Metaheuristics for the Lifetime of WSN: A Review

Chun-Wei Tsai, Tzung-Pei Hong, and Guo-Neng Shiu

***Abstract*— The importance and the possibilities of wireless sensor networks (WSNs) are now quite clear, because they can be found in all kinds of applications, from those in our daily life to those in the military. However, the limited energy of sensors, i.e., the lifetime problems of WSN, has attracted many researchers from different disciplines. Several recent studies showed that metaheuristic algorithms provide promising solutions to the lifetime problems. This paper begins with a brief review of the lifetime problems and the basic ideas of metaheuristic algorithms. Then, the detailed descriptions of metaheuristic algorithms for solving the lifetime problems from the perspectives of problems and algorithms are given. Some simple examples for illustrating how metaheuristic algorithms can be used to solve the lifetime problems and their performance are given. Several important open and possible research issues are discussed to provide the future research trends of this area.**

***Index Terms*— Wireless sensor networks, lifetime, and metaheuristic algorithm.**

I. Introduction

WITH THE advance of information technologies, everything in our life has been digitized gradually because digital data are much easier to transmit, store, analyze, and so on than analog data. In order to obtain more information for making a better decision, analyzing data collected by appliances and systems has become a critical issue for every kind of organization. But the cost of data collection and analysis is usually expensive. That is why over the years, we are still looking for a suitable appliance, which is small, cheap, and energy efficiency, to collect the data we need. Wireless sensor network (WSN) provides a possible solution for efficiently collecting data located anywhere, which has been under the public concern for a long time. Several recent studies [1]–[7], such as temperature and humidity monitoring, home safety, healthcare, and military defense, have witnessed the successful applications of WSN.

WSN is generally anticipated to have unlimited possibilities for a number of applications because of the advantages it owns. However, different from traditional appliances, the computational capacity, memory, and storage of sensors are usually limited [1]. Another important limitation is that most sensors are loaded with non-rechargeable battery, which means that the energy (lifetime) of such a network is limited. These limitations led to several research issues that are defined as optimization problems aimed at finding useful design strategies, such as sensor deployment problem [8], routing problem [9], and clustering problem [10]. That is why a numerous efficient mechanisms [11], [12] presented for mitigating the impact of these restrictions on WSN are needed for providing better services. In [3], Tilak et al. pointed out that the energy-efficiency, latency, accuracy, fault-tolerance, and scalability can be used to measure the performance of WSN. The research issues of coverage, connectivity, and node availability can be put together and reduced to an essential problem—*network lifetime* [11].

Most lifetime problems of WSN are as difficult and complex as traditional optimization problems, such as traveling salesman problem, which mean that any efficient mechanism would be useful in prolonging the lifetime of such a network. To better understand the lifetime research issues, a comprehensive overview on the lifetime of WSN from different perspectives of definitions, methods, and example scenarios was presented by Dietrich and Dressler [11]. According to our observation, due to the low computational capacity and the real-time requirement, deterministic algorithms[1] and traditional metaheuristic algorithms [13], [14] may not be able to provide solutions that are applicable to the environment of WSN; rather, lightweight and fast algorithms are much more likely to meet the requirements of such an environment. Several improved metaheuristics [12], [15] have been presented for solving the optimal problem of WSN in recent years. Although lifetime algorithms have been discussed and analyzed in previous studies [15]–[18], an integrated review on metaheuristic algorithms for the lifetime optimization problems of WSN is still needed to provide a roadmap to the researchers working on this field.

This paper attempts not only to give a systematic description of the lifetime problems of WSN but also to discuss in detail how metaheuristic algorithms can be applied to these problems. This review looks at this problem from two different perspectives—*the perspective of problems* and *the perspective of algorithms*. From the perspective of problems, the discussions are focused on which algorithm can be used for solving a particular lifetime problem and how to use it.

Manuscript received May 17, 2015; revised January 16, 2016; accepted January 17, 2016. Date of publication January 28, 2016; date of current version March 16, 2016. This work was supported by the Ministry of Science and Technology, Taiwan, under Contract MOST 104-2221-E-197-005 and Contract MOST 104-2221-E-268-002. The associate editor coordinating the review of this paper and approving it for publication was Prof. Elena Gaura.

C.-W. Tsai is with the Department of Computer Science and Information Engineering, National Ilan University, Yilan City 26047, Taiwan (e-mail: cwtsai0807@gmail.com).

T.-P. Hong is with the Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 81148, Taiwan, and also with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan (e-mail: tphong@nuk.edu.tw).

G.-N. Shiu is with the Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 81148, Taiwan (e-mail: m0935103@mail.nuk.edu.tw).

Digital Object Identifier 10.1109/JSEN.2016.2523061

[1] In this paper, deterministic algorithms indicate algorithms that will always produce the same solution for the optimal problem in question, such as making a decision by using some particular rules.

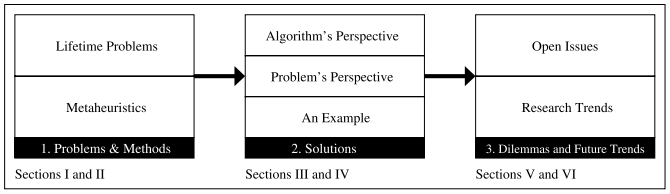

Fig. 1. Roadmap of this paper.

From the perspective of algorithms, the discussions are focused on the representation (encoding) of solutions, the basic idea of operators of different metaheuristics, and the improved methods. In other words, this study is aimed at providing a brief review of metaheuristic algorithms for the lifetime problems of WSN to the researchers who are focusing on prolonging the lifetime of a sensor network to easily understand the relevant issues.

Fig. 1 shows the roadmap of this survey, and the remainder of the paper is organized as follows. Section II begins with a brief introduction to the lifetime problems of WSN, followed by a brief review of metaheuristics. The main purpose of Section III is to provide a comprehensive survey of metaheuristic algorithms for the lifetime problems of WSN. Some simple examples are given in Section IV to illustrate the capabilities of metaheuristic algorithms for the lifetime problems of WSN. Further discussions on metaheuristic algorithms for WSN are given in Section V, which contain advantages and disadvantages of these metaheuristic algorithms and the open issues they face. The possible research trends are discussed in Section VI.

II. Related Works

In this section, the discussion will be divided into two parts: (1) the lifetime problems of WSN which include the definition of *problems* discussed in this paper and (2) a brief review of metaheuristic algorithms that provide *solutions* to the problems. To simplify the discussion, the following notations are used throughout the rest of the paper.

- S set of sensor nodes, i.e., $S = \{s_1, s_2, \ldots, s_n\}$, where s_i is the i-th sensor node and n the number of sensor nodes.
- C set of covers, i.e., $C = \{c_1, c_2, \ldots, c_k\}$, where c_j is the j-th cover and k the number of covers. Moreover, C also denotes the set of clusters when the lifetime problems are considered as clustering problems.
- T set of targets, i.e., $T = \{t_1, t_2, \ldots, t_m\}$, where t_i is the i-th target of WSN and m the number of targets.
- p set of solutions (population) of metaheuristic algorithms, i.e., $p = \{p_1, p_2, \ldots, p_\theta\}$, where p_i is the i-th solution and θ the number of solutions. Note that $\theta = 1$ when the metaheuristic algorithm used is a single-solution-based algorithm; $\theta > 1$ when the metaheuristic algorithm used is a population-based algorithm. Also, the i-th solution at iteration t is denoted by $p_i^t = \{p_{i,1}^t, p_{i,2}^t, \ldots, p_{i,\psi}^t\}$, where ψ is the number of sub-solutions. The solutions can be regarded as the chromosomes of genetic algorithm (GA), ants of ant colony optimization (ACO), and particles of particle swarm optimization (PSO) as far as this paper is concerned.
- ℓ maximum number of iterations or generations.

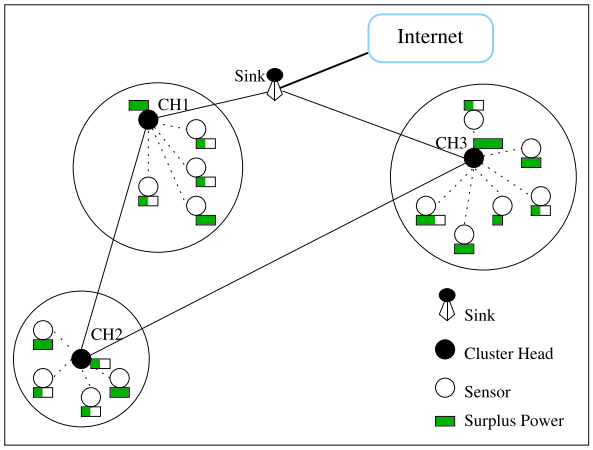

Fig. 2. A simple example of WSN.

A. Lifetime Problems of WSN

In [11], Dietrich and Dressler gave a comprehensive survey on the lifetime problems of WSN which provides a cornerstone for finding better solutions to these problems. Fig. 2 provides a simple example of WSN, which contains sinks, cluster heads, sensors, surplus power, etc. The lifetime problems of WSN are complex because they are composed of sensors, systems, and software. It is not hard to imagine that a number of factors have to be taken into account to prolong the lifetime of a sensor network, such as mobility, heterogeneity, and quality of service.

For example, a well-known definition is "the time until the first sensor is drained of its energy," which is used to describe the lifetime problem of a sensor network the aim of which is to find a better solution to prolong the lifetime of a WSN. Another example is that some recent studies [19], [20] have attempted to use "the time until the first cluster head is drained of its energy" to define and describe the lifetime problems of WSN to make them as close to the reality as possible because sensors are grouped into clusters to transit the data. In this case, even though some sensors in a cluster run out of energy, the sensor network can still work as usual. In a later study [21], Asorey-Cacheda et al. pointed out that real-time and network lifetime constraints are often used in highly-reliable applications of WSN, as shown in Fig. 3. For these applications, these two constraints also involve soft and hard deadline requirement issues. According to the assumptions, requirements, and definitions of network lifetime problems from the studies of [4], [11], [21], and [22], the following sections will briefly discuss the network lifetime problems from different perspectives.

1) Number of Alive Nodes Problem: The network lifetime problems based on the number of alive nodes typically can be divided into n-of-n, k-of-n, and m-in-k-of-n lifetimes [11], [20], [23], [24]. The definition of n-of-n lifetime says that the network lifetime T_n^n ends when the first node

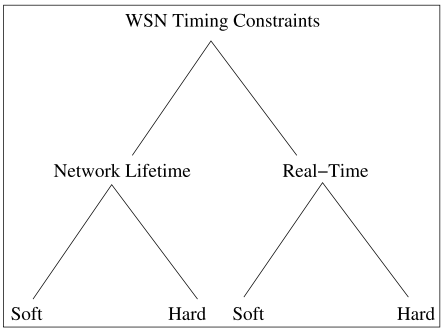

Fig. 3. Time constraints on WSN [21].

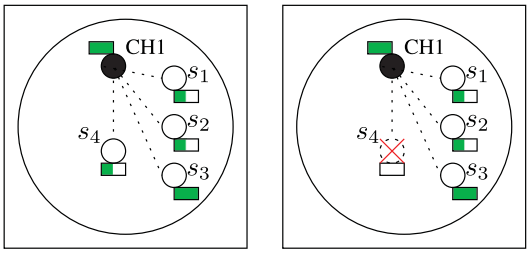

Fig. 4. Examples illustrating the number of alive nodes problem. (a) n-of-n. (b) k-of-n.

fails (runs out of energy). That is, T_n^n is defined as

$$T_n^n = \min_{v \in V} T_v, \tag{1}$$

where T_v denotes the lifetime of node v and V denotes the set of nodes, i.e., all the sensors in the sensor network in question. As shown in Fig. 4(a), the n-of-n lifetime problem attempts to prolong the "time" T_n^n of all the sensors so that they are able to work until one of the nodes ran out of energy, say, sensor s_4. In other words, it is the time T_n^n that is used to measure the performance of a sensor network.

The definition of k-of-n lifetime says that the network lifetime ends when a certain percentage of the nodes (in terms of a predefined threshold) run out of energy. As shown in Fig. 4(b), the k-of-n lifetime problem is in fact very similar to the n-of-n lifetime problem except that it only requires that k or more out of n sensor nodes be alive; otherwise, the remaining nodes will not be able to transmit any data to the sink node [11]. Thus, the k-of-n lifetime T_n^k is defined as

$$T_n^k = \min_{v \in V} T_v. \tag{2}$$

A more complete consideration on the lifetime of WSN is the m-in-k-of-n lifetime [20], which divides all the nodes in a WSN into two groups: critical and non-critical. Based on this definition, the network will survive if all the m critical nodes and at least k nodes are alive, where $1 \leq m \leq k \leq n$. In other words, the WSN will fail if either $n-k+1$ non-critical nodes or any critical node runs out of energy.

2) Cluster Head Election Problem: Similar to the alive nodes problem, one of the main concerns of the cluster head election problem is also seeking an efficient mechanism to extend the lifetime of WSN. In [11], Dietrich and Dressler pointed out that the cluster head election problem is a variant

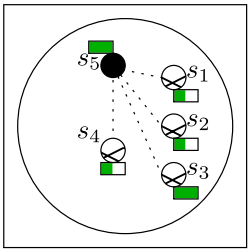

Fig. 5. Examples illustrating the cluster head election problem.

of the n-of-n lifetime problem. In addition, some algorithms also take into consideration the issues of data routing among sensors and cluster head (CH). As shown in Fig. 5, the main objective of the cluster head election problem is to select cluster heads from all the sensors of a WSN in such a way that the lifetime of the WSN is maximized. Although every sensor node can be elected as the cluster head, because the cluster heads (s_5 in this case) are responsible for collecting data from their neighboring nodes and then forwarding the aggregated data to the base station (sink node), their energy consumption is usually higher than sensors used only for monitoring the targets. However, if the cluster head election mechanism always selects the sensor that has the highest surplus power, the lifetime of a WSN will depend on the lifetime of that particular sensor.

One of the most well-known cluster head election algorithms aimed at minimizing the global energy usage of a WSN is the low-energy adaptive clustering hierarchy (LEACH) [10]. The basic idea of LEACH is to achieve load balancing among sensors to prolong the lifetime of a network. Each sensor can be elected as the cluster head based on a probability model. This mechanism can be used to avoid overloading and thus over-consuming the energy of some particular nodes. In this way, the lifetime of a network can be prolonged. To evaluate the clustering result, a general metric [10], [25] for measuring the energy expended in transmitting and receiving an l-bit data over a distance d, denoted $T(l,d)$ and $R(l)$, is defined as

$$T(l,d) = T_e \times l + \epsilon \times l \times d^2, \tag{3}$$

$$R(l) = T_e \times l, \tag{4}$$

where T_e is the energy dissipated per bit to run the transmitter or receiver circuitry; ϵ the energy dissipation of the amplifier of the transmitter. The clustering algorithm will select the number of clusters k that is appropriate for all the sensors in a WSN based on Eqs. (3) and (4). Moreover, some recent studies tried to narrow down the gap between the models and real environments of a WSN, by adding more constraints to the models. An example is given in [26], which takes into consideration the average of sensor energy (excluding CH) and density at the same time.

3) Deployment Coverage Problem: If the main concern is in finding a set of "locations" for the sensors so that they can cover the region of interest, the two typical considerations in the deployment of the sensors are: maximizing the "sensing coverage" and minimizing the "power consumption" for the transmission of data. For this kind of problem, a number of definitions for the coverage metrics of sensors

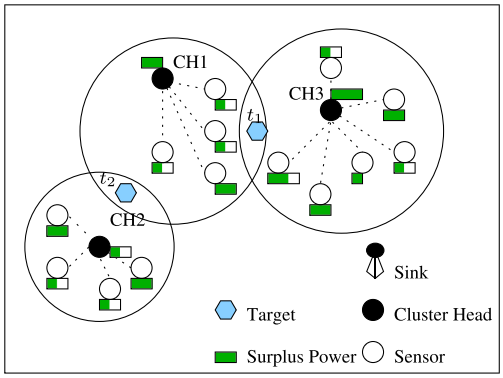

Fig. 6. Examples illustrating the set-cover problem.

and energy consumption functions are used in different researches [11], [27]–[29]. For example, the deployment problem considers the sensing coverage and energy consumption at the same time, by using the Dijkstra's algorithm and simulated annealing [27]. Another example [28] also considers the coverage and energy factors for the sensor deployment problem at the same time, as follows:

$$f(s) = \frac{100 \times P_c(s)^2}{P_T^2 \times H}, \tag{5}$$

where s is the set of sensors; $P_c(s)$ the covered points of s; P_T the total number of points; H the highest energy load.

4) Set-Cover Problem: If the main concern is in that a finite set of targets has to be covered (monitored), a number of recent studies [30]–[32] used the set k-cover problem to formulate the problem of finding the maximum number of "covers" for the longest lifetime of a WSN. Besides, the search algorithm has to determine the membership of each sensor, i.e., to which cluster it belongs. The basic definition of this problem [32] is as follows: Given a collection $S = \{s_1, s_2, \ldots, s_n\}$ of subsets of a finite set $T = \{t_1, t_2, \ldots, t_m\}$, find the maximum number of disjoint covers $c_1, c_2, \ldots, c_k \subseteq S$; i.e., $c_i \cap c_j = \emptyset$, $i \neq j$. In this definition, a *cover* indicates a certain number of sensors (a subset of S) chosen for monitoring all the targets in such a network. The definition of this lifetime problem is to artfully define n sensors S that are deployed to monitor m targets T, where every target t_j has to be covered by at least one element of c_i. Because every cover set can be employed to monitor all the targets in this network, if the sensor network only uses a set of sensors that belong to the same cover at a time, the network lifetime can be prolonged by increasing the number of covers k; i.e., the network lifetime is proportional to the number of coves k. Fig. 6 shows a case in which two targets need to be covered.

5) Data Routing Problem: The main concern of data routing problem is to find the best way (or path) to transmit the data from the source node (e.g., a sensor node) to the destination node (e.g., another sensor node, cluster head, or sink node). Thus, the number of hops, sum of the transmitter power, optimal traffic distribution, and shortest routing path can all be used as the measure for the routing problem of WSN. As shown in Fig. 7, the routing path "CH2 to CH1 to sink" is better than the routing path "CH2 to CH3 to sink," if the source node is CH2, the destination node is sink, and the main concern is the distance of the routing path. For the routing problem, the energy consumption is also a critical issue because an efficient routing path usually also implies energy-efficiency [33], [34]. More precisely, an efficient solution (i.e., any "good" routing path) can be used to send the data from the source node to the sink node for data gathering and aggregation to further save the energy consumption of a WSN [35].

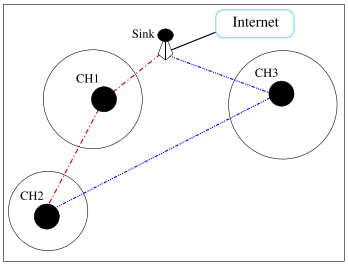

Fig. 7. An example of the data routing problem.

Algorithm 1 Metaheuristic Algorithm

```
Create the initial solution s;
while (the termination criterion is not met) do
    Transition();
    Evaluation();
    Determination();
end
```

B. Metaheuristic Algorithms

Now, we will turn our discussion to solutions to these problems based on metaheuristics [13], [14]. Before we go any further, it should be noted that the distinguishing characteristic of metaheuristic algorithms is the use of the so-called *transition*, *evaluation*, and *determination* operators to "guess" applicable solutions in the solution space [36]. Because metaheuristics are widely used in many problem domains, a number of metaheuristic algorithms have been presented. A useful taxonomy of these metaheuristic algorithms is to categorize them into two groups [13]: single-solution-based algorithm (SSBA) and population-based algorithm (PBA). For instance, simulated annealing (SA) [37], [38] and tabu search (TS) [39], [40] are both single-solution-based algorithm while genetic algorithm (GA) [41], [42], ant colony optimization (ACO) [43]–[45], and particle swarm optimization (PSO) [46], [47] are all population-based algorithm.

As shown in Algorithm 1 [36], the main operators of metaheuristics—transition, evaluation, and determination—will usually be performed at each iteration. Table I briefly explains how they are performed for different metaheuristics. In Table I, p_i^t denotes the i-th solution at iteration t; p_i^{t+1} the i-th solution at iteration $t+1$; p the set of solutions (i.e., the population); p_i^t and p_j^t, two different solutions (i.e., we assume $i \neq j$); and v_i^t and v_i^{t+1} velocities of p_i^t and p_i^{t+1}.

TABLE I
OPERATORS OF METAHEURISTIC ALGORITHMS

Algorithm	Transition	Evaluation	Determination
SA	NeighborSelection(p^t) TemperatureUpdate()	NeighborSelection(p^t)	ObjectiveValueComparison(p^t, p^{t+1})
TS	NonTabu-NeighborSelection(p^t) TabuListUpdate(p^t)	NonTabu-NeighborSelection(p^t)	ObjectiveValueComparison(p^t, p^{t+1})
GA	Crossover(p_i^t, p_j^t) Mutation(p_i^t)	FitnessFunction(p_i^t)	Selection(p^t)
ACO	PheromoneUpdate(p^t)		
	SolutionConstruction(p^t, τ), PheromoneUpdate(p^t), and LocalSearch(p^t)		
PSO	v_i^{t+1} = VelocityUpdate(p_i^t, v_i^t) p_i^{t+1} = PositionUpdate(p_i^t, v_i^{t+1})	LocalBestUpdate(p_i^{t+1}) GlobalBestUpdate(p^{t+1})	

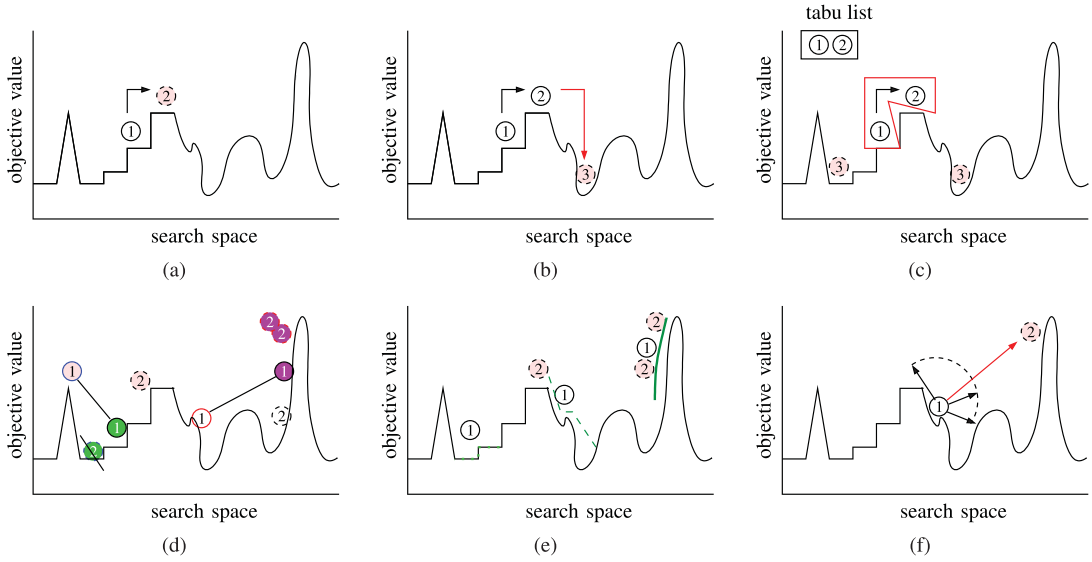

Fig. 8. Examples illustrating strategies used by metaheuristic algorithms in the search of solutions. (a) Hill Climbing. (b) Simulated Annealing. (c) Tabu Search. (d) Genetic Algorithm. (e) Ant Colony Optimization. (f) Particle Swarm Optimization.

From this table, it is easy to understand that the operators of a metaheuristic algorithm may play multiple roles during its convergence process. For instance, NeighborSelection() of SA plays the role of transiting and evaluating the solution. Of course, sometimes, the three operators may need more than one sub-operator to accomplish their mission. For instance, the transition operator of GA involves two sub-operators, namely, crossover() and mutation().

Fig. 8 gives examples to illustrate how metaheuristics work; that is, how metaheuristics search for the solutions and what strategies they take during the convergence process. In addition, Figs. 8(a), (b), and (c) are for the single-solution-based algorithms while Figs. 8(d), (e), and (f) are for the population-based algorithms.

Moreover, as depicted in Fig. 8(a), hill climbing (HC) is a greedy search algorithm, for it differs from other metaheuristics in that it has no mechanism to escape from local optimum. As shown in Figs. 8(b) and (c), SA and TS use different strategies to solve the problem of getting stuck at local optimum. First, the determination operator of SA has a chance to choose a solution that is worse than the current solution. That is why SA chooses point 3 of Fig. 8(b). The idea behind this is simple. If SA chooses point 3, then it has a chance to escape from the local optimum (point 2 in this case) so as to find a better solution on the right side of the figure. The main advantage of SA is that it is very easy to implement, and in the ideal case, it has a chance to find the optimum solution. However, the main disadvantage of SA is that it cannot find the global optimum or approach it quickly because it searches for a single direction at a time. Unlike SA, TS uses the so-called tabu list to restrict searching for the solutions that have been recently searched as shown in Fig. 8(c). As a result, points 1 and 2 will not be searched again for a while (depending on the size of the tabu list). Although the main advantage of TS is restricting the search of the same solutions in the near future to avoid falling into local optimum, the disadvantages are how to set the length of the tabu list and how to create the neighbors.

As shown in Fig. 8(d), GA uses multiple search directions at each iteration. For instance, all points marked 1, be it shaded or not, represent chromosomes at iteration 1. After mating (crossover and mutation) and selection, all points marked 2

are created as the search directions of the next iteration, i.e., iteration 2. Note that at iteration 2, the point on the bottom left does not survive the selection while the points on the upper right do because the latter fit better than the former in terms of the fitness value. The main advantage of GA is its global search ability because it employs multiple search directions at each iteration to enable exchange of the search information. Thus, GA can find solutions close to the optimum very quickly. However, the main disadvantages of GA are: (1) the quality of the final solution is normally poor because the fine-tuning ability of GA is worse than SA, TS, and other local search methods; and (2) the representations of the solutions and most of the operators are problem specific, thus needing to be redesigned for the new problem because simple GA is designed for the generality; thus, it is not suitable for many problems. For these reasons, we need to use domain knowledge to redesign the operators of GA to enhance its performance.

Like the TS, ACO needs to keep track of the information accumulated on the convergence process, as shown in Fig. 8(e). Unlike the TS, it keeps the search knowledge in the so-called pheromone table; thus, ants can find regions that have been visited by most of the ants and that may provide better results than other regions. The main advantage of ACO is that it combines the global and local search methods on the convergence process. For this reason, it can provide better results than SA, TS, and simple GA by using the same number of iterations. However, the main disadvantages are: (1) ACO is usually computationally more expensive than SA, TS, and GA, and (2) because the original design of ACO is for combinatorial optimization, for continuous optimization, the representation of the solutions and the search strategies need to be redesigned. As shown in Fig. 8(f), the search directions of PSO will be influenced by the global best, local best, and personal trajectory. According to the observation of several studies, the main advantage of PSO is that it can find an approximate solution quickly. However, the main disadvantages are: (1) PSO suffers from the premature problem, and (2) because PSO is originally designed for function optimization, it is not suitable for combinatorial optimization. That is why for the combinatorial optimization problem, the representation of PSO needs to be transformed by using, say, random key method.

C. Discussion

In summary, the lifetime of WSN is affected by several factors. Among others, these factors are how to determine the alive nodes, coverage of sensor, cluster head, connectivity of network, and quality of service [11]. Because WSN, until now, still leaves open many research issues and because it is part of several real world approaches, this area is rising and flourishing in these years. Most of the lifetime problems facing by this new environment (WSN) are complex and difficult,[2] in the sense that they would take an extraordinary amount of time to find the optimal solution. Thus, efficient and effective algorithms are needed to improve the performance of the network so as to prolong the lifetime of the network. Among them, the algorithms proposed by early studies are simple and easy to implement, such as LEACH [10]. Since the performance of rule-based and deterministic algorithms for these lifetime problems are simply not good enough, many researchers in this research domain have kept looking for more efficient and effective algorithms. An alternative solution is by using *metaheuristic algorithms* to solve these complex problems because the main characteristic of these algorithms is that it can find an approximate solution from the large solution space within a reasonable time.

To apply metaheuristics to the lifetime problem of WSN, the very first thing is to understand the assumptions, the environment parameters, and the domain knowledge of the lifetime problem. Because most metaheuristics can be used for the optimization problem, modeling the lifetime problem as an optimization problem based on the assumptions, the environment parameters, and the domain knowledge can help us understand how to use metaheuristics to find a better solution to the lifetime problem to further prolong the lifetime of a WSN. For example, the goal of the set-cover problem is to maximize the number of disjoint cover sets for the targets. This implies that the number of cover sets can be regarded as the objective value that can also be used to evaluate the possible solution of a metaheuristic algorithm. In addition to defining the objective function of a metaheuristic algorithm, the solution representation is another critical work that depends on how the solution of the lifetime problem is encoded. For the set-cover problem, if the solution is encoded as an integer string; i.e., $p_i^t = \{p_{i,1}^t, p_{i,2}^t, \ldots, p_{i,\psi}^t\}$, then $p_{i,j}^t$ can be used to represent which target is covered by the j-th sensor. For instance, $p_{i,1}^7 = 3$ can be regarded as indicating that the third target is covered by the first sensor at iteration seven.

Once the solution representation is determined, the transition operator can then be chosen because this operator typically depends on the optimization problem in question and the way the solutions are represented. For example, the one-point crossover of GA is not suitable for the solution representation of routing path because it may break the solution structure found by GA. A similar situation can be easily found in PSO because it is designed in such a way that velocity is used to exchange the search information between solutions. The transition operator of PSO must be modified when the solution representation is changed to an integer string. Since the modification of the determination operator of metaheuristic algorithm is usually based on the domain knowledge and assumption of the lifetime problem, it has to take into account the objective function of the lifetime problem in question. Although several things need to be considered in applying a metaheuristic algorithm to the lifetime problem of WSN, it provides a way to find a better solution than the rule-based algorithm. That is why an increasing number of studies [15], [48]–[50] using *metaheuristic algorithm*s have been presented in recent years, which brought up several

[2]In this paper, the complex and difficult problem means that the number of candidate solutions (solution space) is simply way too large to find the optimal solution within a reasonable time by using an exhaustive search algorithm.

successful results, the details of which will be discussed in the following section.

III. Metaheuristic Algorithms for WSNs

A. From the Perspective of Problems

The focus of the following subsections will be on the research issues of metaheuristic algorithms for different lifetime problems, such as the concerns of implementation. Because each lifetime problem has its own characteristics, the following discussions will be divided into four parts: the number of alive nodes problem, the deployment problem, the k-coverage problem, and the cluster head election problem.

1) Metaheuristics for the Number of Alive Nodes: The number of alive nodes problem is a well-known traditional optimization problem for the lifetime of WSN. Although it can usually be defined as three kinds of problems, namely, n-of-n, k-of-n, and m-in-k-of-n lifetimes (as mentioned in Section II-A1), they still cannot fully describe the networks we are facing nowadays. Some definitions are harsh for the recent applications, such as n-of-n [51]. However, efficient solutions to these problems are still very useful and can be applied to other lifetime problems after minor modification. For the number of alive nodes lifetime problem defined and assumed herein, even the simple genetic algorithm is able to provide a better result than a traditional algorithm [51] does. Although metaheuristic algorithm-based algorithms can provide better results than traditional rule-based algorithms, most studies disregard how much energy has to be spent by using a metaheuristic algorithm to find solutions. In [9], Bari et al. presented a fast GA-based algorithm to find the final solution. It is fast in the sense that it takes a shorter computation time than simple GA and many traditional algorithms. This implies that for the general optimization problem, the design of metaheuristic algorithms needs to avoid falling into local optimum (i.e., fast convergence) at early iterations. For some real world problems (e.g., the number of alive nodes lifetime problem), in addition to the quality of the solution, the computation time is also critical. Some studies [19], [52], [53] showed that PSO is another metaheuristic algorithm which can provide better results than traditional rule-based algorithms in solving the number of alive nodes lifetime problem, not only for the n-of-n problem but also for the k-of-n problem.

2) Metaheuristics for the Cluster Head Election: The cluster head election problem is another traditional lifetime problem for which metaheuristic algorithms play a critical role, as a large number of previous studies show. The results of using GA [54]–[58], ACO [59], and PSO [26], [60], [61] showed that metaheuristic algorithms are capable of finding better results than traditional deterministic or rule-based algorithms do. For instance, PSO can find a better result than LEACH [60] does. To find better solutions for this problem, the strategy adopted by some researches [56], [58], [60] is to take into consideration more factors, as for the k-coverage problem. Another strategy is to use two different selections (one for ranking selection; the other for elitist selection) as in [57]. From the perspective of search strategies, a potential solution for this problem is the hybrid algorithm because combination of metaheuristic algorithms can be used to mitigate the drawback of these algorithms. These combinations usually rely on the concept of using one method for global search while using the other for local search, such as GA + SA [62], [63] and PSO + k-means [25].

3) Metaheuristics for Deployment Coverage: As mentioned in Section II-A3, a well-known coverage lifetime problem is the deployment coverage problem the aim of which is to find applicable locations for the sensors to cover more regions or targets. The metaheuristic algorithms adopted for this lifetime problem, such as SA [28], [64], TS [29], [65], GA [49], [66]–[68], and PSO [69]–[71], have produced a large number of successful results. An example of those described in [70] shows that PSO can be used to relay the nodes deployment problem. The simulation results show that the energy consumption of WSN can be improved by about 50-70% when adding 4–12 relay nodes to the network by using PSO to plan the relay nodes deployment.

Because this kind of problem is aimed at finding a set of applicable locations for the sensors, the solutions are generally structured and encoded as a set of coordinates or a grid (array) to represent the *good* positions for the sensors. In addition to the coverage, the energy consumption is another measurement often used in this problem. For instance, the two major concerns of [28] are the maximum coverage and the minimum power spending for the transmission of data to prolong the lifetime of a WSN. Of course, when we take more factors of a problem into account at the same time, some of the objectives may conflict, such as coverage and lifetime.[3] To solve this issue, some studies [65], [67] tried to define the problem as a multi-objective problem for which the search algorithms will then try to find the Pareto optimal sets to help us make a suitable decision for managing WSN.

4) Metaheuristics for Set-Cover: Different from the deployment problem that is aimed at finding the "good locations" for the sensors, the goal of set k-cover problem is to find the "good partitions" of the sensors for which GA [30], [31], [72], [73] and SA [74] and TS [32] are often used. One of the typical representations for these algorithms is a set of cluster ID's to represent to which cluster (group) a sensor belongs. From this point of view, it can be easily seen that this problem is similar to the traditional partitional clustering problem; thus, clustering algorithms may be able to be used for this problem. To measure the performance of solutions to this kind of problem, we have to take into account not only the number of covers but also the success rate to achieve the maximum number of covers or the average running time of successful runs [32]. Like the deployment problem, the conflict may also exist as far as this problem is concerned. Therefore, NSGA-II is used for solving this problem [75].

5) Metaheuristics for Data Routing: Finding a better routing path to send data from one node to another is the main concern of the data routing problem of WSN. In addition to SA [76] and GA [33], [77]–[80], a large number of

[3]The conflict of coverage and lifetime comes from the fact that the more regions we want to cover, the more energy we need to spend, thus the less the overall lifetime of WSN, especially when the network has only limited energy.

approaches use ACO-based algorithms [34], [35], [81]–[87]. This is because the original design of ant colony optimization is aimed at finding the shortest path (i.e., at constructing the solution) for ant routing. Each solution is constructed step by step (i.e., sub-solution by sub-solution), and this means that it can take into account the order and relationship between sub-solutions. For instance, to select the best next hop node, in [35], $\eta_{\mathbf{r},\mathbf{s}}$ of ACO has been defined as the inverse of the hop count from node **s** to the next node **r**.

B. From the Perspective of Algorithms

The focus of the following subsections will be on the considerations, restrictions, and implementation issues of metaheuristic algorithms. To make these discussions as clearly and easily to understand as possible while at the same time taking into account the fact that some of the algorithms described here have similar research issues, the discussions are divided into three parts: single-solution-based algorithms, evolutionary algorithms, and swarm intelligence.

1) Single-Solution-Based Algorithms: Because simulated annealing (SA) and tabu search (TS) are typical of the single-solution-based metaheuristic algorithms, a number of studies for WSN tried to use SA [27], [28], [62], [64], [74], [76] and TS [29], [32], [48], [65], [88] to prolong the lifetime of WSN.

As one of the typical single-solution-based metaheuristic algorithms, simulated annealing (SA) also confronts the problem of small search diversity at each iteration when applied to the lifetime problem. That is why Wang et al. [27] tried to combine SA with PSO (another metaheuristic algorithm) to provide a better result. Another hybrid algorithm presented in [62] combines SA with GA. In such a combination, SA usually plays the role of searching for the possible solution in a local region which can be regarded as intensification-oriented while the other metaheuristic algorithm plays the role of finding the better search directions with a global view on the solution space that can be regarded as diversification-oriented. Another striking issue is that the end result may be affected by the initial solution of a single-solution-based algorithm. It means that a good initial solution (starting point) may lead SA to find a better solution; a bad initial solution may make SA fall into local optimum at early iterations even though it owns an escape mechanism. In addition, because only one solution is searched at a time, if the initial solution is far away from the optimum solution, SA will take much longer to find it. To solve this problem, in [76], the broadcast incremental power (BIP) [89] is used to create a better initial solution for SA. The results of [76] also illustrate that a refined initial operator can clearly improve the search performance of SA.

Another well-known single-solution-based metaheuristic algorithm for the lifetime problem of WSN is tabu search (TS) [29], [32], [48], [65], [88]. Similar to SA, some research issues of SA also exist in TS. For instance, in [88], a greedy heuristic is used to create a better initial solution to improve the quality of the end result. Moreover, since TS is also a single-solution-based metaheuristic algorithm, a better transition operator may enhance the performance of the search diversity during the convergence process. In [29] and [65], Aitsaadi et al. presented two neighbor generation methods for transiting the current solution to the next state which is either the suppression-oriented stage (to reduce sensors from those deployed in the over-covered areas) or addition-oriented stage (to add sensors to those deployed in the under-covered areas). Another study [48] presented a multiple neighbor generation method for TS which considers changing the solution structure in the regular node, active node, and cluster head at the same iteration.

2) Population-Based Algorithms—Genetic Algorithm: Genetic algorithm (GA) is a population-based algorithm inspired by the Darwin's theory to simulate the evolution of humans in a computer. It uses a set of chromosomes (also called individuals) to represent the feasible candidate solutions on the evolution process. This means that GA can search more candidate solutions at the same time, thus having a higher search diversity than single-solution-based algorithms have. The disadvantage is that it takes a longer computation time. Since genetic algorithm can be regarded as a high performance search tool, a large number of methods based on it for solving the lifetime of WSN [30], [31], [33], [49], [54]–[58], [66]–[68], [72], [73], [75], [77]–[80], [90], [91] have been presented in recent years. Since the fitness function, selection, crossover, and mutation are the main operators of GA, they will be performed at each generation until the termination criterion is met in most cases. The fitness value of each chromosome will be used by the selection operator. For example, the roulette wheel selection determines the search direction by computing the probability of each chromosome being the parent for chromosomes of next generation based on the fitness value of each chromosome as follows:

$$\mathbf{P}_i = \frac{f_i}{\sum_{j=1}^{\theta} f_j}, \tag{6}$$

where $\mathbf{P}_i$ is the probability of the i-th chromosome being the parent of chromosomes of next generation, f_i the fitness value of the i-th chromosome, and θ the number of chromosomes in the population. The crossover operator is one of the transition operators of GA which typically plays the role of exchanging information between chromosomes with a certain probability, called the crossover rate, c_r. For instance, $c_r = 0.6$ means that only about 60% of the chromosomes will be used by the crossover operator to exchange the information with another chromosome while the other 40% will not be exchanged by the crossover operator. As shown in Table I, after the crossover operator, Crossover(p_i^t, p_j^t), is carried out, the new chromosomes are formed from the i-th and j-th chromosomes. Another representative transition operator of GA is the mutation operator which is also associated with a probability, called the mutation rate, m_r, implying that each gene has a probability m_r of being changed. If the mutation operator, Mutation(p_i^t), is applied to a new chromosome, a small portion of genes will be changed randomly, such as randomly swapping the values of two different genes of the chromosome. To give a systematic description for these studies, the discussions that follow are divided into three parts: *representation*, *fitness function and selection*, and *transition*.

a) Representation: To apply GA to a new optimization problem, the very first thing to do is to find or develop a suitable representation to characterize a solution of the problem. For the lifetime problems, we also need to do it to design a high performance algorithm. The typical representations of GA for the lifetime problems are as follows:

- Bit string: This representation assumes only the values 0 and 1. In other words, a candidate solution is composed of a string of bits. In this kind of representation for the lifetime problems of WSN, the length of solutions is usually set equal to the number of sensors, where the value 1 is used to represent a sensor node for a particular task, say, the sensor node being a cluster head [58] or the sensor node being selected to be used [75]. For example, if the cluster head election lifetime problem [58] of a WSN with $\psi = n$ sensor nodes is encoded as a bit string, then each gene $p_{i,j} \in \{0, 1\}$ of a chromosome $p_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,\psi}\}$ represents that the j-th sensor node will be elected to be the cluster head if $p_{i,j} = 1$; otherwise, the j-th sensor node will not be elected as the cluster head.
- Integer: This representation is much more flexible than bit string in representing the solutions we are looking for. For example, in [30], [31], [49], [72], and [73], each gene encodes the cluster ID (an integer) to which a sensor node belongs. More precisely, a chromosome can be used to encode a solution as follows: $p_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,\psi}\}$ where $1 \leq p_{i,j} \leq k$ and $\psi = n$ is the number of sensors; k the maximum number of disjoint cover sets. For instance, $p_{i,1} = 1$ means that $p_{i,1}$ will be scheduled to cover set 1 while $p_{i,2} = 3$ means that $p_{i,2}$ will be scheduled to cover set 3. Another kind of integer representation is used to represent different states of each sensor. For instance, in [91], 0 represents there is no node in this grid, 1 represents the cluster number, and 2 represents the cluster head, respectively. In [57], the integer value of a gene represents the work directions of a sensor.
- Tree structure: Compared to the other representations, this representation is usually used to represent the routing-based lifetime problems [68], [79], [80]. One way to use this representation is to use the value of the j-th gene to represent the parent node of j [79], [80]. Another way to use this representation, as described in [68], is to use the j-th gene to represent the next hop node which also associates with the angle of this node.
- Others: Different from the above representations, some studies developed their own representations [54], [55], [66] to encode (represent) the solutions of GA. In [66], each chromosome is used to encode an $L \times L$ square field, the length of which is $2L^2$ because this algorithm uses two bits to represent the state (mode) of a node. More precisely, it uses 00, 01, 11 to represent, respectively, the highest energy, medium energy, and lowest energy. A similar representation is used in [54] where the state of each sensor node is represented by 3 bits instead of 2 bits. They are inactive node, node to be the cluster head, node to be the inter-cluster router, and node to be a sensor node. Moreover, a hybrid structure of representations is also used by GA for the lifetime problems of WSN. For example, in [55], a chromosome can be divided into two columns: one for the cluster to which a node will belong while the other for denoting whether this cluster is in use or not; that is, each node is associated with a cluster ID and a flag (0 means in use; 1 means not in use).

b) Fitness Function and Selection: Once the representation of GA is determined, the next thing to do is to determine how the *fitness* of each chromosome is evaluated so as to make a decision to *select* the suitable chromosomes (individuals) to be passed on to the next generation. Although some studies use the so-called objective function to measure the chromosome directly, the so-called fitness function is not always equivalent to the objective function because some optimization problems use more than one measurement, such as the clustering problem. Traditionally, only one objective is used to measure the results of the lifetime problem; the new trend in this research domain, however, has been to use multiple objectives [58], [75]. For example, for scheduling the data gathering of relay nodes on WSN, Bari et al. [9] used initial energy of a relay node $E(i)$ and maximum energy dissipated by any relay node in the individual $E(m)$ to compute the fitness value f of each chromosome as follows: $f = E(i)/E(m)$. Another example for routing strategy to prolong the lifetime of WSN can be found in [33], Singh and Sharma considered using distance between nodes and energy balance at the same time to evaluate the fitness of each chromosome; that is, $f_i = \sum_{i=1}^{n-1} d_i^2 + f_e$, where n is the number of nodes; f_e the measurement of energy balance; and d_i the distance between the i-th node and the $(i + 1)$-th node.

Several traditional selection methods can also be applied to the lifetime problems of WSN, such as roulette wheel selection [56], [68], [79], [80] and tournament selection [33], [91]. Moreover, some recent studies also used ranking selection [55] or even combined ranking selection with elitist selection [57] for the GA to solve the lifetime problems. In [31], [73], and [78], the $(\mu + \lambda)$ survivor selection and tournament selection were used for selecting the parents and survivors; that is, the survivor selection merges the parents and the children to pass them on to the next generation.

c) Transition: The transition operator of GA usually plays several roles in searching for the optimal solution in the search space. The two typical sub-operators for the transition operator of this search algorithm are crossover and mutation. The crossover sub-operator plays the role of exchanging the information between two chromosomes (solutions) whereas the mutation sub-operator plays the role of fine-tuning the solution to avoid the search process from falling into local optimum. When applying GA to different lifetime problems of WSN, these two transition sub-operators may need to be changed or redesigned, which can be found in several studies. To avoid illegal solutions, repair sub-operators have also been presented in some studies [33], [91]. GA normally uses the one-point crossover to split a chromosome into two parts, one of which will be passed directly

on to the next generation while the other of which will be exchanged with another chromosome. For instance, given two chromosomes, p_1^t and p_2^t, if the crossover point is at locus i, then after crossover, the two offspring will be $p_1^{t+1} = \{p_{1,1}^t, p_{1,2}^t, \ldots, p_{1,i}^t, p_{2,i+1}^t, p_{2,i+2}^t, \ldots, p_{2,\psi}^t\}$ and $p_2^{t+1} = \{p_{2,1}^t, p_{2,2}^t, \ldots, p_{2,i}^t, p_{1,i+1}^t, p_{1,i+2}^t, \ldots, p_{1,\psi}^t\}$, where ψ is the length of each chromosome. In [55] and [78], the one-point crossover is used in GA for the lifetime problems which are used for the general and hybrid structures of chromosomes. The two-point crossover can be considered as a variant of the one-point crossover. The information exchange mechanism will first randomly choose two points (loci) to determine which segments should be exchanged between two chromosomes. Then, after crossover, one of the offspring will be $p_1^{t+1} = \{p_{1,1}^t, p_{1,2}^t, \ldots, p_{1,i}^t, p_{2,i+1}^t, p_{2,i+2}^t, \ldots, p_{2,j}^t, p_{1,j+1}^t, p_{1,j+2}^t, \ldots, p_{1,\psi}^t\}$, assuming i and j, $i < j$, denote the two crossover points. This kind of operator is used in solving not only the lifetime problems [33], [56] but also the single-objective problems [33] and the multi-objective problems [56]. The uniform crossover uses a fixed mixing ratio between two chromosomes, p_1 and p_2, to mix up the genes. If the ratio is set equal to 0.5, the value of the i-th gene of the z-th offspring $p_{z,i}^{t+1}$ has 50% of the chance to come from the gene of either parent; i.e., either $p_{1,i}^t$ or $p_{2,i}^t$. That is, multiple points (genes) will be exchanged between two chromosomes so that after crossover, p_1^{t+1} and p_2^{t+1} may assume the following values $p_1^{t+1} = \{p_{1,1}^t, p_{1,2}^t, p_{2,3}^t, p_{2,4}^t, p_{2,5}^t, p_{1,6}^t\}$ and $p_2^{t+1} = \{p_{2,1}^t, p_{2,2}^t, p_{1,3}^t, p_{1,4}^t, p_{1,5}^t, p_{2,6}^t\}$, assuming $\psi = 6$ and the loci are 3, 4, and 5; that is, the genes at loci 3, 4, and 5 are exchanged. The studies of [30], [49], [68], [73], [77], [80], and [91] showed that the uniform crossover now has been applied to several different lifetime problems, such as the set k-cover problem [30], [73] and the coverage problem [49]. An interesting observation of these studies is that GA will usually encode the solutions as integers when the uniform crossover is used.

Some studies on the lifetime problems use distinctive crossover, such as sub-tree exchange [79], block exchange [57], and order-based exchange [31], [72] because of the structure of representation, the meaning of chromosome, or the search strategy.

In addition to the repair operator [33], [91] that is used to avoid passing illegal chromosomes on to the next generation, the goal of the mutation operator in these studies [30], [31], [33], [49], [54]–[58], [66]–[68], [72], [73], [75], [77]–[80], [90], [91] is to fine-tune the solutions to avoid getting stuck at the same region. For example, in [91], the mutation operator is used to insert sensors to or delete sensors from the network. Another distinctive mutation operator described in [33] is to randomly choose a sensor node from the best chromosome to pass it on to the offspring instead of randomly selecting a gene from some chromosome so as to avoid getting stuck in a local minimum.

3) Population-Based Algorithm—Swarm Intelligence: Ant colony optimization (ACO) [43]–[45] and particle swarm optimization (PSO) [46], [47] are another two metaheuristic algorithms, which are also referred to as swarm intelligence (SI). Although these metaheuristic algorithms can be used to solve most kinds of optimization problems, our observation shows that the ACO-based algorithms suit discrete optimization problems while the PSO-based algorithms fit continuous optimization problems. Thus, it can be easily seen that a large number of studies [34], [35], [50], [59], [81]–[87], [92], [93] that use the ACO-based algorithms have been successfully applied to the lifetime problems of WSN, such as the scheduling and routing problem. Also, a large number of studies [25], [26], [60], [61], [63], [69]–[71], [94], [95] that use the PSO-based algorithms have also been successfully applied to the lifetime problems, such as the clustering problems.

In the studies using ACO for the lifetime problems, similar to GA, the very first issue is how to encode (represent) the solution for the lifetime problems. The good news is that most researches using the ACO-based algorithms for the lifetime problems use the routing path as a solution, such as those described in [50], [92], and [93]. As long as this kind of representation (just like the solution structure of the traveling salesman problem used in [43]) is used for the solution of the lifetime problem, another very important research issue is to modify the construction operator to make it conform to the problem in question. For instance, some studies [50], [92], [93] attempted to modify the construction operator of ACO so as to apply it to an $(n + k) \times m$ graph to find the possible solutions for assigning sensors to the connected covers, where n is the number of sensors, m the number of sinks, and k the number of disjoint connected covers. This kind of design for the construction operator illustrates that it is relevant to the solution representation because which kind of solution will be generated depends somehow on the construction operator. Because the construction operator generates possible solutions based on the construction probability $\mathbf{p_k}(\mathbf{r}, \mathbf{s})$ (e.g., the pheromone value of paths and the desirability of a sub-solution to the other sub-solutions), we have to take into consideration how to transfer the original probability equation of ACO

$$\mathbf{p_k}(\mathbf{r}, \mathbf{s}) = \begin{cases} \dfrac{[\tau_{\mathbf{r},\mathbf{s}}]^\alpha [\eta_{\mathbf{r},\mathbf{s}}]^\beta}{\sum_{\mathbf{s} \in \mathcal{N}_\mathbf{r}^\mathbf{k}} [\tau_{\mathbf{r},\mathbf{s}}]^\alpha [\eta_{\mathbf{r},\mathbf{s}}]^\beta} & \text{if } s \in \mathcal{N}_\mathbf{r}^k, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where $\mathcal{N}_r^k$ denotes the set of possible candidate sub-solutions; $\tau_{\mathbf{r},\mathbf{s}}$ and $\eta_{\mathbf{r},\mathbf{s}}$, respectively, the pheromone value and the heuristic value associated with $e_{\mathbf{r},\mathbf{s}}$, to define the lifetime problem of WSN. To compute the probability of the routing problem of WSN, in [84], the following pieces of information, namely, the number of sensing sensors, the received data, the transmitted data, the lifetime estimated previously, and the total input data rate, are used in $\mathbf{p_k}(\mathbf{r}, \mathbf{s})$. It means that $\mathbf{p_k}(\mathbf{r}, \mathbf{s})$ has to be modified to fit the requirements of the lifetime problem in question. To compute the desirability of $\mathbf{r}$ to $\mathbf{s}$, an example of $\eta(\mathbf{r}, \mathbf{s})$ for the clustering problem [59] is used to describe and define the energy consumption of the sensors, as follows:

$$\eta(\mathbf{r}, \mathbf{s}) = \frac{(I - e_r)^{-1}}{\sum_{\mathbf{s} \in \mathcal{N}_\mathbf{r}^k} (I - e_r)^{-1}}, \tag{8}$$

TABLE II
SUMMARIZATION OF METAHEURISTIC ALGORITHMS FOR LIFETIME PROBLEMS

$\mathbb{P}$ / $\mathbb{A}$	SA	TS	GA	ACO	PSO	Hybrid
NANP			[52], [9]		[19], [53]	
CHEP			[55], [56], [57], [58], [59]	[60]	[61], [27], [62]	[63], [64], [26]
DCP	[65], [29]	[66], [30]	[67], [68], [50], [69]		[70], [71], [72]	
SCP	[75]	[33]	[31], [73], [32], [74] [76]	[94]		
DRP	[77]	[89] [49]	[78], [79], [34] [92] [91]	[82], [35], [83], [84], [85], [86], [87] [51], [93]	[96]	

where I is the initial energy; e_r the current energy level of the receiving node; and $\mathcal{N}_{\mathbf{r}}^k$ all the receiving nodes. Also, a similar change of $\eta(\mathbf{r}, \mathbf{s})$ to represent the information of energy can be found in [81]–[83] and [86]. In [35], instead of the energy information, $\eta(\mathbf{r}, \mathbf{s})$ is the inverse of the hop count from node s to the slink node plus one which is used for the data aggregation problem of WSN.

Another famous swarm intelligence algorithm is particle swarm optimization which also has some successful results in the lifetime problems of WSN [25], [26], [60], [61], [63], [69]–[71], [94], [95]. Different from the ACO that uses a set of integers to represent the routing path, the solution of the original PSO was encoded as a set of real numbers. In [71], the particles encode a set of coordinates for the locations of the base stations as *real numbers*. For instance, suppose the number of dimensions is 2, and the number of possible clusters is 5, then the i-th particle p_i encodes two sets of coordinates: the first set for the five x-coordinates and the second set for the five y-coordinates. A similar representation is also used in [60]. The integer representation of PSO for the lifetime problems can be found in [70] which uses a uniform distribution within a range so that the particle encodes a set of integers for the deployment of the relay nodes. Similar to the integer representation, binary representation also needs some mechanism to map the real numbers to a range of 0 to 1. In [61], the sigmoid function Eq. (9) and a modified position update function Eq. (10) were used to ensure that the position of each particle is between 0 and 1.

$$\text{sig}(v_i) = \frac{1}{1 + e^{-v_i}} \tag{9}$$

$$p_i = \begin{cases} 0 & \text{if } r(t) \geq \text{sig}(v_i(t+1)), \\ 1 & \text{if } r(t) < \text{sig}(v_i(t+1)), \end{cases} \tag{10}$$

where v_i is the velocity of the i-th particle, r a random number in the range [0, 1].

Some studies [25], [26], [60], [69] attempted to include more objectives to the fitness function (i.e., to the evaluation operator) of PSO. A simple way is to combine more than one kind of fitness value to represent the fitness value of each particle. In [26], the fitness value of i-th particle is defined as $f_i = \alpha \times f_1 + (1 - \alpha) \times f_2$, where f_1 is the sum of the energy of all the member nodes; f_2 is the largest average distance between the cluster head and the joined member nodes. Many more factors were considered in [60], namely, the average distance of sensors to their cluster head, the energy of nodes in a cluster, the degree of nodes associated with particle p_i, and the probability of choosing a node from particle p_i. Similar changes to the fitness function can also be found in [25] and [69]. To improve the quality of the final results, some studies [63], [94], [95] tried to combine the PSO with other metaheuristic algorithms or local search algorithms. A simple combination can be found in [25] which uses the k-means algorithm to fine-tune the solution of each particle to further improve the quality of the end result. Of course, many other combinations have been found in recent studies, such as simulated annealing with PSO [63], chaotic system with PSO [95], and PSO with Kalman filter [94]. This illustrates that not all the metaheuristic algorithms are perfect because the results of the above studies showed that the other search algorithms may be able to help PSO improve the quality of the end result.

C. Summarization

Table II shows which lifetime problem can be solved by which metaheuristic algorithm. In Table II, $\mathbb{P}$ represents the problem type and $\mathbb{A}$ the algorithm. Abbreviations are used to represent problems and algorithms summarized in the table. For the problems, NANP denotes the number of alive node problem (as mentioned in Section II-A1); CHEP the cluster head election problem (as mentioned in Section II-A2); DCP the deployment coverage problem (as mentioned in Section II-A3); SCP the set cover problem (as mentioned in Section II-A4); and DRP the data routing problem (as mentioned in Section II-A5). For the algorithms, SA denotes the simulated annealing; TS the tabu search; GA the genetic algorithm; ACO the ant colony optimization; and PSO the particle swarm optimization. The detail descriptions of these algorithms are given in Section II-B.

Based on the classification given in Table II, now it can easily differentiate which kinds of algorithms are often used for which kinds of lifetime problems. However, for the empty field in this table, we are not arguing that the particular algorithm $\mathbb{A}$ cannot be applied to the particular problem $\mathbb{P}$. For example, for the NANP, the fields of SA and TS are empty. Rather, this simply implies that not much study is given to employ these two algorithms to solve NANP. Similar situations apply to hybrid algorithms for these lifetime problems; although some hybrid algorithms can provide good results, there is not much study focusing on these hybrid algorithms. Thus, this information indicates that using hybrid algorithms to solve these problems is a promising research topic.

According to our observation, although most metaheuristic algorithms can provide better results than traditional rule-based algorithms do, the time and energy taken by metaheuristic algorithms and rule-based algorithms are not

discussed in most of the research papers. But some researches [9] showed that the metaheuristic algorithm they propose can provide better results than traditional rule-based algorithms do, in terms of not only the lifetime time of the sensor network but also the computation time. This also implies that the computation cost of the management method is also a very important factor. The implementation details of these metaheuristic algorithms also are not discussed in most researches because most of the researches are based on some assumptions, objectives, and simulations. Of course, simulations may be able to point us to the right research directions which can also be used to predict possible characteristics of these algorithms, but real world approaches are needed, especially when some of the sensor technologies and hardware are not mature enough. In brief, metaheuristic algorithms have a high potential to prolong the lifetime of WSN, and a large number of studies already showed the capabilities of these methods.

As mentioned in Section II-C, the problem definitions, assumptions, and environment parameters need to be considered before we apply a metaheuristic algorithm to the lifetime problem of WSN. The discussions that follow will be on the characteristics of the following metaheuristics: TS, SA, GA, ACO, and PSO.

- SA and TS: Since these two metaheuristics are both single-solution-based algorithm, the complexity of the operators and the computation cost of these two algorithms are usually less than population-based metaheuristics per iteration, e.g., GA, ACO, and PSO. The advantages of these two algorithms are consequently faster each iteration and much easier to implement compared to the population-based metaheuristics in most cases. For example, the implementation of SA [28], [64] and TS [29], [65] for the DCP is much easier than the implementation of GA [49], [66]–[68]. Because most single-solution-based algorithms for WSN use one search direction to find the possible solution at the same time, it may easily fall into a local optimum. One way to solve this problem is to modify the search strategy. For instance, a random walk operator is added in [32] to disturb the current solution to avoid falling into local optimum at early iterations. From the practical perspective, TS and SA are suitable for real-time lifetime problem and systems with low computation ability.
- GA: Different from TS and SA, GA is a population-based metaheuristic algorithm. Because most of the solution representations can be used for GA, it can be used in solving most lifetime problems of WSN, as shown in Table II. As mentioned in Section II-B, because GA uses multiple search directions at each iteration, it has a higher chance to find better solutions than single-solution-based metaheuristics. In addition to using crossover to exchange information between chromosomes and mutation to avoid falling into local optimum, the selection and fitness functions are the characteristics of GA that may have a strong impact on the search strategy. That is why the selection method and the fitness function are modified so as to change the search strategy for the lifetime problem. For example, in [49], the number of cover sets and the percentage of coverage are taken into account at the same time in the fitness function. One of the important advantages of GA is that it can provide a better result than simple TS and SA for complex lifetime problem in most cases; however, its disadvantage is that it will spend much more computation time than single-solution-based algorithm each iteration.
- ACO: The main characteristic of ACO is that it employs a pheromone table to keep track of the search information. Because the original design of ACO is for the traveling salesman problem, it is usually more suitable for the combinatorial optimization problem. That is why ACO has been applied to several routing-based lifetime problems of WSN [34], [50], [81]–[86], [92]. Because ACO constructs the sub-solutions step by step and considers the overall distance of a complete routing path of each ant, it can take into consideration the relationship between sub-solutions and the quality of a complete solution at the same time. That is why it can find a better solution than the other metaheuristics for the routing-based lifetime problems in most cases.
- PSO: Different from ACO, PSO is another swarm intelligent which is suitable for the lifetime problem in the continuous space because the particle (i.e., candidate solution) of PSO can be regarded as a position in the solution space. The main characteristic of PSO is to use the velocity information to change the position of the particle in its transition operator. As a result, PSO can find a better solution very quickly. As shown in Table II, it can be applied to most lifetime problems of WSN. However, the drawbacks of PSO are that it may fall into local optimum at early iterations and that it is not suitable for the combinatorial optimization problem of WSN.

All these metaheuristic algorithms have pros and cons; that is, none of them can provide a better result in all the lifetime problems of WSN in terms of both the computation time and the quality of the end results. Hybrid algorithm is a promising solution. The basic idea is to integrate two or more metaheuristics into a single search algorithm, to leverage the strength of each metaheuristic algorithm. However, the implementation will be more difficult than that of a standalone metaheuristic algorithm. So far, only a few studies have been focusing on the development of hybrid algorithm for the lifetime problem of WSN [25], [62], [63]. Moreover, two essential issues need to be considered. First is the higher computation cost. Second is that the integration may degrade the quality of the search results. In summary, each metaheuristic algorithm has its unique advantages and disadvantages. This implies that there is plenty of room to improve.

IV. The Examples

To further explain how to apply metaheuristics to the lifetime optimization problems of WSN, a number of examples are given in this section. The examples show how to apply SA, TS, and GA to the set-cover problem (SCP), ACO to the data routing problem (DRP), and PSO to the number of alive nodes problem (NANP).

A. SA for SCP

Since some studies assumed that the lifetime problem of WSN can be regarded as the set k-cover problem. A better solution to the set k-cover problem can then be used for prolonging the lifetime of WSN. For instance, Lin and Chiu [74] employed SA to find a better solution of this problem. Algorithm 2 shows an outline of how SA is applied to the set k-cover problem of WSN, called SA-SCP.

The input parameters of this algorithm are a set of n sensor nodes, the number of covers required for the WSN k, a matrix a where $a_{i,l}$ takes the values 0 and 1, with $a_{i,l} = 1$ representing that the l-th service point can be covered by the i-th sensor; $a_{i,l} = 0$ otherwise. The solution is encoded as a matrix p where $p_{i,j}$ assumes the values 0 and 1, with $p_{i,j} = 1$ indicating that the i-th sensor node is allocated to the j-th cover; $p_{i,j} = 0$ otherwise. Thus, a solution to the set k-cover problem can be defined as a 0/1 bit string. After lines 1–3 are performed, the initial temperature $\mathcal{T}$ and the candidate solution p will then be initialized. As long as the temperature $\mathcal{T}$ has not reached the termination temperature $\mathcal{T}_f$, SA-SCP will perform lines 5–21 repeatedly. As shown in line 6, in each iteration of SA-SCP, one of three actions will be randomly selected to transit p^t to p^{t+1}. These three actions are, respectively, remove a sensor node from a cover, add a sensor node to a cover, and exchange a sensor node in one cover with one in another. The energy E on line 7 is defined by

$$E = (1 + \mathcal{P}_e \sum_{j=1}^{k} \sum_{h=1}^{o} g_{h,j})(1 + \mathcal{P}_e^2(1 - d_m)) \sum_{w=1}^{W} \sum_{j=1}^{k} f_i \, p_{i,j}, \tag{11}$$

where $\mathcal{P}_e$ is a constant for penalty; $g_{h,j}$, which represents whether the h-th grid point (e.g., target node) is covered by sensors in the j-th cover, is defined as

$$g_{h,j} = \begin{cases} 1, & \text{for } \sum_{w=1}^{W} a_{i,h} \, p_{i,j} = 0, \\ 0, & \text{otherwise;} \end{cases} \tag{12}$$

d_m, which represents the minimum Hamming distance between $a_{i,j}$ and $a_{i',j}$, $i \neq i'$, is defined as

$$d_m = \begin{cases} 1, & \text{for } \min \sum_{w=1}^{W} (a_{i,j} - a_{i',j})^2 y_i \geq 1, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

W is the number of candidate locations; k is the number of covers; o is the number of grid points; f_i is the cost of the i-th sensor node. Moreover, E_o, E_n and E_m in Algorithm 2 are the energy value E of the old solution p^t, the new candidate solution p^{t+1}, and the minimum energy, respectively; and β is set equal to 1.3 for the cooling schedule. As shown in lines 5–20 of Algorithm 2, SA-SCP will try to find the candidate solutions r times each iteration. Lines 11–18 show that SA-SCP is similar to SA which uses the energy E to determine whether the new candidate solution is acceptable or not. The simulation results given in [74] show that the number of sensor nodes of SA-SCP is about 10.39% of the intuitive approach.

Algorithm 2 Outline of SA for SCP [74]

1 *Input* the information of WSN
2 *Initialize* the parameters
3 *Generate* randomly an initial solution p^0.
4 **While** the termination criterion is not met
5 **Repeat** r times
6 *Generate* neighbor randomly by removing, adding, or exchanging p^t
7 *Calculate* E using Eq. (11)
8 $E_n = E$
9 $\Delta E = E_n$ - E_o
10 *Randomly* generate γ which is uniformly distributed in (0, 1)
11 **If** $\Delta E \leq 0$ or $\gamma < e^{-\Delta E/t}$
12 $E_o = E_n$
13 **If** $E_n < E_m$
14 $E_m = E_n$
15 **Else**
16 Recover the change made in step 6
17 **End**
18 **End**
19 *Update* the temperature $\mathcal{T}$
20 $r = r \times \beta$
21 $t = t + 1$
22 **End**
23 *Output* the solution

Algorithm 3 Outline of TS for SCP [32]

1 *Input* the information of WSN
2 *Initialize* the parameters
3 *Generate* randomly an initial solution p.
4 **While** the termination criterion is not met
5 *Random* generate γ which is uniformly distributed in (0, 1)
6 **If** $\gamma < H$
7 $p^{t+1} = \text{RandomWalk}(p^t)$
8 **Else**
9 Generate a set of neighbors $N(p^t)$ randomly
10 Find the best neighbor p^b
11 $p^{t+1} = p^b$
12 **End**
13 Update the tabu list
14 Update the best so far solution p^*
15 $t = t + 1$
16 **End**
17 *Output* the solution

B. TS for SCP

In addition to using SA for SCP, Ting et al. [32] used the tabu search for SCP, called the TS-SCP in this paper. Different from [74] that uses a 0/1 string to represent how to allocate sensor nodes to cover the targets, the study of [32] employed an "order" of sensor nodes to be collected for covers. As shown in Algorithm 3, like the simple tabu search algorithm, TS-SCP also generates a set of neighbors at each

Algorithm 4 Outline of GA for SCP [49]

1 *Input* the information of WSN
2 *Initialize* the parameters
3 *Generate* randomly an initial population p.
4 **While** the termination criterion is not met
5 p^{t+1} = RecombinationSelection (p^t) using Eqs. (15) and (16)
6 p^{t+1} = Mutation (p^{t+1})
7 p^{t+1} = SchedulingTransition (p^{t+1})
8 f^{t+1} = Evaluation (p^{t+1}) using Eq. (16)
9 $t = t + 1$
10 **End**
11 *Output* the solution

iteration, as shown in line 9, but only one of them will be used as the next solution p^{t+1} if it is not in the tabu list or it satisfies the aspiration criterion, as shown in lines 10–11. After these procedures are carried out, TS-SCP will then update the tabu list and the best-so-far solution at each iteration. Unlike the simple tabu search, TS-SCP adds a random walk procedure to keep the search process from falling into local optimum at early iterations, as shown in lines 6–7. The random walk operator will change the current solution p^t by randomly swapping two of its sub-solutions with a probability H. For example, a solution $p^t = (\mathbf{1}, 2, \mathbf{3}, 4, 5)$ after the swap procedure is performed for the first and third sub-solutions will become $p^{t+1} = (\mathbf{3}, 2, \mathbf{1}, 4, 5)$. As shown in [32], TS-SCP can improve the quality of the end result of simple tabu search significantly, such as the success rate which is defined as

$$S_r = \frac{\text{number of successful runs}}{\text{number of runs}}, \tag{14}$$

where the number of successful runs means the number of runs the algorithm can achieve the maximum number of covers which is obtained from the maximum covers using mixed integer programming (MCMIP) [96]. According to the observations of Ting et al., the computation time TS-SCP takes is only about 0.1% of MCMIP. This means that TS-SCP provides an alternative solution for the lifetime optimization problem of WSN with a reasonable time compared to MCMIP. The simulation results also show that TS-SCP can provide a better result than hill climbing and integer-coded GA in terms of the success rate.

C. GA for SCP

In [49], Hu et al. combined the genetic algorithm with three scheduling transit operators to maximize the lifetime of WSN, which is referred to as GA-SCP in this paper. The solution representation of GA-SCP is different from the solution representation of SA-SCP [74] and TS-SCP [32]. Each gene $p^t_{i,j}$ is associated with a sensor, and its value represents the scheduling number of the sensor node for activation. For example, suppose a chromosome $p^t_i = (1, 1, 2, 2, 1, 3, 2, 1)$ of GA-SCP represents to which group (i.e., cover set) each of the eight sensor nodes belongs. If a complete cover requires three or more sensors, then sets 1 and 2 are complete cover sets, but set 3 is not because it has only one sensor. This chromosome also represents that sensors 1, 2, 5, and 8 belong to the first group; sensors 3, 4, and 7 belong to the second group; and sensor 6 belongs to the third group. The sensor nodes with a gene value 1 indicate that these nodes will be activated first, and the sensor nodes with a gene value 2 will fall asleep until the second cover set is activated. As shown in Algorithm 4, the GA-SCP is an extended genetic algorithm which adds the *SchedulingTransition*() operator to GA. All the genes of each chromosome p_i in the initial population p will be set to 1 to represent that all the sensor nodes are activated to cover the targets. For each chromosome, a predefined number of genes, say $\mathcal{K}$, will be randomly selected to check. If the selected gene is "redundant," the value of this gene will be incremented by 1. More precisely, that the gene is redundant means that the coverage circumstance (i.e., covergence percentage) will not be changed if this sensor node is set to be in the sleep mode. After these procedures are performed, the initial population will be created, as shown in line 3.

As shown in line 5, the transition and evaluation operators of GA-SCP are integrated into one operator to make sure that it can find better candidate solutions. This RecombinationSelection operator will first randomly select two chromosomes p^t_i and p^t_j from the current population p^t and then use the uniform crossover to create a new offspring p^{t+1}_x, as follows:

$$p^{t+1}_{x,k} = \begin{cases} p^t_{i,k}, & \text{if } q_0 < 0.5, \\ p^t_{j,k}, & \text{otherwise.} \end{cases} \tag{15}$$

This means that the k-th gene of the new offspring p^{t+1}_x has a fifty-fifty percent of chance to come from the k-th gene of either p^t_i or p^t_j. The offspring p^{t+1}_x will be selected to next generation if it has a fitness value better than its parents; otherwise, the offspring p^{t+1}_x will be replaced by a better parent chromosome. The fitness value can be divided into two parts, as shown below.

$$f_i = \omega_1 c_i + \omega_2 p_{c_i+1}, \tag{16}$$

where c_i is the number of disjoint complete cover sets of chromosome p_i, p_{c_i+1} is the percentage of coverage of the (c_i+1)-th cover set, and ω_1 and ω_2 are the predefined weights for c_i and p_{c_i+1}. By using Eq. (16), the RecombinationSelection operator can then find better candidate solutions from the current chromosomes and their offspring. As shown in line 6 of Algorithm 4, mutation is an important step to keep the search diversity of GA-SCP. Because the mutation operator may worsen a chromosome, GA-SCP applies it once every G_m generations and only to the best chromosome in the current population. In addition, only particular genes of the best chromosome will be mutated with a predefined mutation rate if these genes are in an incomplete cover set. As shown in line 7 of Algorithm 4, the SchedulingTransition() operator, which is used to reduce the redundancy of sensor nodes to improve the end result, essentially contains the mixed, forward, and critical transition operators. The mixed transition operator is responsible for moving a redundant sensor node to another cover set; the forward transition operator is responsible for enhancing the percentage of coverage of an incomplete

Algorithm 5 Outline of ACO for DRP [82]

1 *Input* the information of WSN
2 *Initialize* the parameters
3 *Generate* randomly a set of ants p.
4 **While** the termination criterion is not met
5 p^{t+1} = SolutionConstruction(p^t, τ) using Eq. (17)
6 τ = PheromoneUpdate(p^{t+1}) using Eq. (18)
7 $t = t + 1$
8 **End**
9 *Output* the solution

cover set; and the critical transition operator is responsible for ensuring that an incomplete cover set will be covered by at least one sensor node. This operator can be regarded as a fine-tuning procedure of GA-SCP, which is aimed to improve the quality of the end result. As shown in lines 8 and 9 of Algorithm 4, after the evaluation operator is performed and the generation number is incremented, the search process of GA will go back to line 5. The simulation results of [49] show that GA-SCP can provide a better result than the other state-of-the-art methods [30], [97] in terms of both the computation time and the number of complete cover sets.

D. ACO for DRP

Since one of the characteristics of ant colony optimization (ACO) is that a solution is constructed sub-solution by sub-solution, it typically can find a better result for routing-based optimization problems. Camilo et al. [82] used ACO and improved ACO to solve the routing problem of WSN to prolong its lifetime, called ACO-DRP in this paper.

As shown in Algorithm 5, the basic idea of ACO-DRP is similar to the simple ACO which puts a set of ants p to the network nodes, as shown in line 3. When the i-th ant is at node $\mathbf{r}$, ACO-DRP will use Eq. (17) to compute the possible next hop node $\mathbf{s}$ to construct the sub-solutions step by step. Different from the simple ACO which considers the pheromone value and distance between a pair of nodes $\mathbf{r}$ and $\mathbf{s}$, ACO-DRP uses the pheromone value $\tau_{\mathbf{r},\mathbf{s}}$ and the energy E to compute the probability of the next node, as follows:

$$\mathbf{p_i}(\mathbf{r},\mathbf{s}) = \begin{cases} \dfrac{[\tau_{\mathbf{r},\mathbf{s}}]^{\alpha}[E(\mathbf{s})]^{\beta}}{\sum_{\mathbf{s}\in\mathcal{N}_{\mathbf{r}}^{i}}[\tau_{\mathbf{r},\mathbf{s}}]^{\alpha}[E(\mathbf{s})]^{\beta}}, & \text{if } \mathbf{s}\in\mathcal{N}_{\mathbf{r}}^{\mathbf{i}}, \\ 0, & \text{otherwise}, \end{cases} \tag{17}$$

where $E(s) = 1/\left(E^0 - e(\mathbf{s})\right)$, with E^0 being the initial energy level of the nodes and $e(\mathbf{s})$ being the actual energy level of the $\mathbf{s}$-th node. As shown in line 6, ACO-DRP updates the pheromone by using the following equations:

$$\Delta\tau_i(\mathbf{r},\mathbf{s}) = (1-\rho)\tau_i(\mathbf{r},\mathbf{s}) + \Delta\tau_{\mathbf{i}}, \tag{18}$$

$$\Delta\tau_i = \frac{1}{n - d_i}. \tag{19}$$

where n is the number of nodes and d_i is the distance traveled by ant i. This means that the pheromone value of each ant depends on the overall distance that the ant can find.

Algorithm 6 Outline of PSO for NANP [53]

1 *Input* the information of WSN
2 *Initialize* the parameters
3 *Generate* randomly an initial population of particles p.
4 **While** the termination criterion is not met
5 **For** each particle i
6 *Calculate* the lifetime of p_i using Eq. (21)
7 *Find* the particle p_i that has the minimal lifetime using Eq. (22)
8 *Update* the personal best pb_i and the global best gb
9 *Change* the velocity v_i and the position p_i using Eqs. (23) and (24)
10 **End**
11 $t = t+1$
12 **End**
13 *Output* solution

In the same study [82], Camilo et al. also presented another pheromone calculation method, as defined below.

$$\Delta\tau_i = \frac{1}{E^0 - \text{avg}\,E(i) - (1/\min E(i))}. \tag{20}$$

Different from Eq. (19), Eq. (20) takes into consideration much more information for the solution of each ant which includes: the initial energy level of the nodes E^0, the average of the energy values of the i-th ant '$\text{avg}\,E(i)$', and the minimum energy value of the i-th ant '$\min E(i)$'. The results of [82] show that the energy of ACO-DRP can be reduced significantly by using Eq. (20) to replace Eq. (19) to further prolong the lifetime of WSN.

E. PSO for NANP

In general, WSN can be divided into two categories: homogeneous WSN and heterogeneous WSN in terms of the ability of sensors, such as data transmission rates, initial energies, and parameter values. When different kinds of sensors or application nodes exist in a wireless network environment, it is expected that many more factors and limitations have to be taken into consideration for finding a certain number of predefined critical nodes to be the base stations (BSs), which can be regarded as the m-in-k-of-n lifetime problem we mentioned in Section II-A1. Basically, the task of finding applicable base station locations is quite similar to that of finding the food locations originated from particle swarm optimization. As such, in this section, we use it as an example to demonstrate the performance of metaheuristics in solving the m-in-k-of-n lifetime problem of WSN under general power-consumption constraints in WSN.

Algorithm 6 gives an outline of how PSO is applied to the m-in-k-of-n lifetime problem, called PSO-MNK, to find the best base stations [53]. The input of this algorithm is a set of n sensor nodes each of which consists of the location (e.g., (x_j, y_j) for a two-dimensional problem), data transmission rate r_j, initial energy e_j^0, and parameter values $\alpha_{j,1}$ and $\alpha_{j,2}$. In addition, according to the definition of m-in-k-of-n, the input also consists of the number of alive sensors k and the

set S_m of the m predefined critical (supporting) sensors. Given the input, here is a brief description of how the algorithm works. First, all the parameters, such as the fitness value f_i, the personal best pb_i for each particle p_i, and the global best pb for the whole population, are initialized, as shown in line 2 of Algorithm 6, Then, as shown in line 3 of Algorithm 6, PSO-MKN randomly creates a set of n particles each of which, denoted by p_i^t, represents a possible base station location at iteration t. The initial position p_i^0 is randomly created in the initialization step, so does the initial velocity p_i^0. More precisely, to solve the m-in-k-of-n lifetime problem under general constraints, each particle is used to represent (encode) a possible base station location.

Then, as shown in lines 4–12 of Algorithm 6, the search process of PSO-MKN will be repeated until the termination criterion is met. As far as this algorithm is concerned, as shown in line 6 of Algorithm 6, the lifetime $l_{i,j}$ of each particle p_i is computed, as follows:

$$l_{i,j} = \frac{e_j^0}{r_j(\alpha_{j,1} + \alpha_{j,2} d_{i,j})}, \tag{21}$$

where e_j^0 is the initial energy of the j-th sensor; r_j the data transmission rate of the j-th sensor; $\alpha_{j,1}$ and $\alpha_{j,2}$ the distance-independent parameter and the distance-dependent parameter of the j-th sensor; and $d_{i,j}$ the Euclidean distance between the i-th particle (BS) to the j-th sensor. Then, in line 7 of Algorithm 6, the fitness value of the i-th particle of PSO-MKN is computed as the smaller of the minimal lifetime among nodes in the critical node set S_m and the minimal lifetime among nodes outside the critical node set S_m, as follows:

$$f_i = \min\left\{\min_{j \in S_m}\{l_{i,j}\}, \min_{j \notin S_m}{}_{n-k+1}\{l_{i,j}\}\right\}, \tag{22}$$

where n denotes the number of sensors and k the smallest number of sensors that must be alive.

Then, as shown in line 8 of Algorithm 6, the personal best $\text{pb}_i = \{\text{pb}_{i,1}, \text{pb}_{i,2}, \ldots, \text{pb}_{i,D}\}$ and the global best $\text{gb} = \{\text{gb}_1, \text{gb}_2, \ldots, \text{gb}_D\}$, where D is the number of dimensions, are updated if a better solution is found. Then, as shown in line 9 of Algorithm 6, the velocity and position of particle i at iteration $t+1$ are changed, as follows:

$$v_{i,j}^{t+1} = \omega v_{i,j}^t + a_1\varphi_1(\text{pb}_{i,j}^t - p_{i,j}^t) + a_2\varphi_2(\text{gb}_j^t - p_{i,j}^t), \tag{23}$$

and

$$p_{i,j}^{t+1} = p_{i,j}^t + v_{i,j}^{t+1}, \tag{24}$$

where $v_{i,j}^t$ and $p_{i,j}^t$ denote the velocity and position of the j-th dimension of the i-th particle at iteration t, ω an inertial weight, φ_1 and φ_2 two uniformly distributed random numbers, a_1 and a_2 two constant values which represent, respectively, the cognitive and social learning rate.

The empirical analysis was conducted on an AMD PC with a 2.0GHz processor and 1GB of memory running Microsoft Window XP and the programs are written in C. The test datasets are in a two-dimensional real space of $1,000 \times 1,000$; thus, both x and y axes are in the range of 0 to 1,000. The data transmission rate is limited in the range of 1 to 10, and the initial energy is limited in the range of 100,000,000 to 999,999,999. The data for all the sensors, each with its own location, data transmission rate, and initial energy, are randomly generated.

TABLE III
OPERATORS OF METAHEURISTIC ALGORITHMS [36]

Algorithm	Lifetime	Running Time
PSO-MKN	**79.0876**	**0.08**
EGS (gs = 1)	79.0350	36.19
EGS (gs = 0.1)	79.0821	2,763.36
EGS (gs = 0.001)	79.0871	212,087.81

The lifetime and running time (in seconds) of the PSO approach and the exhaustive grid search (EGS) with different grid sizes are shown in Table III. The comparison shows that PSO-MKN can provide a better result than the exhaustive grid search algorithm does. In addition, Table III also shows the running time of these algorithms. The results illustrate that the traditional or rule-based algorithms may spend a considerable amount of computation time. For instance, the computation time of the exhaustive grid search algorithm will increase quickly as the grid size becomes smaller. The comparison further shows that metaheuristic algorithm is a promising method for solving the lifetime problem that not only takes less computation time but also provides a better WSN environment.

V. OPEN ISSUES

A large number of studies using metaheuristic algorithm, including SA, TS, GA, PSO, and PSO, are reviewed and discussed in the paper from the perspective of problems and the perspective of algorithms. According to our observation, in addition to the issue of computation cost, many open issues remain to be considered to develop a high performance WSN environment for the lifetime problems. To better understand the situations of WSN we are in, here are some open issues for the lifetime problems of WSN, as we observed:

- Representation, information exchange, and evaluation: Until now, most metaheuristic algorithm-based algorithms for the lifetime problems of WSN inherit the design issues of the original algorithm. Therefore, many research issues still need to to be taken into account, such as how different problems are represented, how information exchange operators are designed, and how algorithms for lifetime problems are evaluated. Among them, many lifetime problems can be considered as a combinatorial optimization problem; thus, the representation which uses real number usually needs to be transformed. For the information exchange, how to balance the solution structure (intensification) and perturbation mechanism (diversification) would be the open issue in this kind of researches. For the evaluation, a complete comparison between different evaluation methods still does not exist in most papers, but we do believe that it is needed to ensure the performance of the algorithms.
- Parameter setting: This issue usually was neglected in most studies. Based on our observations, most researches

simply disregard how the parameters of their metaheuristic algorithm are determined. Also, the impact the parameter setting may have on the results is not discussed in most papers. However, even for the same algorithm, the results may be quite different with different parameter settings, such as setting the crossover rate of GA to 0.1 and 0.8. Thus, how to determine the parameter settings for different WSN environments and problems is an important issue.

- Multi-objective: Until now, most researches on the lifetime problems of WSN use weights to measure the importance of different objectives, but the reality is that the lifetime problems may not be a linear programming problem. In addition, for WSN, even though there are many factors to be considered, not many researches define the lifetime problems as a multi-objective problem. As such, multi-objective will be a promising research direction.
- Computation cost and convergence speed: The cost and convergence speed are another two open issues for the lifetime problems using metaheuristic algorithms because increasing the search diversity to avoid falling into local optimum at early iterations so as to improve the quality of the solution will in turn increase the computation cost. Different from the other research domains, instead of just the quality, taking into account the cost and quality at the same time is an open issue for the lifetime problems of WSN.

VI. Conclusions

In this paper, we review studies on applying metaheuristic algorithms to different lifetime problems of WSN. We also give a brief introduction to the lifetime problems and metaheuristic algorithms. Further discussions on metaheuristic algorithms for several different lifetime problems are given to show the capabilities and potential of metaheuristic algorithms. Discussions on the open issues are presented to explain the disadvantages and possible improvements we are facing nowadays. To point out the possible research trends, some promising research issues are given below:

- Self-organization and self-learning: There is no doubt at all that a lot of mechanisms for solving the lifetime problems of WSN are not able to handle a dynamic environment because they generally assume the network will be fixed once it is developed. However, uncertain factors and issues will affect the environment of WSN. For instance, contingency may reduce the number of sensors or cluster heads. A self-organization and self-learning network will certainly be very useful in this case. This implies that swarm intelligence will be very suitable because the underlying ideas of swarm intelligence are self-organization and self-learning.
- Combination of metaheuristic algorithm and rule-based algorithm: One of the useful strategies in using metaheuristic algorithms for other traditional optimization problems is to combine metaheuristic algorithm with rule-based algorithm designed particularly for this problem. In other words, because most metaheuristic algorithms are designed for general cases, not for a specific problem. Their performance may not be as good as we expect. For the lifetime problems, not many researches are focused on combining the traditional rule-based algorithm with metaheuristic algorithm. But according to our observation, it has a high potential to enhance the performance of these algorithms when applying to the lifetime problems of WSN.
- Internet of things, cloud computing, and big data: With the advance of the internet technologies and information systems, the internet of things, cloud computing, and big data have now had some primitive results. We expect that WSN will play an important role in these new environments and systems. More precisely, WSN will be the infrastructure rather than just a network. For example, for the issues on big data, more and more data will be collected by sensors, how they are handled by the sensors themselves and how they are uploaded to the backend information systems, be it cloud or not, will have a strong impact on the performance of the information systems we use, such as data collected by the sensors of a weather forecast system. An integrated view will be needed for developing a high performance WSN environment to provide good services to other systems.

Acknowledgment

The authors would like to thank the editors and anonymous reviewers for their valuable comments and suggestions on the paper that greatly improve the quality of the paper.

References

[1] 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.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.

[3] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 6, no. 2, pp. 28–36, 2002.

[4] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[5] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2688–2710, Oct. 2010.

[6] N. K. Suryadevara and S. C. Mukhopadhyay, "Wireless sensor network based home monitoring system for wellness determination of elderly," *IEEE Sensors J.*, vol. 12, no. 6, pp. 1965–1972, Jun. 2012.

[7] A. Gaddam, S. C. Mukhopadhyay, and G. S. Gupta, "Elder care based on cognitive sensor network," *IEEE Sensors J.*, vol. 11, no. 3, pp. 574–581, Mar. 2011.

[8] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proc. Int. Conf. Distrib. Auto. Robot. Syst.*, 2002, pp. 299–308.

[9] A. Bari, S. Wazed, A. Jaekel, and S. Bandyopadhyay, "A genetic algorithm based approach for energy efficient routing in two-tiered sensor networks," *Ad Hoc Netw.*, vol. 7, no. 4, pp. 665–676, 2009.

[10] 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.*, Jan. 2000, pp. 1–10.

[11] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 5, no. 1, 2009, Art. ID 5.

[12] R. V. Kulkarni, A. Forster, and G. K. Venayagamoorthy, "Computational intelligence in wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 1, pp. 68–96, Feb. 2011.

[13] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surveys*, vol. 35, no. 3, pp. 268–308, 2003.
[14] F. W. Glover and G. A. Kochenberger, Eds., *Handbook of Metaheuristics*. Boston, MA, USA: Kluwer Academic Publishers, 2003.
[15] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: A brief survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 2, pp. 262–267, Mar. 2011.
[16] A. Gogu, D. Nace, A. Dilo, and N. Meratnia, "Review of optimization problems in wireless sensor networks," in *Telecommunications Networks—Current Status and Future Trends*, J. H. Ortiz, Ed. Rijeka, Croatia: InTech, 2012, pp. 153–180.
[17] K. Kalpakis, K. Dasgupta, and P. Namjoshi, "Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks," *Comput. Netw.*, vol. 42, no. 6, pp. 697–716, 2003.
[18] L. Zhaohua and G. Mingjun, "Survey on network lifetime research for wireless sensor networks," in *Proc. 2nd IEEE Int. Conf. Broadband Netw. Multimedia Technol.*, Oct. 2009, pp. 899–902.
[19] T.-P. Hong and G.-N. Shiu, "Solving the K-of-N lifetime problem by PSO," *Int. J. Eng., Sci. Technol.*, vol. 1, no. 1, pp. 136–147, 2009.
[20] J. Pan, L. Cai, Y. T. Hou, Y. Shi, and S. X. Shen, "Optimal base-station locations in two-tiered wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 4, no. 5, pp. 458–473, Sep./Oct. 2005.
[21] R. Asorey-Cacheda, A. J. Garcá-Sánchez, F. Garcá-Sánchez, J. García-Haro, and F. J. González-Castano, "On maximizing the lifetime of wireless sensor networks by optimally assigning energy supplies," *Sensors*, vol. 13, no. 8, pp. 10219–10244, 2013.
[22] B.-C. Cheng, G.-T. Liao, R.-Y. Tseng, and P.-H. Hsu, "Network lifetime bounds for hierarchical wireless sensor networks in the presence of energy constraints," *Comput. Netw.*, vol. 56, no. 2, pp. 820–831, 2012.
[23] R. Madan, S. Cui, S. Lall, and A. Goldsmith, "Cross-layer design for lifetime maximization in interference-limited wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 5, no. 11, pp. 3142–3152, Nov. 2006.
[24] K. Akkaya, M. Younis, and W. Youssef, "Positioning of base stations in wireless sensor networks," *IEEE Commun. Mag.*, vol. 45, no. 4, pp. 96–102, Apr. 2007.
[25] N. M. A. Latiff, C. C. Tsimenidis, and B. S. Sharif, "Energy-aware clustering for wireless sensor networks using particle swarm optimization," in *Proc. 18th IEEE Int. Symp. Pers., Indoor, Mobile Radio Commun.*, Sep. 2007, pp. 1–5.
[26] Z. W. Siew, C. H. Wong, C. S. Chin, A. Kiring, and K. T. K. Teo, "Cluster heads distribution of wireless sensor networks via adaptive particle swarm optimization," in *Proc. 4th Int. Conf. Comput. Intell., Commun. Syst., Netw.*, 2012, pp. 78–83.
[27] X. Wang, J.-J. Ma, S. Wang, and D.-W. Bi, "Distributed particle swarm optimization and simulated annealing for energy-efficient coverage in wireless sensor networks," *Sensors*, vol. 7, no. 5, pp. 628–648, 2007.
[28] G. Molina and E. Alba, "Wireless sensor network deployment using a memetic simulated annealing," in *Proc. Int. Symp. Appl. Internet*, 2008, pp. 237–240.
[29] N. Aitsaadi, N. Achir, K. Boussetta, and G. Pujolle, "A Tabu search WSN deployment method for monitoring geographically irregular distributed events," *Sensors*, vol. 9, no. 3, pp. 1625–1643, 2009.
[30] C.-C. Lai, C.-K. Ting, and R.-S. Ko, "An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 3531–3538.
[31] C.-K. Ting and C.-C. Liao, "A memetic algorithm for extending wireless sensor network lifetime," *Inf. Sci.*, vol. 180, no. 24, pp. 4818–4833, 2010.
[32] C.-K. Ting, T.-M. Chou, and C.-C. Liao, "Tabu search with random walk for lifetime extension in wireless sensor networks," in *Proc. Conf. Technol. Appl. Artif. Intell.*, 2012, pp. 119–124.
[33] V. K. Singh and V. Sharma, "Elitist genetic algorithm based energy balanced routing strategy to prolong lifetime of wireless sensor networks," *Chin. J. Eng.*, vol. 2014, Feb. 2014, Art. ID 437625.
[34] P. Zeng and H. Yu, "An ant-based routing algorithm to achieve the lifetime bound for target tracking sensor networks," in *Proc. 25th Joint Conf. IEEE Comput. Commun.*, Apr. 2006.
[35] W.-H. Liao, Y. Kao, and C.-M. Fan, "Data aggregation in wireless sensor networks using ant colony algorithm," *J. Netw. Comput. Appl.*, vol. 31, no. 4, pp. 387–401, 2008.
[36] C.-W. Tsai and J. J. P. C. Rodrigues, "Metaheuristic scheduling for cloud: A survey," *IEEE Syst. J.*, vol. 8, no. 1, pp. 279–291, Mar. 2014.
[37] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
[38] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *J. Optim. Theory Appl.*, vol. 45, no. 1, pp. 41–51, 1985.
[39] F. Glover, "Tabu search—Part I," *ORSA J. Comput.*, vol. 1, no. 3, pp. 190–206, 1989.
[40] F. Glover, "Tabu search—Part II," *ORSA J. Comput.*, vol. 2, no. 1, pp. 4–32, 1990.
[41] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, USA: Univ. of Michigan Press, 1975.
[42] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston, MA, USA: Addison-Wesley, 1989.
[43] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.
[44] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.
[45] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA, USA: MIT Press, 2004.
[46] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Nov./Dec. 1995, pp. 1942–1948.
[47] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. New York, NY, USA: Wiley, 2006.
[48] A. El Rhazi and S. Pierre, "A Tabu search algorithm for cluster building in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 4, pp. 433–444, Apr. 2009.
[49] X.-M. Hu *et al.*, "Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 766–781, Oct. 2010.
[50] Y. Lin, J. Zhang, H.-H. Chung, W. H. Ip, Y. Li, and Y.-H. Shi, "An ant colony optimization approach for maximizing the lifetime of heterogeneous wireless sensor networks," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 3, pp. 408–420, May 2012.
[51] M. Qin and R. Zimmermann, "Studying upper bounds on sensor network lifetime by genetic clustering," in *Proc. Distrib. Comput. Sensor Syst.*, vol. 3560, 2005, p. 408.
[52] T.-P. Hong, G.-N. Shiu, and Y.-C. Lee, "Particle swarm optimization," *Finding Base-Station Locations in Two-Tiered Wireless Sensor Networks by Particle Swarm Optimization*, Rijeka, Croatia: InTech, 2009, pp. 261–274.
[53] G.-N. Shiu, "Allocating base stations in two-tiered wireless sensor networks by the particle swarm optimization," M.S. thesis, National Univ. Kaohsiung, Kaohsiung, Taiwan 2006.
[54] R. Khanna, H. Liu, and H.-H. Chen, "Self-organisation of sensor networks using genetic algorithms," *Int. J. Sensor Netw.*, vol. 1, nos. 3–4, pp. 241–252, 2006.
[55] A. Peiravi, H. R. Mashhadi, and S. H. Javadi, "An optimal energy-efficient clustering method in wireless sensor networks using multi-objective genetic algorithm," *Int. J. Commun. Syst.*, vol. 26, no. 1, pp. 114–126, 2013.
[56] D. Lee, W. Lee, and J. Kim, "Genetic algorithmic topology control for two-tiered wireless sensor networks," in *Proc. 7th Int. Conf. Comput. Sci.*, 2007, pp. 385–392.
[57] H.-S. Seo, S.-J. Oh, and C.-W. Lee, "Evolutionary genetic algorithm for efficient clustering of wireless sensor networks," in *Proc. 6th IEEE Conf. Consum. Commun. Netw. Conf.*, Jan. 2009, pp. 1–5.
[58] S. Hussain, A. W. Matin, and O. Islam, "Genetic algorithm for energy efficient clusters in wireless sensor networks," in *Proc. 3rd Int. Conf. Inf. Technol.*, 2007, pp. 147–154.
[59] T. Agarwal, D. Kumar, and N. R. Prakash, "Prolonging network lifetime using ant colony optimization algorithm on leach protocol for wireless sensor networks," in *Proc. Recent Trends Netw. Commun.*, vol. 90, 2010, pp. 634–641.
[60] B. Singh and D. K. Lobiyal, "A novel energy-aware cluster head selection based on particle swarm optimization for wireless sensor networks," *Human-Centric Comput. Inf. Sci.*, vol. 2, no. 13, pp. 1–18, 2012.
[61] N. M. A. Latiff, C. Tsimenidis, B. S. Sharif, and C. Ladha, "Dynamic clustering using binary multi-objective particle swarm optimization for wireless sensor networks," in *Proc. IEEE 19th Int. Symp. Pers., Indoor, Mobile Radio Commun.*, Sep. 2008, pp. 1–5.
[62] J. Zhang, Y. Lin, C. Zhou, and J. Ouyang, "Optimal model for energy-efficient clustering in wireless sensor networks using global simulated annealing genetic algorithm," in *Proc. Int. Symp. Intell. Inf. Technol. Appl. Workshops*, 2008, pp. 656–660.

[63] A. Chakraborty, K. Chakraborty, S. K. Mitra, and M. K. Naskar, "An energy efficient scheme for data gathering in wireless sensor networks using particle swarm optimization," *J. Appl. Comput. Sci. Math.*, vol. 6, no. 3, pp. 9–13, 2009.

[64] M. Mappar, A. M. Rahmani, and A. H. Ashtari, "A new approach for sensor scheduling in wireless sensor networks using simulated annealing," in *Proc. 4th Int. Conf. Comput. Sci. Converg. Inf. Technol.*, 2009, pp. 746–750.

[65] N. Aitsaadi, N. Achirt, K. Boussetta, and G. Pujolle, "A Tabu search approach for differentiated sensor network deployment," in *Proc. 5th IEEE Consum. Commun. Netw. Conf.*, Jan. 2008, pp. 163–167.

[66] A. P. Bhondekar, R. Vig, M. L. Singla, C. Ghanshyam, and P. Kapur, "Genetic algorithm based node placement methodology for wireless sensor networks," in *Proc. Int. Multiconf. Eng. Comput. Sci.*, vol. 1, 2009, pp. 18–20.

[67] M. Romoozi, M. Vahidipour, M. Romoozi, and S. Maghsoodi, "Genetic algorithm for energy efficient and coverage-preserved positioning in wireless sensor networks," in *Proc. Int. Conf. Intell. Comput. Cognit. Inform.*, 2010, pp. 22–25.

[68] R. Katsuma, Y. Murata, N. Shibata, K. Yasumoto, and M. Ito, "Extending k-coverage lifetime of wireless sensor networks using mobile sensor nodes," in *Proc. IEEE Int. Conf. Wireless Mobile Comput., Netw. Commun.*, Oct. 2009, pp. 48–54.

[69] Y. Qu and S. V. Georgakopoulos, "A centralized algorithm for prolonging the lifetime of wireless sensor networks using particle swarm optimization," in *Proc. IEEE 13th Annu. Wireless Microw. Technol. Conf.*, Apr. 2012, pp. 1–6.

[70] C. Zhao and P. Chen, "Particle swarm optimization for optimal deployment of relay nodes in hybrid sensor networks," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 3316–3320.

[71] N. A. A. Latiff, N. M. A. Latiff, and R. B. Ahmad, "Prolonging lifetime of wireless sensor networks with mobile base station using particle swarm optimization," in *Proc. 4th Int. Conf. Modeling, Simulation, Appl. Optim.*, 2011, pp. 1–6.

[72] C.-C. Liao and C.-K. Ting, "Extending wireless sensor network lifetime through order-based genetic algorithm," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2008, pp. 1434–1439.

[73] C.-C. Liao and C.-K. Ting, "Extending the lifetime of dynamic wireless sensor networks by genetic algorithm," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–8.

[74] F. Y. S. Lin and P. L. Chiu, "A simulated annealing algorithm for energy-efficient sensor network design," in *Proc. 3rd Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw.*, 2005, pp. 183–189.

[75] J. Chen, J. Jia, Y. Wen, D. Zhao, and J. Liu, "Modeling and extending lifetime of wireless sensor networks using genetic algorithm," in *Proc. 1st ACM/SIGEVO Summit Genet. Evol. Comput.*, 2009, pp. 47–54.

[76] R. Montemanni, L. M. Gambardella, and A. K. Das, "The minimum power broadcast problem in wireless networks: A simulated annealing approach," in *Proc. IEEE Wireless Commun. Netw. Conf.*, vol. 4, Mar. 2005, pp. 2057–2062.

[77] Y. Pan, W. Peng, and X. Lu, "A genetic algorithm on multi-sensor networks lifetime optimization," in *Proc. 1st Int. Conf. Wireless Algorithms, Syst., Appl.*, 2006, pp. 295–306.

[78] A. Rahmanian, H. Omranpour, M. Akbari, and K. Raahemifar, "A novel genetic algorithm in LEACH-C routing protocol for sensor networks," in *Proc. 24th Can. Conf. Elect. Comput. Eng.*, May 2011, pp. 1096–1100.

[79] Y. Shen, Y. Li, and Y.-H. Zhu, "Maximizing the lifetime of unreliable sensor networks with delay constraint via genetic algorithm," in *Proc. 6th Int. Conf. Adv. Wireless Sensor Netw.*, 2013, pp. 381–392.

[80] S. Wazed, A. Bari, A. Jaekel, and S. Bandyopadhyay, "Genetic algorithm based approach for extending the lifetime of two-tiered sensor networks," in *Proc. 2nd Int. Symp. Wireless Pervasive Comput.*, 2007, pp. 83–87.

[81] M. B. Ahmed, A. A. Boudhir, and M. Bouhorma, "New routing algorithm based on ACO approach for lifetime optimization in wireless sensor networks," *Int. J. Netw. Syst.*, vol. 1, no. 2, p. 64–67, 2012.

[82] T. Camilo, C. Carreto, J. S. Silva, and F. Boavida, "An energy-efficient ant-based routing algorithm for wireless sensor networks," in *Proc. 5th Int. Conf. Ant Colony Optimization and Swarm Intelligence*, 2006, pp. 49–59.

[83] A. P. Zaheeruddin and M. K. Tiwari, "Prolonging the lifetime of wireless sensor network by exponential node distribution and ant-colony optimization routing," in *Computer Networks & Communications*, vol. 131. New York, NY, USA: Springer, 2013, pp. 709–718.

[84] X.-M. Hu and J. Zhang, "Ant routing optimization algorithm for extending the lifetime of wireless sensor networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2010, pp. 738–744.

[85] X. Hui, Z. Zhigang, and Z. Xueguang, "A novel routing protocol in wireless sensor networks based on ant colony optimization," in *Proc. Int. Conf. Environ. Sci. Inf. Appl. Technol.*, vol. 2, 2009, pp. 646–649.

[86] S. Okdem and D. Karaboga, "Routing in wireless sensor networks using ant colony optimization," in *Proc. 1st NASA/ESA Conf. Sensor Technol. Appl.*, 2006, pp. 401–404.

[87] W.-H. Liao, Y. Kao, and C.-M. Fan, "An ant colony algorithm for data aggregation in wireless sensor networks," in *Proc. Int. Conf. Sensor Technol. Appl.*, 2007, pp. 101–106.

[88] E. Güney, I. K. Altinel, N. Aras, and C. Ersoy, "A Tabu search heuristic for point coverage, sink location, and data routing in wireless sensor networks," in *Proc. 10th Eur. Conf. Evol. Comput. Combinat. Optim.*, 2010, pp. 83–94.

[89] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. 19th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, vol. 2, Mar. 2000, pp. 585–594.

[90] Q. Qiu, Q. Wu, D. Burns, and D. Holzhauer, "Lifetime aware resource management for sensor network using distributed genetic algorithm," in *Proc. Int. Symp. Low Power Electron. Design*, 2006, pp. 191–196.

[91] A. Singh and A. Rossi, "A genetic algorithm based exact approach for lifetime maximization of directional sensor networks," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 1006–1021, 2013.

[92] J.-H. Zhong and J. Zhang, "Ant colony optimization algorithm for lifetime maximization in wireless sensor network with mobile sink," in *Proc. 14th Annu. Int. Conf. Genet. Evol. Comput. Conf.*, 2012, pp. 1199–1204.

[93] Y. Lin, X.-M. Hu, and J. Zhang, "An ant-colony-system-based activity scheduling method for the lifetime maximization of heterogeneous wireless sensor networks," in *Proc. 12th Annu. Conf. Genet. Evol. Comput.*, 2010, pp. 23–30.

[94] K.-S. Low, H. Nguyen, and H. Guo, "Optimization of sensor node locations in a wireless sensor network," in *Proc. 4th Int. Conf. Natural Comput.*, vol. 5, 2008, pp. 286–290.

[95] M. Tang, Y. Xin, J. Li, and J. Zhai, "Nonconvex resource control and lifetime optimization in wireless video sensor networks based on chaotic particle swarm optimization," *Appl. Soft Comput.*, vol. 13, no. 7, pp. 3273–3284, 2013.

[96] M. Cardei and D.-Z. Du, "Improving wireless sensor network lifetime through power aware organization," *Wireless Netw.*, vol. 11, no. 3, pp. 333–340, 2005.

[97] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proc. IEEE Int. Conf. Commun.*, vol. 2, Jun. 2001, pp. 472–476.

Chun-Wei Tsai received the Ph.D. degree in computer science from National Sun Yat-sen University, Kaohsiung, Taiwan, in 2009. He was a Post-Doctoral Fellow with the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan. He was a Faculty Member with the Department of Applied Geoinformatics and the Department of Applied Informatics and Multimedia, Chia Nan University of Pharmacy and Science, Tainan, in 2010 and 2012, respectively. He joined the Department of Computer Science and Information Engineering, National Ilan University, Yilan City, Taiwan, in 2014, as an Assistant Professor. His research interests include metaheuristics, data mining, Internet technology, and combinatorial optimization.

Tzung-Pei Hong received the B.S. degree in chemical engineering from National Taiwan University, Taipei, Taiwan, in 1985, and the Ph.D. degree in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1992. From 1987 to 1994, he was with the Laboratory of Knowledge Engineering, National Chiao Tung University, where he was involved in applying techniques of parallel processing to artificial intelligence. He was an Associate Professor with the Department of Computer Science, National Chung Hua University, Hsinchu, from 1992 to 1994, and the Department of Information Management, I-Shou University, Kaohsiung, Taiwan, from 1994 to 1999. He was a Professor with I-Shou University from 1999 to 2001. He was in charge of the whole computerization and library planning with the National University of Kaohsiung, Kaohsiung, from 1997 to 2000, where he also served as the first Director of the Library and Computer Center from 2000 to 2001, the Dean of Academic Affairs from 2003 to 2006, the Administrative Vice President from 2007 to 2008, and the Academic Vice President in 2010. He is currently a Distinguished Professor with the Department of Computer Science and Information Engineering and the Department of Electrical Engineering. His current research interests include knowledge engineering, soft computing, and granular computing.

Guo-Neng Shiu received the B.S. degree in applied mathematics and the M.S. degree in electrical engineering from the National University of Kaohsiung, in 2004 and 2006, respectively. He is currently an Information Engineer with the Electronics Group, Formosa Plastics Corporation, Taiwan. His research interests include particle swarm optimization, genetic algorithms, wireless sensor network, and fuzzy theory.