Fig. 10. On the contrary, the sojourn time in GMRE is still distributed on the marginal part.

For the network “12”, there is an important characteristic: too many candidate sink sites will share a same or a similar heuristic value, since the sensors will aggregate densely to a randomly moving point on the sensing region. This leads to two phenomena of the sojourn time. Firstly, in Fig. 10, the sojourn times in GMRE are distributed in the region with small X and small Y . X and Y are the coordinate values of the sensing region. This is because when many sensors aggregate in a small region, many candidate sink sites have a same maximum residual energy and GMRE can not distinguish the difference from them. Therefore, when the algorithm visits all candidate sink sites from small X and Y to large X and Y to select the candidate sink site with the highest heuristic value,

the algorithm will always prefer the first sink site with the highest heuristic value which lies in small X and Y . Secondly, in Fig. 10, the sojourn time in the transfer HHF is quite uniform across the whole sensing region. It means that the H_2 from the transfer HHF acts like a random selection because of the mismatching prior knowledge. These two types of distributions of sojourn time in sink sites imply that the GMRE and the transfer HHF can not effectively tell the quality of the candidate sink sites. On the contrary, benefitting from the global information and the prior knowledge, ACO and the distinct HHF can perform very well in network “12”.

To summary, the proposed HHF can have a very competitive performance with the centralized ACO in terms of AL metric. Furthermore, the outputted H_2 s can have a much shorter response time (i.e., DT) than that of ACO. The above advantages make our method more convenient, flexible, and suitable of real-world applications.

VII. CONCLUSIONS

In this paper, we have proposed a hyper-heuristic framework (HHF) to design heuristic rules which can intelligently schedule the movements of mobile sinks in the wireless sensor networks to maximize the network lifetime. In the proposed framework, training networks and low-level heuristics are designed at the beginning according to prior knowledge. Then, the GP algorithm is developed to automatically construct high-level heuristics based on the training networks and the predefined low-level heuristics. The GP algorithm finally outputs the heuristics with the highest fitness as the results. The outputted heuristic rules can then be applied to new networks in practical applications. To validate the performance of the HHF, a number of stationary networks and dynamic networks have been designed for testing. The experimental results demonstrate that the H_2 s provided by the proposed framework can perform much better than the human-design heuristics in terms of prolonging the network lifetime. Besides, the H_2 s can perform competitively with the evolutionary algorithm ACO in terms of the network lifetime, but in much shorter response time. Generally, the proposed method is more effective and practical than the centralized ACO method and other well-known human-design distributed methods. As for future work, we plan to improve the readability of the found H_2 s by utilizing multi-objective optimization techniques to optimize the quality and complexity of H_2 s at the same time.

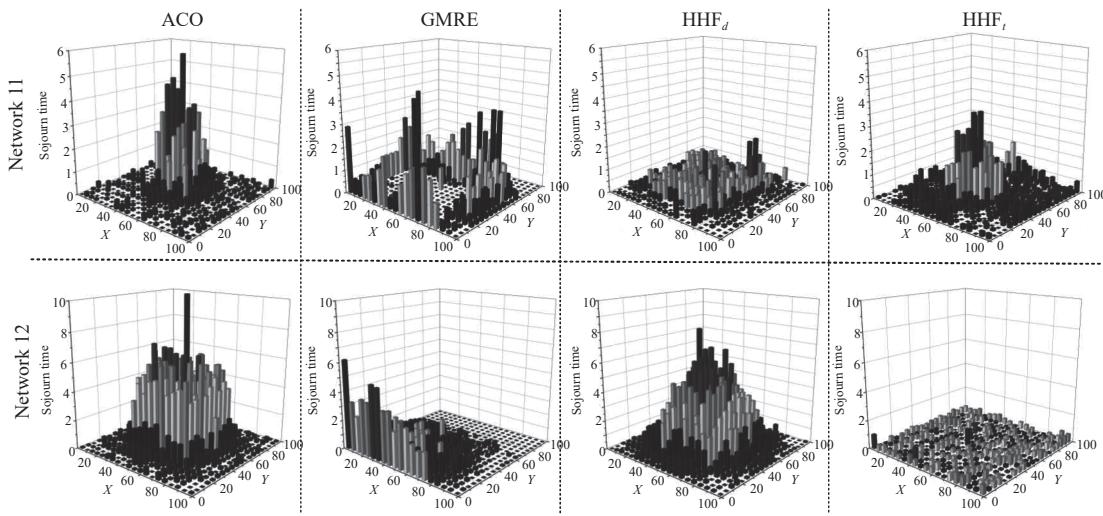


Fig. 10. The sojourn time of dynamic networks.

REFERENCES

- [1] W. Du, Z. K. Xing, M. Li, B. S. He, L. H. C. Chua, and H. Y. Miao, "Sensor placement and measurement of wind for water quality studies in urban reservoirs," *ACM Trans. Sensor Networks*, vol. 11, no. 3, pp. 167–178, 2015.
- [2] W. Du, Z. J. Li, J. C. Liando, and M. Li, "From rateless to distanceless: Enabling sparse sensor network deployment in large areas," *IEEE/ACM Trans. Networking*, vol. 24, no. 4, pp. 2498–2511, 2016.
- [3] A. Alaiad and L. Zhou, "Patients' adoption of wsn-based smart home healthcare systems: an integrated model of facilitators and barriers," *IEEE Trans. Professional Communication*, vol. 60, no. 1, pp. 4–23, 2017.
- [4] Z. G. Sheng, S. Pfersich, A. Eldridge, J. S. Zhou, D. X. Tian, and V. C. M. Leung, "Wireless acoustic sensor networks and edge computing for rapid acoustic monitoring," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 1, pp. 64–74, Jan. 2019.
- [5] H. O. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2688–2710, 2010.
- [6] F. Ye, H. Y. Luo, J. Cheng, S. W. Lu, and L. X. Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks," in *ACM/IEEE Int. Conf. Mobile Computing and Networking*, Atlanta, Georgia, USA: ACM, pp. 148–159, Mar. 2002.
- [7] C. Lin, P. Chou, and C. F. Chou, "HCDD: hierarchical cluster-based data dissemination in wireless sensor networks with mobile sink," in *Int. Conf. Wireless Communications and Mobile Computing*, Vancouver, Canada, 2006, pp. 1189–1194.
- [8] D. Jea, A. Somasundara, and M. Srivastava, "Multiple controlled mobile elements (data mules) for data collection in sensor networks," vol. 3560, Marina del Rey, CA, United States, 2005, pp. 244–257.
- [9] H. Huang, A. V. Savkin, and C. Huang, "I-UMDPC: the improved-unusual message delivery path construction for wireless sensor networks with mobile sinks," *IEEE Internet of Things J.*, vol. 4, no. 5, pp. 1528–1536, Oct. 2017.
- [10] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Trans. Sensor Networks*, vol. 5, no. 1, pp. 5, 2009.
- [11] V. L. Hoesel, T. Nieberg, J. Wu, and P. J. M. Havinga, "Prolonging the lifetime of wireless sensor networks by cross-layer interaction," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 78–86, 2004.
- [12] N. B. Long, H. Tran-Dang, and D. Kim, "Energy-aware real-time routing for large-scale industrial internet of things," *IEEE Internet of Things J.*, vol. 5, no. 3, pp. 2190–2199, Jun. 2018.
- [13] Z. Hong, R. Wang, and X. L. Li, "A clustering-tree topology control based on the energy forecast for heterogeneous wireless sensor networks," *IEEE/CAA J. Autom. Sinica*, vol. 3, no. 1, pp. 68–77, Jan. 2016.
- [14] X. Y. Luo, L. Feng, J. Yan, and X. P. Guan, "Dynamic coverage with wireless sensor and actor networks in underwater environment," *IEEE/CAA J. Autom. Sinica*, vol. 2, no. 3, pp. 274–281, Jul. 2015.
- [15] C. Q. Xia, W. Liu, and Q. X. Deng, "Cost minimization of wireless sensor networks with unlimited-lifetime energy for monitoring oil pipelines," *IEEE/CAA J. Autom. Sinica*, vol. 2, no. 3, pp. 290–295, Jul. 2015.
- [16] E. Fitzgerald, M. Piro, and A. Tomaszewski, "Energy-optimal data aggregation and dissemination for the internet of things," *IEEE Internet of Things J.*, vol. 5, no. 2, pp. 955–969, Apr. 2018.
- [17] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and Z. M. Wang, "A new milp formulation and distributed protocols for wireless sensor networks lifetime maximization," in *Proc. IEEE Int. Conf. Communications*, vol. 8, Jun. 2006, pp. 3517–3524.
- [18] J. Luo and J. Hubaux, "Joint sink mobility and routing to maximize the lifetime of wireless sensor networks: the case of constrained mobility," *IEEE/ACM Trans. Networking*, vol. 18, no. 3, pp. 871–884, 2010.
- [19] J. Zhong and J. Zhang, "Ant colony optimization algorithm for lifetime maximization in wireless sensor network with mobile sink," in *Proc. 14th Int. Conf. Genetic and Evolutionary Computation*, Philadelphia, PA, United States, 2012, pp. 1199–1204.
- [20] Y. Li, Z. X. Huang, J. H. Zhong, and L. Feng, "Genetic programming for lifetime maximization in wireless sensor networks with a mobile sink," in *Proc. Asia-pacific Conf. Simulated Evolution and Learning*, Springer, Cham, 2017, pp. 774–785.
- [21] A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Srivastava, and D. Estrin, "Intelligent fluid infrastructure for embedded networks," in *Proc. 2nd ACM/SIGMOBILE Int. Conf. Mobile Systems, Applications, and Services, MobiSys 2004*, 2004, pp. 111–124.
- [22] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines," in *Proc. 25th IEEE Int. RealTime Systems Symposium*, Dec 2004, pp. 296–305.
- [23] W. Wang, V. Srinivasan, and K. Chua, "Using mobile relays to prolong the lifetime of wireless sensor networks," in *Proc. Annual Int. Conf. Mobile Computing and Networking, MOBICOM*, Cologne, Germany, 2005, pp. 270–283.
- [24] S. R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, "Energy efficient schemes for wireless sensor networks with multiple mobile base stations," in *Proc. GLOBECOM '03. IEEE Global Telecommunications Conf.*, vol. 1, Dec 2003, pp. 377–381.
- [25] Z. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, "Exploiting sink mobility for maximizing sensor networks lifetime," in *Proc. 38th Annual Hawaii Int. Conf. System Sciences*, Jan 2005, pp. 287a–287a.
- [26] J. Luo and J. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *Proc. IEEE 24th Annual Joint Conf. IEEE Computer and Communications Societies.*, vol. 3, Mar. 2005, pp. 1735–1746.
- [27] X. J. Deng, B. Wang, W. Y. Liu, and L. T. Yang, "Sensor scheduling for multi-modal confident information coverage in sensor networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 26, no. 3, pp. 902–913, Mar. 2015.
- [28] W. F. Liang and J. Luo, "Network lifetime maximization in sensor networks with multiple mobile sinks," in *Proc. IEEE 36th Conf. Local*

- Computer Networks*, Oct. 2011, pp. 350–357.
- [29] J. N. Al-Karaki, R. Ul-Mustafa, and A. E. Kamal, “Data aggregation and routing in wireless sensor networks: optimal and heuristic algorithms,” *Computer Networks*, vol. 53, no. 7, pp. 945–960, 2009.
- [30] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and Z. M. Wang, “Controlled sink mobility for prolonging wireless sensor networks lifetime,” *Wireless Networks*, vol. 14, no. 6, pp. 831–858, 2008.
- [31] Y. S. Yun, Y. Xia, B. Behdani, and J. C. Smith, “Distributed algorithm for lifetime maximization in a delay-tolerant wireless sensor network with a mobile sink,” *IEEE Trans. Mobile Computing*, vol. 12, no. 10, pp. 1920–1930, Oct. 2013.
- [32] F. Carrabs, R. Cerulli, C. D’Ambrosio, and A. Raiconi, “A hybrid exact approach for maximizing lifetime in sensor networks with complete and partial coverage constraints,” *J. Network and Computer Applications*, vol. 58, pp. 12–22, 2015.
- [33] M. Xie and H. C. Shi, “Ant-colony optimization based in-network data aggregation in wireless sensor networks,” in *Proc. 12th Int. Symposium on Pervasive Systems, Algorithms and Networks*, Dec. 2012, pp. 77–83.
- [34] D. P. Kumar, T. Amgoth, and C. S. R. Annavarapu, “Aco-based mobile sink path determination for wireless sensor networks under non-uniform data constraints,” *Applied Soft Computing*, vol. 69, pp. 528–540, 2018.
- [35] N. Primeau, R. Falcon, R. Abielmona, and E. M. Petriu, “A review of computational intelligence techniques in wireless sensor and actuator networks,” *IEEE Communications Surveys and Tutorials*, vol. 20, no. 4, pp. 2822–2854, 2018.
- [36] C. W. Tsai, T. P. Hong, and G. N. Shiu, “Metaheuristics for the lifetime of wsn: a review,” *IEEE Sensors J.*, vol. 16, no. 9, pp. 2812–2831, 2016.
- [37] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Öezcan, and R. Qu, “Hyper-heuristics: a survey of the state of the art,” *J. Operational Research Society*, vol. 64, no. 12, pp. 1695–1724, 2013.
- [38] F. F. Zhang, Y. Mei, and M. J. Zhang, “Evolving dispatching rules for multi-objective dynamic flexible job shop scheduling via genetic programming hyper-heuristics,” in *Proc. IEEE Congress on Evolutionary Computation*, 2019.
- [39] S. Nguyen, Y. Mei, and M. J. Zhang, “Genetic programming for production scheduling: a survey with a unified framework,” *Complex and Intelligent Systems*, vol. 3, no. 1, pp. 41–66, 2017.
- [40] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, “Automatic design of a hyper-heuristic framework with gene expression programming for combinatorial optimization problems,” *IEEE Trans. Evolutionary Computation*, vol. 19, no. 3, pp. 309–325, 2015.
- [41] M. A. Ardeh, Y. Mei, and M. J. Zhang, “Transfer learning in genetic programming hyper-heuristic for solving uncertain capacitated arc routing problem,” in *Proc. IEEE Congress on Evolutionary Computation*, 2019.
- [42] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, “A dynamic multiarmed bandit-gene expression programming hyper-heuristic for combinatorial optimization problems,” *IEEE Trans. Cybernetics*, vol. 45, no. 2, pp. 217–228, 2015.
- [43] J. Park, Y. Mei, S. Nguyen, G. Chen, and M. J. Zhang, “An investigation of ensemble combination schemes for genetic programming based hyperheuristic approaches to dynamic job shop scheduling,” *Applied Soft Computing*, vol. 63, pp. 72–86, 2018.
- [44] J. H. Chang and L. Tassiulas, “Maximum lifetime routing in wireless sensor networks,” *IEEE/ACM Trans. Networking*, vol. 12, no. 4, pp. 609–619, 2004.
- [45] J. H. Zhong, Y. S. Ong, and W. T. Cai, “Self-learning gene expression programming,” *IEEE Trans. Evolutionary Computation*, vol. 20, no. 1, pp. 65–80, Feb. 2016.
- [46] L. F. Ma, Z. D. Wang, Q.-L. Han, and H.-K. Lam, “Variance-constrained distributed filtering for time-varying systems with multiplicative noises and deception attacks over sensor networks,” *IEEE Sensors J.*, vol. 17, no. 7, pp. 2279–2288, 2017.
- [47] L. F. Ma, Z. D. Wang, H.-K. Lam, and N. Kyriakoulis, “Distributed eventbased set-membership filtering for a class of nonlinear systems with sensor saturations over sensor networks,” *IEEE Trans. Cybernetics*, vol. 47, no. 11, pp. 3772–3783, 2017.
- [48] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proc. 33rd Annual Hawaii Int. Conf. System Sciences*, Maui, USA: IEEE, vol. 2, Jan. 2000.
- [49] Y. Shi and Y. T. Hou, “Theoretical results on base station movement problem for sensor network,” in *Proc. IEEE INFOCOM 27th Conf. Computer Communications*, Apr. 2008, pp. 1–5.



Jinghui Zhong received the Ph.D. degree from the School of Information Science and Technology, Sun Yat-Sen University, China, in 2012. He is currently an Associate Professor at the School of Computer Science and Engineering, South China University of Technology, China. During 2013 to 2016, he worked as a Postdoctoral Research Fellow at the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include evolutionary computation (e.g., genetic programming and differential evolution), machine learning, and agent-based modeling.



Zhixing Huang received the B.E. degree from the School of Computer Science and Engineering, South China University of Technology, China, in 2018. He is currently an M.S. student at the School of Computer Science and Engineering, South China University of Technology, China. His research interests include evolutionary computation (e.g., genetic programming and its applications), artificial intelligence, program synthesis, and genetic improvement.



Liang Feng received the Ph.D. degree from the School of Computer Engineering, Nanyang Technological University, Singapore, in 2014. He was a Postdoctoral Research Fellow at the Computational Intelligence Graduate Laboratory, Nanyang Technological University, Singapore. He is currently an Assistant Professor at the College of Computer Science, Chongqing University, China. His research interests include computational and artificial intelligence, memetic computing, big data optimization and learning, as well as transfer learning.



Wan Du is currently an Assistant Professor at the University of California, Merced. Dr. Du worked as a Research Fellow at the School of Computer Science and Engineering, Nanyang Technological University, Singapore from 2012 to 2017. He received the B.E. and M.S. degrees in electrical engineering from Beihang University, China and the Ph.D. degree in electronics from the University of Lyon (cole centrale de Lyon), France. His research interests include the internet of things, cyber-physical system, distributed networking systems, and mobile systems.



Ying Li received the bachelor degree from South China University of Technology, China, in 2018. She is currently a Product Manager in Binxun Technology Co.,Ltd, Shenzhen, China. Her research interests include evolutionary computation and its application such as using genetic programming to schedule the wireless sensor network.