# An Ant Colony Optimization Approach for Maximizing the Lifetime of Heterogeneous Wireless Sensor Networks

Ying Lin[1], Jun Zhang[1], Henry Shu-Hung Chung[2], Wai Huang Ip[3], Yun Li[4], and Yu-Hui Shi[5]

*Abstract* – Maximizing the lifetime of wireless sensor networks (WSNs) is a challenging problem. Although some methods exist for addressing the problem in homogeneous WSNs, research on this problem in heterogeneous WSNs is progressed at a slow pace. Inspired by the promising performance of ant colony optimization (ACO) in solving combinatorial problems, this paper proposes an ACO-based approach that can maximize the lifetime of heterogeneous WSNs. The methodology is based on finding the maximum number of disjoint connected covers that satisfy both sensing coverage and network connectivity. A construction graph is designed with each vertex denoting the assignment of a device in a subset. Based on pheromone and heuristic information, the ants seek an optimal path on the construction graph to maximize the number of connected covers. The pheromone serves as a metaphor of the search experiences in building connected covers. The heuristic information is used to reflect the desirability of device assignments. A local search procedure is designed to further improve the search efficiency. The proposed approach has been applied to a variety of heterogeneous WSNs. The results show that the approach is effective and efficient in finding high-quality solutions for maximizing the lifetime of heterogeneous WSNs.

*Index Terms* – Ant colony optimization (ACO), connectivity, coverage, network lifetime, wireless sensor networks (WSNs)

[1] Y. Lin and J. Zhang (corresponding author) are with the Department of Computer Science, Sun Yat-sen University, Guangzhou, P.R. China, the Key Laboratory of Digital Life (Sun Yat-sen University), Ministry of Education, P.R. China and the Key Laboratory of Software Technology (Sun Yat-sen Uni- versity), Education Department of Guangdong Province, P.R. China (e-mail: junzhang@ieee.org).
[2] H. S.-H. Chung is with the Department of Electronic Engineering, City Uni- versity of Hong Kong, Kowloon Tong, Hong Kong.
[3] W. H. IP is with the Hong Kong Polytechnic University, Hong Kong.
[4] Y. Li is with the University of Glasgow, Glasgow, U.K.
[5] Y.-H. Shi is with the Xi'an Jiaotong-Liverpool University, Suzhou, P.R. China.

## I. Introduction

With the advance of electronics and communication technology, real-time monitoring such as battlefield surveillance [1], environment supervision [2], and traffic control [3] has become a reality. These applications generally require the use of wireless sensor networks (WSNs) and their quality of services is strongly dependent on the network performance. A fundamental criterion for evaluating a WSN is the network lifetime [4], which is defined as the period that the network satisfies the application requirements. Since most devices of WSNs are powered by non-renewable batteries, studies of prolonging the network lifetime have become one of the most significant and challenging issues in WSNs.

The existing methods for prolonging the lifetime of WSNs focus on the issues of device placement [5], data processing [6], routing [7]-[14], topology management [15], and device control [16]-[18]. Among them, the device control approach that schedules the devices' sleep/wakeup activities has shown to be promising [19], [20]. Devices in a WSN carry out both monitoring and communication tasks. The monitoring task requires devices to offer satisfying sensing coverage to the target. The communication task demands devices to form a connected network for collecting and disseminating information via radio transmissions. In a WSN where devices are densely deployed, a subset of the devices can already address the coverage and connectivity issues. The rest of the devices can be switched to a sleep state for conserving energy. Therefore, the lifetime of a WSN can be prolonged by planning the active intervals of devices. At every point during the network lifetime, the active devices must form a *connected cover* for fulfilling sensing coverage and network connectivity.

A number of methods have been proposed for finding one connected cover from a WSN. The connected cover obtained may be optimal under certain criteria, such as minimum size [21], [22] or minimum energy consumption [23], [24]. Nevertheless, generating a sequence of optimal connected covers by repeating the above methods may not lead to lifetime maximization. Maximizing the number of connected covers is a more direct way for maximizing the network lifetime.

The problem of finding the maximum number of connected covers is difficult because each connected cover must fulfill sensing coverage and network connectivity simultaneously. Its subproblem of maximizing the number of subsets that fulfill sensing coverage is already in the non-deterministic polynomial time (NP-complete) complexity class [25]. Many methods focus on solving the above subproblem but ignore the issue of connectivity [26]-[29]. These methods is able to maximize the lifetime of WSNs while maintaining both sensing coverage and network connectivity with a premise that the devices are identical and have a transmission range at least twice of the sensing range. However, they cannot ensure the network connectivity when the required premise is not satisfied. Their robustness in real-world applications thus cannot be guaranteed. Zhao *et al.*[30] proposed a greedy algorithm that addressed both sensing coverage and network connectivity, but the algorithm can only handle the coverage of discrete points. It is also difficult to extend the algorithm to heterogeneous WSNs that comprise different types of devices.

Recent studies have shown that heterogeneous WSNs have inherent advantages in terms of reliability, robustness, and energy efficiency [31], [32]. A growing trend of heterogeneous designs has also been witnessed in a number of applications [33], [34]. In order to prolong the lifetime of heterogeneous WSNs, novel device placement methods [35], routing protocols [36], and topology management strategies [37] have been introduced. The device control approach for planning the activities of different devices, however, remains unexplored. In this paper, a common type of heterogeneous WSNs is considered and a novel activity planning approach for maximizing the network lifetime is proposed. The approach can be used in both cases of discrete point coverage and area coverage. We focus on the more general case, i.e., area coverage, in the following study.

The considered heterogeneous WSNs comprise two types of devices: sensors and sinks. The sensors monitor the target and transmit the monitoring results to the sinks. The sinks relay the monitoring results to the destination (e.g., data processing center). Therefore, a connected cover in the heterogeneous WSNs must satisfy the following three constraints: 1) the sensors form complete coverage to the target; 2) all the monitoring results obtained by the sensors are transmitted to the sinks; 3) the sinks compose a connected wireless network. These three constraints interact with each other as the second constraint involves both sensors and sinks. Finding the maximum number of connected covers is thus more difficult than either the problem of maximizing the number of sensor subsets under the coverage constraint or the problem of maximizing the number of sink subsets under the connectivity constraint. It is unlikely to have a polynomial-time deterministic algorithm for solving the problem. Heuristic methods are more promising for finding high-quality solutions.

Ant colony optimization (ACO) is a well-known metaheuristic inspired by the foraging behavior of real ants [38], [39]. In ACO, ants are stochastic constructive procedures that build solutions while walking on a construction graph. Such constructive search behavior makes ACO suitable for solving combinatorial optimization problems [40]. Besides, ACO utilizes search experiences (represented by *pheromone*) and domain knowledge (represented by *heuristic information*) to accelerate the search process. ACO algorithms have been successfully applied to a number of industrial and scientific problems [41]-[47]. In the fields of WSNs, ACO-based routing algorithms have been used for improving the power efficiency in unicasting [8]-[11], broadcasting [12], [13], and data gathering [14]. Different from the above ACO algorithms that focus on the routing issue in homogeneous WSNs, this paper proposes an ACO-based approach for maximizing the lifetime of heterogeneous WSNs by finding the maximum number of connected covers.

The proposed ACO-based approach for maximizing the number of connected covers

(ACO-MNCC) first transforms the search space of the problem into a construction graph. Each vertex in the graph denotes an assignment of a device in a subset. Heuristic information is associated to each assignment for measuring its utility in reducing constraint violations. Pheromone is deposited between every two devices to record the historical desirability of assigning them to the same subset. In each iteration, the number of subsets is adaptively determined as one plus the number of connected covers in the best-so-far solution. The ants thus concentrate on finding one more connected cover and avoid constructing subsets excessively. A local search procedure is designed to refine the solutions by reassigning redundant devices. The ACO-MNCC is applied to thirty-three heterogeneous WSNs with different characteristics. Experimental results validate the effectiveness and efficiency of the proposed approach.

The remainder of this paper is organized as follows. Section II defines the problem addressed in this paper. A method for estimating an upper bound of the number of connected covers is also given. Section III is devoted to the development of ACO-MNCC. Experimental results and discussions are presented in Section IV. Section V draws a conclusion and provides guidelines for future research.

## II. Preliminary

In this section, the problem of finding the maximum number of disjoint connected covers is defined. We also introduce a method for estimating an upper bound of the number of connected covers in a WSN.

### *A. Problem Definition*

Randomly deploy a set of sensors $\boldsymbol{SE} = \{SE_1, SE_2, \ldots, SE_{|\boldsymbol{SE}|}\}$ and a set of sinks $\boldsymbol{SI} = \{SI_1, SI_2, \ldots, SI_{|\boldsymbol{SI}|}\}$ in an $L\times W$ area ($|\cdot|$ denotes the size of a set). Suppose the sensors have a sensing range $r_s$ and a transmission range $r_t$. The sinks have a transmission range $R_t$ larger than $r_t$. By denoting the number of disjoint connected covers as $C$, the problem can be stated as maximizing $C$, with each connected cover $\boldsymbol{S}_i$ ($\boldsymbol{S}_i \subseteq \boldsymbol{SE} \cup \boldsymbol{SI}$, $i$=1,2,…,$C$) subject to

*1) The coverage constraint*, which requires the sensors in $\boldsymbol{S}_i$ to fully cover a target area $\boldsymbol{T}$. In other words, for any given point $P \in \boldsymbol{T}$, at least one sensor $SE_j \in \boldsymbol{S}_i$ satisfies

$$\left\| SE_j - P \right\| \le r_s, \tag{1}$$

where $||x-y||$ represents the distance between the two points $x$ and $y$.

*2) The collection constraint*, which requires the sinks to collect all the monitoring results obtained by the sensors in the same subset. Assume that the sensors cannot relay data. For each sensor $SE_j \in \boldsymbol{S}_i$, at least one sink $SI_k \in \boldsymbol{S}_i$ has

$$\left\| SE_j - SI_k \right\| \le r_t. \tag{2}$$

*3) The routing constraint*, which requires the sinks in $\boldsymbol{S}_i$ to form a connected network for transmitting the collected monitoring results to the destination. Mathematically, this constraint can be written as follows. Between any two sinks $SI_j$, $SI_k \in \boldsymbol{S}_i$, there is a path $\wp$ satisfying

$$\max_{(SI_x, SI_y) \in \wp} \left\| SI_x - SI_y \right\| \le R_t. \tag{3}$$

Fig. 1 shows an example of a connected cover. It can be observed from the above three constraints that the connected cover addresses both sensing coverage and network connectivity. The tasks of monitoring and communication can be successfully carried out.

*B. The Upper Bound of C*

In a WSN, the maximum number of connected covers cannot exceed the maximum number of full cover subsets that satisfy the coverage constraint. Therefore, the maximum number of full cover subsets can be used as the upper bound of the number of connected covers, i.e., the upper bound of $C$. Although finding the maximum number of full cover subsets is an NP-complete problem [21], we can estimate the upper bound of the number of full cover subsets with the following method.

When all the deployed sensors are active, the target area is logically divided into a number of *fields*, each of which is a set of points covered by the same set of sensors [27]. Fig. 2 shows an example of five sensors forming fifteen fields. Among all the fields, the ones covered by the minimum number of sensors are denoted as the *critical fields* (e.g., $F_1$, $F_3$, $F_5$, $F_{13}$, and $F_{15}$ in Fig. 2). If a set of sensors can form complete coverage to the target area, each critical field is covered by at least one sensor. The number of sensors covering a critical field can be estimated as the upper bound of the number of full cover subsets [20], [27], [48], [49]. Therefore, the minimum number of sensors covering a field, which is denoted by $\hat{C}$, can be used as the upper bound of $C$.

## III. ACO-MNCC

In this section, we introduce the ACO-MNCC approach for maximizing the number of connected covers in a heterogeneous WSN. First, the objective function is formulated. The ants' search behavior is then described. An introduction of the local search procedure follows. Finally, the process of the whole algorithm will be summarized. To facilitate the descriptions, the notations used in this section are tabulated in Table I.

### *A. The Objective Function*

Define a solution to the problem as $\boldsymbol{S} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_N\}$, where $\boldsymbol{S}_i \subseteq \boldsymbol{SE} \cup \boldsymbol{SI}$ denotes a subset composed of $U_i$ sensors and $V_i$ sinks, $i$=1,2,…,$N$, and $N$ is the number of subsets. Each subset is disjoint with the others and the union of the $N$ subsets equals the set of $\boldsymbol{SE} \cup \boldsymbol{SI}$. Three criteria are designed to evaluate each subset's satisfaction to the three constraints in Section II-A. The objective value of the solution is calculated on the basis of the three criteria.

*1) Criterion for the Coverage Constraint.* The coverage percentage achieved by the sensors in $\boldsymbol{S}_i$ can be directly used as the criterion for the coverage constraint. If the target is a group of discrete points, the coverage percentage is the proportion of covered points. If the target is an area, the coverage percentage can be calculated based on the idea of fields. This paper uses area coverage as a study case and the coverage percentage $\kappa_i$ is the ratio of the number of covered fields to the number of existing fields, i.e.,

$$\kappa_i = \frac{\left| \bigcup_{SE_j \in \boldsymbol{S}_i} \boldsymbol{F}_j \right|}{|\boldsymbol{F}|}, \tag{4}$$

where $\boldsymbol{F}_j \subseteq \boldsymbol{F}$ denotes the set of fields covered by a sensor $SE_j \in \boldsymbol{S}_i$, $i=1,2,\ldots,N$.

*2) Criterion for the Collection Constraint.* Term a sensor with at least one sink in its transmission range as a collected sensor. Obviously, a subset with a larger proportion of collected sensors can better satisfy the collection constraint. The proportion $\chi_i$ of collected sensors in $\boldsymbol{S}_i$ can be employed as the criterion, i.e.,

$$\chi_i = \frac{H_i}{U_i}, \tag{5}$$

where $H_i$ is the number of collected sensors in $\boldsymbol{S}_i$.

*3) Criterion for the Routing Constraint.* Consider a communication graph $G_i$, where the $V_i$ sinks in $\boldsymbol{S}_i$ compose the vertex set and the edge set is $\{(SI_j, SI_k): \|SI_j - SI_k\| \le R_{\mathrm{t}}, SI_j, SI_k \in \boldsymbol{S}_i, j \ne k\}$. The sinks in $\boldsymbol{S}_i$ constitute a connected network if and only if $G_i$ is a connected graph. Based on the graph theory, the connectivity of a graph can be measured by the relative size $\lambda_i$ of its largest connected subgraph [50]. The criterion for the routing constraint is defined as

$$\lambda_i = \frac{B_i}{V_i}, \tag{6}$$

where $B_i$ is the number of sinks in the largest connected subgraph of $G_i$.

The values of the above three criteria are all in the range of [0,1]. A larger value indicates a smaller violation of the constraint. We use the average value of the three criteria, $(\kappa_i+\chi_i+\lambda_i)/3$, to summarize how well the set $\boldsymbol{S}_i$ satisfies the three constraints. If the average value equals one, i.e., $\kappa_i=\chi_i=\lambda_i=1$, $\boldsymbol{S}_i$ fulfills all the three constraints and becomes a connected cover. After applying the three criteria to evaluate all the $N$ subsets, the objective value of the solution $\boldsymbol{S}$ can be calculated as

$$\Phi(\boldsymbol{S}) = \omega_1 \sum\nolimits_{i=1}^{N} (\kappa_i + \chi_i + \lambda_i)/3 + \omega_2 C, \tag{7}$$

where $\omega_1$, $\omega_2 > 0$ are predefined weights and $C$ is the number of connected covers in $\boldsymbol{S}$.

It can be observed that the objective function has two components. The first component summarizes the constraint violations of all the subsets. The second component awards the objective value based on the number of connected covers. Since the goal of ACO-MNCC is to find a solution that maximizes the number of connected covers, the objective value should grow as $C$ increases. For ensuring this, the values of $\omega_1$ and $\omega_2$ need to satisfy

$$\frac{\omega_2}{\omega_1} \geq \hat{C}. \tag{8}$$

Please refer to the Appendix for the proof of the above proposition. In this paper, we set $\omega_1=1$ and $\omega_2 = \hat{C}$.

*B. Search Behavior of Ants*

In ACO, an ant's search behavior is mainly influenced by three components: the construction graph, the solution construction rule, and the pheromone management. The following subsections describe the three components.

*1) The Construction Graph*

Fig. 3 shows an example of the construction graph with five sensors and three sinks ($|\boldsymbol{SE}|=5$ and $|\boldsymbol{SI}|=3$). It can be observed that the vertices of the construction graph are arranged into an $N_t$ by ($|\boldsymbol{SE}|+|\boldsymbol{SI}|$) matrix, where $N_t$ is the number of available subsets in iteration $t$. Each vertex $v_{ij}$ ($i$=1,2,…,$N_t$, $j$=1,2,…, $|\boldsymbol{SE}|+|\boldsymbol{SI}|$) in the construction graph denotes a device assignment to a subset. If $j$ is smaller than $|\boldsymbol{SE}|$, $v_{ij}$ represents the assignment of sensor $SE_j$ to $\boldsymbol{S}_i$. Otherwise, $v_{ij}$ is to assign the sink $SI_{j-|\boldsymbol{SE}|}$ to $\boldsymbol{S}_i$. Every pair of vertices in the adjacent columns is connected with an undirected arc, which indicates a potential route of ants. An ant following the arcs throughout the construction graph selects exactly one vertex from each column, resulting in a solution with $N_t$ disjoint subsets. Each subset $\boldsymbol{S}_i$ is composed of the devices corresponding to the selected vertices on row $i$, $i$=1,2,…,$N_t$. Still take Fig. 3 as an example. The ant's path ($v_{11}$, $v_{42}$, $v_{23}$, $v_{34}$, $v_{15}$, $v_{36}$, $v_{27}$, $v_{48}$) (denoted by black arrows) represents a solution $\boldsymbol{S}=\{\boldsymbol{S}_1,\boldsymbol{S}_2,\boldsymbol{S}_3,\boldsymbol{S}_4,\boldsymbol{S}_5\}$, where $\boldsymbol{S}_1=\{SE_1,SE_5\}$, $\boldsymbol{S}_2=\{SE_3,SI_2\}$, $\boldsymbol{S}_3=\{SE_4,SI_1\}$, $\boldsymbol{S}_4=\{SE_2,SI_3\}$, and $\boldsymbol{S}_5=\varnothing$ are the five subsets in the solution.

A novelty of the above construction graph lies in the determination of $N_t$. Assume that the maximum number of connected covers is already known as $C_{\max}$. A reasonable setting of $N_t$ is $N_t = C_{\max}+1$, in which the extra one denotes a subset for the devices that are not included in the $C_{\max}$ connected covers. However, since the value of $C_{\max}$ is the optimization objective and is unknown in advance, the above setting is infeasible. It is also undesirable to replace $C_{\max}$ with the upper bound $\hat{C}$ because an overlarge $N_t$ can lead to excessive subset building and cause waste on computational cost. In order to address the problem, we adapt $N_t$ to the search process of ACO-MNCC, i.e.,

$$N_t = C^{\mathrm{bs}}+1, \tag{9}$$

where $C^{\mathrm{bs}}$ is the number of connected covers in the best-so-far solution $\boldsymbol{S}^{\mathrm{bs}}$ until iteration $t$. This way, $N_t$ never exceeds $C_{\max}+1$ and the approach avoids building subsets excessively. Besides, this setting makes the ants concentrate on finding one more connected cover than $\boldsymbol{S}^{\mathrm{bs}}$, which helps improve the search efficiency by setting an explicit goal.

*2) The Construction Rule*

The construction rule of ACO guides the ants to build their own solutions by selecting vertices from the construction graph. In ACO-MNCC, the construction rule guides an ant to assign each device to a subset. The core of the construction rule, as in other ACO algorithms, is the design of pheromone and heuristic information.

In ACO-MNCC, the pheromone is deposited between every two devices to record the historical desirability for assigning them to the same subset. Heavier pheromone indicates higher desirability. When assigning an unassigned device to a subset, the ants consider the average pheromone between this device and the devices already in the subset. Suppose the pheromone between two devices $J$ and $K$ is denoted by $\tau(J,K)$. The average pheromone between an unassigned device $J$ and the existing devices in a subset $\boldsymbol{S}_i$ ($i$=1,2,…, $N_t$) can be calculated as

$$\mathrm{T}_i(J)=\begin{cases}\dfrac{1}{|\boldsymbol{S}_i|}\sum_{K\in\boldsymbol{S}_i}\tau(J,K), & \text{if } \boldsymbol{S}_i\neq\varnothing \\ \tau_0, & \text{otherwise}\end{cases}, \tag{10}$$

where $\tau_0=(1+\hat{C})^{-1}$ is the initial pheromone value. $\mathrm{T}_i(J)$ can show the historical information about grouping device $J$ and the existing devices in $\boldsymbol{S}_i$ into the same subset. Note that different from many ACO algorithms, ACO-MNCC does not deposit pheromone on the components in the construction graph. This is because the key of the considered problem is the relative relation among devices but not the absolute relation between devices and subsets.

The heuristic information in ACO-MNCC is associated to each device assignment for measuring the improvement that the device can bring to the subset. The sensors directly influence the coverage of the target area. The heuristic information for the sensors is thus based on the increment in the coverage percentage. The sinks affect the violation of the collection constraint. The heuristic information for the sinks is therefore related to the change in the proportion of collected sensors. Mathematically, the heuristic value for assigning an unassigned device $J$ to a subset $\boldsymbol{S}_i$ ($i$=1, 2, …, $N_t$) can be formulated as

$$\eta_i(J) = \begin{cases} \kappa_i' - \kappa_i, & \text{if } J \text{ is a sensor} \\ \chi_i' - \chi_i, & \text{if } J \text{ is a sink} \end{cases}, \tag{11}$$

where $\kappa_i'$ and $\chi_i'$ denote the coverage percentage and the proportion of collected sensors after $J$ joins $\boldsymbol{S}_i$. This way, the heuristic information biases the ants to choose the subsets in which the devices are more helpful for reducing constraint violations.

With the above design of pheromone and heuristic information, the probability for assigning an unassigned device $J$ to a subset $\boldsymbol{S}_i$ ($i$=1,2,…, $N_t$) is calculated by

$$p_i(J) = \frac{\mathrm{T}_i(J)[\eta_i(J)]^\beta}{\sum_{k=1}^{N_t} \mathrm{T}_k(J)[\eta_k(J)]^\beta}, \tag{12}$$

where $\beta$>0 is a predefined parameter that controls the influence of heuristic information. An ant chooses subset $i$ for device $J$ based on the following construction rule

$$i = \begin{cases} \arg\max_{1 \le k \le N_t} p_k(J), & \text{if } q \le q_0 \\ I, & \text{otherwise} \end{cases}, \tag{13}$$

where $q_0 \in (0,1)$ is a predefined parameter, $q \in [0,1]$ is a uniform random number, and $I$ is the index of the subset selected by the roulette wheel selection [51] based on the probability distribution given by (12).

In a special case that the unassigned device $J$ fails to improve the coverage percentage or satisfaction to the collection constraint of any subset, the values of $\eta_i(J)$ are all zero for $i$=1,2,…, $N_t$. Consequently, the probabilities are also zero and the construction rule in (13) cannot be applied. To address this special case, we design a complementary rule that assigns $J$ either to one of the $N_t$ subsets or to a new subset $\boldsymbol{S}_{N_t+1}$. The details of the complementary rule are as follows: given a uniform random number $q$, device $J$ is assigned to subset $i$ according to

$$i = \begin{cases} rand(1, N_t), & \text{if } q \le q_0 \text{ or } N_t = \hat{C} + 1 \\ N_t + 1, & \text{otherwise} \end{cases}, \tag{14}$$

where $rand(1,N_t)$ generates a random integer in $[1,N_t]$. Note that the new subset will not be created if $N_t$ is already equal to $\hat{C}+1$, preventing the ants from building excessive subsets.

*3) The Pheromone Management*

The pheromone management in ACO-MNCC comprises a local pheromone updating rule and a global pheromone updating rule. The local updating rule is performed after an ant $a$ has finished building its solution $\boldsymbol{S}^{(a)}$, where $a=1,2,\ldots,m$, and $m$ is the number of ants. Suppose $\boldsymbol{S}^{(a)}$ comprises $N^{(a)}$ subsets. The pheromone between any two devices $J$ and $K$ in the same subset $\boldsymbol{S}_i^{(a)}$ ($i=1,2,\ldots,N^{(a)}$) is updated by

$$\tau(J,K)=(1-\rho)\cdot\tau(J,K)+\rho\cdot\tau_0, \tag{15}$$

where $\rho\in(0,1)$ is the evaporation rate of the local pheromone update.

The global updating rule is performed in the end of each iteration after the best-so-far solution $\boldsymbol{S}^{\text{bs}}$ is determined. Instead of updating the pheromone between every pair of devices in each subset, this rule only updates the pheromone between two devices that are in the same connected cover of $\boldsymbol{S}^{\text{bs}}$. Suppose $\boldsymbol{S}^{\text{bs}}$ comprises $N^{\text{bs}}$ subsets. The global updating rule can be written as

$$\tau(J,K)=\begin{cases}(1-\xi)\cdot\tau(J,K)+\xi\cdot\Delta\tau, & \text{if } J,K\in\boldsymbol{S}_i^{\text{bs}} \text{ and } \kappa_i=\chi_i=\lambda_i=1\\ \tau(J,K), & \text{otherwise}\end{cases}, \tag{16}$$

where $\xi\in(0,1)$ is the evaporation rate in the global pheromone update, $i=1,2,\ldots,N^{\text{bs}}$, and $\Delta\tau$ is the pheromone increment calculated by

$$\Delta\tau=\frac{\Phi(\boldsymbol{S}^{\text{bs}})}{(1+\hat{C})^2}. \tag{17}$$

From the two updating rules, we know that the pheromone remains unchanged until the iteration in which $\boldsymbol{S}^{\text{bs}}$ finds the first connected cover (i.e., $C^{\text{bs}}\geq 1$). Since that iteration, the objective value $\Phi(\boldsymbol{S}^{\text{bs}})$ calculated by (7) exceeds $1+\hat{C}$ and $\Delta\tau$ is larger than $\tau_0$. The global pheromone update is thus able to reinforce the pheromone between each pair of devices in every connected cover of $\boldsymbol{S}^{\mathrm{bs}}$. More ants will then be attracted to exploit the neighborhood of $\boldsymbol{S}^{\mathrm{bs}}$. In the following iterations, if the corresponding pairs of devices are again assigned to one subset, the reinforced pheromone will be reduced by the local pheromone update but still remain above $\tau_0$. The ants that build solutions afterwards benefit from the reduced pheromone and gain more opportunities to explore the search space. With the two updating rules, ACO-MNCC is able to strike a balance between exploration and exploitation.

*C. The Local Search Procedure*

In the iterations when the best-so-far solution $\boldsymbol{S}^{\mathrm{bs}}$ is updated, a local search procedure is performed to refine $\boldsymbol{S}^{\mathrm{bs}}$. Before describing the implementation of the local search, we first introduce a definition termed 'redundant device'. A redundant device can be removed from the subset to which it used to belong without reducing the coverage and connectivity of the subset. For example, a redundant sensor of a subset $\boldsymbol{S}_i$ can be removed without reducing $\kappa_i$, whereas a redundant sink of $\boldsymbol{S}_i$ can be removed without reducing $\chi_i$ and $\lambda_i$, $i$=1,2,…$N$.

Based on the idea of redundant devices, the local search uses two modules for scheduling sensors and sinks respectively. The pseudo-code of the two modules is shown in Fig. 4. The sensor module (from lines 2 to 24 in Fig. 4) examines the coverage of critical fields for the subsets of $\boldsymbol{S}^{\mathrm{bs}}$. If a subset $\boxed{\phantom{S}}$ fails to cover a critical field $F_v$, the sensor module seeks a redundant sensor that covers $F_v$ in the other subsets and moves it to $\boxed{\phantom{S}}$. The sink module (from lines 25 to 35 in Fig. 4) is performed after evaluating the new $\boldsymbol{S}^{\mathrm{bs}}$ generated by the sensor module. In the sink module, all the redundant sinks of every connected cover are moved to the subsets that have not satisfied all the constraints. If all the subsets in $\boldsymbol{S}^{\mathrm{bs}}$ are already connected covers, the redundant sinks will be moved to a new subset $\boxed{\phantom{S_N}}$. Note that the changes made by the local search procedure are only accepted when they indeed improve $\boldsymbol{S}^{\mathrm{bs}}$. Otherwise, $\boldsymbol{S}^{\mathrm{bs}}$ will be reverted to the state before the local search.

*D. A Summary of ACO-MNCC*

As shown in the overall flowchart of Fig. 5 (a), the ACO-MNCC approach can be summarized as follows.

At the beginning of the approach, the control parameters including $m$, $\beta$, $q_0$, $\rho$, and $\xi$ are first given. The upper bound $\hat{C}$ is calculated and the pheromone is initialized as $\tau_0$. After the initialization, the $m$ ants build their own solutions following the instructions in Section III-B. Fig. 5 (b) depicts the detailed construction procedure of one ant. After all the ants finish building solutions, if the best-so-far solution $\boldsymbol{S}^{\mathrm{bs}}$ has been updated, the local search procedure will be carried out to refine $\boldsymbol{S}^{\mathrm{bs}}$. The global pheromone updating rule is then applied to update the pheromone between each pair of devices in the connected covers of $\boldsymbol{S}^{\mathrm{bs}}$. An iteration of ACO-MNCC is completed after the global pheromone update. The approach reports $\boldsymbol{S}^{\mathrm{bs}}$ as the result if the termination criterion has been satisfied. Otherwise, it starts a new iteration and dispatches ants to build solutions based on the updated pheromone.

## IV. Experiments and Discussions

In this section, a series of experiments are performed to evaluate the performance of ACO-MNCC. Since the proposed approach is the first algorithm for maximizing the number of connected covers in heterogeneous WSNs, a greedy algorithm that applies the same heuristic information as ACO-MNCC is used for comparison. The effectiveness of pheromone, heuristic information, and local search procedure in the ACO-MNCC is also investigated.

*A. Test Cases*

Three sets of heterogeneous WSNs with different scales and redundancy are employed in the experiments. In Set A, WSNs are generated by randomly deploying sensors and sinks in a 50 by 50 rectangle. Table II tabulates the settings of these networks, including the scale |***SE***| and |***SI***|, $r_s$ and $r_t$ of sensors, $R_t$ of sinks, and the upper bound $\hat{C}$ of the number of connected covers. In the following study, we will show that ACO-MNCC is able to find a solution with $\hat{C}$ connected covers for each case. Therefore, the value of $\hat{C}$ is the maximum number of connected covers for each case in Set A.

Based on an optimal solution of a case in Set A, a new case can be generated by removing redundant devices from the connected covers in the solution. Herein, we remove every device with a predefined probability $\varphi$. The device will be resumed if it is found necessary in its original subset. After checking all the devices, the remaining devices form a new network. Apparently, the new network has less redundancy than the original one, but the maximum number of connected covers remains unchanged because deleting redundant devices from a connected cover doesn't reduce the sensing coverage and network connectivity. The WSNs in Sets B and C are obtained by performing the above process on the WSNs of Set A with $\varphi$=0.3 and $\varphi$=0.6, respectively. The settings of WSNs in Sets B and C are also displayed in Table II. Since Sets B and C are derived from Set A, the three cases on the same row of Table II have identical settings in $r_s$, $r_t$, $R_t$, and $\hat{C}$ but are different in the network scale |***SE***| and |***SI***|.

*B. Experimental Settings*

During the experiments, the control parameters of ACO-MNCC are set as $m$=10, $q_0$=0.9, $\beta$=2, and $\rho$=$\xi$=0.1 [40]. The termination criterion is set as 20,000 function evaluations (FEs). Using the same heuristic information as ACO-MNCC, the greedy algorithm always assigns a device to a subset where the device can bring the greatest improvement in sensing coverage or network connectivity. If two or more subsets share the highest heuristic value, devices are randomly assigned to one of them. A detailed description of the greedy algorithm can be referred to Fig. 6.

As the ACO-MNCC and the greedy algorithm both contain certain randomness in the search process, the two algorithms are performed thirty independent runs on each test case for fair comparison. All the experiments are carried out on a Dell computer with Intel® Core™2 Quad CPU at 2.40GHz and RAM of 2GB.

*C. Comparison with the Greedy Algorithm*

The results of ACO-MNCC and the greedy algorithm on Sets A, B, and C are given in Tables III, IV, and V, respectively. ‘Best’, ‘Worst’, and ‘Avg’ denote the maximum, minimum, and average numbers of connected covers in the solutions found in the thirty runs, respectively. Better results between ACO-MNCC and the greedy algorithm are **bolded**.

From Table III, it can be observed that the best results of ACO-MNCC can reach the upper bound of each WSN in Set A. Such a phenomenon implies that the optimal solution of each WSN in Set A has exactly $\hat{C}$ connected covers. We use ‘SR’ to indicate the percentage of finding an optimal solution in thirty runs. ACO-MNCC can obtain SR=100% for all WSNs in Set A, whereas the greedy algorithm only has an average SR lower than 90%. On A1, A9, A10, and A11, the greedy algorithm cannot find the optimal solution even once.

The advantage of ACO-MNCC against the greedy algorithm further enlarges on WSNs with fewer redundant devices. According to the results in Table IV, ACO-MNCC can maintain 100% successful rates for finding an optimal solution in all WSNs in Set B except for B10. The greedy algorithm can only find an optimal solution on B3, B7, and B9 with a successful rate lower than 50%. Results of Set C in Table V show that the greedy algorithm can find one or even no connected cover on C1, C2, C4, C6, and C9. In contrast, ACO-MNCC can still obtain the optimal or near-optimal solutions in these networks.

Concluded from the above, ACO-MNCC comprehensively outperforms the greedy algorithm. For all tested networks, ACO-MNCC is able to find the optimal solution or a near-optimal solution with an error no more than two connected covers. Such a promising performance confirms the effectiveness and efficiency of the proposed approach.

In order to further analyze the computational time of the approach, Table VI tabulates the average CPU time for ACO-MNCC to obtain the above results. From the table, it can be known that the CPU time is not only affected by the network scale but also its redundancy. Removing redundant devices reduces the network scale. However, the decrement in redundancy may also increase the difficulty in solving the problem. When the time cost by the increased difficulty overtakes the time saved by the decreased scale, ACO-MNCC may need to spend more time on finding an optimal or near-optimal solution. Nevertheless, ACO-MNCC can solve most test cases in a short period of time. Among the thirty-three tested networks, ACO-MNCC can find satisfying solutions for twenty networks within one second. Twelve of the remaining thirteen networks can be solved within one minute. Only C8, which has not only a large scale but also small redundancy, takes the ACO-MNCC more than one minute to solve.

*D. Effectiveness of Pheromone, Heuristic Information, and Local Search*

Pheromone, heuristic information, and local search are three important components in the proposed ACO-MNCC. In this part, we validate their effectiveness by comparing the results of ACO-MNCC with its variants without these components. The variants without pheromone, heuristic information, and local search are termed ACO-noPhe, ACO-noHeu, and ACO-noLS, respectively. The settings of these variants are exactly the same as ACO-MNCC except that one of the three components is not applied. The following study takes C2, C4, C5, and C8 as an example. The situations on the other networks are similar.

Table VII shows the results of the four algorithms averaged over thirty independent runs. It can be observed that ACO-MNCC obtains the best results, followed by ACO-noLS and ACO-noPhe, whereas ACO-noHeu performs the worst. The difference between ACO-noPhe and ACO-noLS is insignificant, but there is a large gap between ACO-noHeu and the other three algorithms. The advantages of ACO-MNCC over the other three algorithms confirm that the three components are indeed effective for finding high-quality solutions. Moreover, the results in Table VII show that the heuristic information is significantly influential to the algorithm performance. However, the algorithm comparison in Section IV-C shows that the greedy algorithm using the same heuristic information is still far less efficient than ACO-MNCC. The advantage of the solution construction behavior in ACO-MNCC is thus confirmed.

Fig. 7 displays the convergence curves of ACO-MNCC, ACO-noLS, and ACO-noPhe. The convergence curves of ACO-noHeu are not drawn because its performance is so poor that the convergence curves are out of the ranges of the graphs. It can be observed from Fig. 7 that the convergence curves of ACO-MNCC generally rise above the other two, showing that ACO-MNCC can find satisfying solutions of C2, C4, C5, and C8 at a faster speed. Such a phenomenon reveals the advantage of ACO-MNCC in search efficiency. All the results in Table VII and Fig. 7 show that pheromone, heuristic information, and local search are necessary and important components in the proposed approach.

## V. Conclusion

With the objective of maximizing the network lifetime, this paper considers the problem of finding the maximum number of connected covers in a heterogeneous WSN. An ACO-based approach, termed ACO-MNCC, is proposed to solve the problem. The approach searches for the optimal solution by always pursuing one more connected cover than the best-so-far solution. This way, the approach not only avoids building excessive subsets but also improves the search efficiency by setting an explicit goal for the ants. Pheromone and heuristic information are also designed to accelerate the search process. A local search procedure is proposed to refine the best-so-far solution in the end of one iteration. Experimental results on thirty-three heterogeneous WSNs with different characteristics validate the effectiveness and efficiency of the approach, indicating that ACO-MNCC is a promising method for prolonging the lifetime of heterogeneous WSNs.

The present framework of ACO-MNCC can be dedicated to discrete point coverage by researching an appropriate objective function. It is expected that the implicit parallelism of the ACO framework can also be utilized for further reducing the computational time of ACO-MNCC when tackling large-scale WSNs.

## Appendix

**Proposition.** The objective function in (8) guarantees the objective value of a solution to increase with the number of connected covers if the positive weights $\omega_1$ and $\omega_2$ satisfy

$$\frac{\omega_2}{\omega_1} \geq \hat{C}. \tag{A.1}$$

**Proof.** First, we would like to point out a general fact that for any solution $\boldsymbol{S}=\{\boldsymbol{S}_1,\boldsymbol{S}_2,\ldots,\boldsymbol{S}_N\}$ with $C$ connected covers, its objective value satisfies

$$(\omega_1+\omega_2)C \leq \Phi(\mathbf{S}) < \omega_1 N + \omega_2 C. \tag{A.2}$$

Given two solutions $\boldsymbol{S}^{(1)}=\{\boldsymbol{S}_1^{(1)},\boldsymbol{S}_2^{(1)},\ldots,\boldsymbol{S}_{N^{(1)}}^{(1)}\}$ and $\boldsymbol{S}^{(2)}=\{\boldsymbol{S}_1^{(2)},\boldsymbol{S}_2^{(2)},\ldots,\boldsymbol{S}_{N^{(2)}}^{(2)}\}$. Suppose $\boldsymbol{S}^{(1)}$ has more connected covers than $\boldsymbol{S}^{(2)}$, i.e., $C^{(1)}>C^{(2)}\geq 0$. Then according to (A.2), the lower bound of the difference between their objective values can be estimated as

$$\begin{aligned}\Phi(\boldsymbol{S}^{(1)})-\Phi(\boldsymbol{S}^{(2)}) &> (\omega_1+\omega_2)C^{(1)}-(\omega_1 N^{(2)}+\omega_2 C^{(2)}) \\ &= \omega_2(C^{(1)}-C^{(2)})-\omega_1(N^{(2)}-C^{(1)})\end{aligned}. \tag{A.3}$$

Thus, a sufficient condition for ensuring $\Phi(\boldsymbol{S}^{(1)})>\Phi(\boldsymbol{S}^{(2)})$ is

$$\omega_2(C^{(1)}-C^{(2)})-\omega_1(N^{(2)}-C^{(1)}) \geq 0. \tag{A.4}$$

Both $\omega_1$ and $C^{(1)}-C^{(2)}$ are positive, thus

$$\frac{\omega_2}{\omega_1} \geq \frac{N^{(2)}-C^{(1)}}{C^{(1)}-C^{(2)}}. \tag{A.5}$$

As $N^{(2)}$ is clamped to the range of $[1,\hat{C}+1]$ by the ants' solution construction behavior, $C^{(1)}\geq 1$, and $C^{(1)}-C^{(2)}\geq 1$, the upper bound of the right part of (A.5) is $\hat{C}$. Therefore, $\Phi(\boldsymbol{S}^{(1)})>\Phi(\boldsymbol{S}^{(2)})$ can be guaranteed with

$$\frac{\omega_2}{\omega_1} \geq \hat{C}. \tag{A.6}$$

The proposition has been proven. □

## Acknowledgment

The authors would like to thank the editor-in-chief, associate editor, and reviewers for their valuable comments and suggestions that improved the paper's quality.

## Reference

[1] K. Römer, F. Mattern, and E. Zurich, "The design space of wireless sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 54-61, Dec. 2004.

[2] T. D. Räty, "Survey on contemporary remote surveillance systems for public safety," *IEEE Trans. Systems, Man, and Cybernetics, Part C*, vol. 40, no. 5, pp. 493-515, Sept. 2010.

[3] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor network: a survey," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 32-39, Dec. 2007.

[4] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Trans. Sensor Networks*, vol. 5, no. 1, p. Artical No. 5, Feb. 2009.

[5] C.-Y. Chang, J.-P. Sheu, Y.-C. Chen, and S.-W. Chang, "An obstacle-free and power-efficient deployment algorithm for wireless sensor networks," *IEEE Trans. Systems, Man, and Cybernetics, Part A*, vol. 39, no. 4, pp. 795-806, July 2009.

[6] F. Marcelloni and M. Vecchio, "A simple algorithm for data compression in wireless sensor networks," *IEEE Communications Letters*, vol. 12, no. 6, pp. 411-413, June 2008.

[7] S. Yang, H. Cheng, and F. Wang, "Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile Ad Hoc networks," *IEEE Trans. Systems, Man, and Cybernetics, Part C*, vol. 40, no. 1, pp. 52-63, Jan. 2010.

[8] S. Okdem and D. Karaboga, “Routing in wireless sensor networks using ant colony optimization,” in *First NASA/ESA Conference on Adaptive Hardware and Systems*, Istanbul, 2006, pp. 401-404.

[9] S. Okdem and D. Karaboga, “Routing in wireless sensor networks using ant colony optimization (ACO) router chip,” *Sensors*, vol. 9, no. 2, pp. 909-921, Feb. 2009.

[10] Y. P. Zhong, P. W. Huang, and B. Wang, “Maximum lifetime routing based on ant colony algorithm for wireless sensor networks,” in *IET Conference on Wireless, Mobile, and Sensor Networks*, Shanghai, China, 2007, pp. 789-792.

[11] Y.-F. Wen, Y.-Q. Chen, and M. Pan, “Adaptive ant-based routing in wireless sensor networks using Energy*Delay metrics,” *Journal of Zhejiang University Science A*, vol. 9, no. 4, pp. 531-538, Mar. 2008.

[12] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray, “The minimum power broadcast problem in wireless networks: An ant colony system approach,” in *IEEE CAS Workshop on Wireless Communications and Networking, California*, USA, 2002, pp. 5-6.

[13] F. Ge, Y. Wang, Q. Wang, and J. Wang, “Energy effcient broadcasting based on ant colony optimization in wireless sensor networks,” in *IEEE International Conference on Natural Computation*, Haiko, China, 2007, pp. 129-133.

[14] A. Acharya, A. Seetharam, A. Bhattacharyya, and M. K. Naskar, “Balancing energy dissipation in data gathering wireless sensor networks using ant colony optimization,” in *International Conference on Distributed Computing and Networking*, Hyderabad, India, 2009, pp. 437-443.

[15] H. Chen, C. K. Tse, and J. Feng, “Impact of topology on performance and energy efficiency in wireless sensor networks for source extraction,” *IEEE Trans. Parallel and Distributed Systems*, vol. 20, no. 6, pp. 886-897, June 2009.

[16] Y. Liang, J. Cao, L. Zhang, R. Wang, and Q. Pan, “A biologically inspired sensor wakeup control method for wireless sensor networks,” *IEEE Trans. Systems, Man, and Cybernetics, Part C*, vol. 40, no. 5, pp. 525-538, Sept. 2010.

[17] R. Tharmarasa, T. Kirubarajan, and M. L. Hernandez, “Large-scale optimal sensor array management for multi-target tracking,” *IEEE Trans. Systems, Man, and Cybernetics, Part C*, vol. 37, no. 5, pp. 803-814, Sept. 2007.

[18] R. Tharmarasa, T. Kirubarajan, P. Jiming, and T. Lang, “Optimization-based dynamic sensor management for distributed multitarget tracking,” *IEEE Trans. Systems, Man, and Cybernetics, Part C*, vol. 39, no. 5, pp. 534-546, Sept. 2009.

[19] T. R. Park, K.-J. Park, and M. J. Lee, “Design and analysis of asynchronous wakeup for wireless sensor networks,” *IEEE Trans. Wireless Communications*, vol. 8, no. 11, pp. 5530-5541, Nov. 2009.

[20] X.-M. Hu, J. Zhang, Y. Yu, H. S.-H. Chung, Y. L. Li, Y. H. Shi, and X. N. Luo, “Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks,” *IEEE Trans. Evolutionary Computation*, vol. 14, no. 5, pp. 766-781, Oct. 2010.

[21] Boukerche, X. Fei, and R. B. Araujo, “An optimal coverage-preserving scheme for wireless sensor networks based on local information exchange,” *Computer Communications*, vol. 30, no. 14-15, pp. 2708-2720, Oct. 2007.

[22] S. Funke, A. Kesselman, F. Kuhn, Z. Lotker, and M. Segal, “Improved approximation for connected sensor cover,” *Wireless Networks*, vol. 13, no. 2, pp. 153-164, Apr. 2007.

[23] A. Chamam and S. Pierre, “On the planning of wireless sensor networks: energy-efficient clustering under the joint routing and coverage constraint,” *IEEE Trans. Mobile Computing*, vol. 8, no. 8, pp. 1077-1086, Aug. 2009.

[24] M. Cardei and J. Wu, “Energy-efficient coverage problem in wireless ad-hoc network,” *Computer Communications*, vol. 29, no. 4, pp. 413-420, Feb. 2006.

[25] M. Cardei and D.-Z. Zhang, “Improving wireless sensor network lifetime through power aware organization,” *Wireless Networks*, vol. 11, no. 3, pp. 333-340, May 2005.

[26] H. Zhang and J. Hou, “Maintaining sensing coverage and connectivity in large sensor networks,” *Ad Hoc & Sensor Wireless Networks*, vol. 1, pp. 89-124, Mar. 2005.

[27] S. Slijepcevic and M. Potkonjak, “Power efficient organization of wireless sensor networks,” in *IEEE International Conference on Communication*, Helsinki, Finland, 2001, pp. 472-476.

[28] H. X. Bai, X. Chen, B. Li, and D. F. Han, “A location-free algorithm of energy-effcient connected coverage for high density wireless sensor networks ” *Discrete Event Dynamic Systems - Theory and Applications*, vol. 17, no. 1, pp. 1-21, Mar. 2007.

[29] C. F. Huang, Y. C. Tseng, and H. L. Wu, “Distributed protocols for ensuring both coverage and connectivity of a wireless sensor networks,” *ACM Trans. Sensor Networks*, vol. 3, no. 1, p. Article No. 5, Mar. 2007.

[30] Q. Zhao and M. Gurusamy, “Lifetime maximization for connected target coverage in wireless sensor networks,” *IEEE/ACM Trans. Networking*, vol. 16, no. 6, pp. 1378-1391, Dec. 2008.

[31] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh, “Exploiting heterogeneity in sensor networks,” in *IEEE International Conference on Computer Communications*, Miami, USA, 2005, pp. 878-890.

[32] E. J. Duarte-Melo and M. Liu, “Analysis of energy consumption and lifetime of heterogeneous sensor networks,” in *IEEE Global Telecommunications Conference*, St. Louis, USA, 2005, pp. 21-25.

[33] H. Wang, D. Estrin, and L. Girod, “Preprocessing in a tiered sensor network for habitat monitoring,” *EURASIP Journal on Applied Signal Processing*, vol. 2003, pp. 392-401, Jan. 2003.

[34] Y. M. Huang, M. Y. Hsieh, H. C. Chao, S. H. Hung, and J. H. Park, “Pervasive, secure access to a hierarchical sensor-based healthcare monitoring architectutre in wireless heterogeneous networks,” *IEEE Journal on Selected Area in Communications*, vol. 27, no. 4, pp. 400-411, May 2009.

[35] Q. Wang, K. Xu, G. Takahara, and H. Hassanein, “Device placement for heterogeneous wireless sensor networks: Minimum cost with lifetime constraint,” *IEEE Trans. Wireless Communications*, vol. 6, no. 7, pp. 2444-2453, July 2007.

[36] X. Du, M. Guizani, Y. Xiao, and H.-H. Chen, “Two-tier secure routing protocol for heterogeneous sensor networks,” *IEEE Trans. Wireless Communications*, vol. 6, no. 9, pp. 3395-3401, Sept. 2007.

[37] Y. Ma and J. H. Aylor, “System lifetime optimization for heterogeneous sensor networks with a hub-spoke technology,” *IEEE Trans. Mobile Computing*, vol. 3, no. 3, pp. 286-297, July - Aug. 2004.

[38] M. Dorigo, V. Maniezzo, and A. Colorni, “The ant system: Optimization by a colony of cooperating agents,” *IEEE Trans. Systems, Man, and Cybernetics, Part B*, vol. 26, no. 1, pp. 1-13, Feb. 1996.

[39] M. Dorigo and L. M. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem,” *IEEE Trans. Evolutionary Computation*, vol. 1, no. 1, pp. 53-66, Aug. 1997.

[40] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Massachusetts: MIT Press, 2004.

[41] A. C. Zecchin, H. R. Maier, A. R. Simpson, and J. B. Nixon, “Parametric study for an ant algorithm applied to water distribution system optimization,” *IEEE Trans. Evolutionary Computation*, vol. 9, no. 2, pp. 175-191, Apr. 2005.

[42] D. Martens, M. D. Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens, “Classification with ant colony optimization,” *IEEE Trans. Evolutionary Computation*, vol. 11, no. 5, pp. 651-665, Oct. 2007.

[43] W.-N. Chen and J. Zhang, “An ant colony optimization approach to a grid flow scheduling problem with various QoS requirements,” *IEEE Trans. Systems, Man, and Cybernetics, Part C*, vol. 39, no. 1, pp. 29-43, Jan. 2009.

[44] X.-M. Hu, J. Zhang, H. S.-H. Chung, O. Liu, and J. Xiao, “An intelligent testing system embedded with an ant-colony-optimization-based test composition method,” *IEEE Trans. Systems, Man, and Cybernetics, Part C*, vol. 39, no. 6, pp. 659-669, Nov. 2009.

[45] J. Zhang, H. S.-H. Chung, A. W. Lo, and T. Huang, “Extended ant colony optimization algorithm for power electronic circuit design,” *IEEE Trans. Power Electronics*, vol. 24, no. 1, pp. 147-162, Jan. 2009.

[46] W.-N. Chen, J. Zhang, H. S.-H. Chung, R.-Z. Huang, and O. Liu, “Optimizaing discounted cash flow in project scheduling: An ant colony optimization approach,” *IEEE Trans. Systems, Man, and Cybernetics, Part C*, vol. 40, no. 1, pp. 64-77, Jan. 2010.

[47] Z.-H. Zhan, J. Zhang, Y. Li, O. Liu, S. K. Kwok, W. H. Ip, and O. Kaynak, “An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem,” *IEEE Trans. Intelligent Transportation Systems*, vol. 11, no. 2, pp. 399-412, June 2010.

[48] H. Zhang and J. C. Hou, “Maximizing α-lifetime for wireless sensor networks,” *International Journal of Sensor Networks*, vol. 1, no. 1-2, pp. 70-77, Jan. 2006.

[49] D. Zorbas, D. Glynos, P. Kotzanikolaou, and C. Douligeris, "Solving coverage problems in wireless sensor networks using cover sets," *Ad Hoc Networks*, vol. 8, no. 4, pp. 400-415, June 2010.

[50] G. Chartrand and P. Zhang, *Introduction to Graph Theory*. Peking: The People's Post and Telecommunication Press, 2006.

[51] Z. Michalewicz, Genetic Algorithms + Data Structure = Evolutionary Programs. London: Springer-Verlag, 1996.

**Figure Captions**

Fig. 1. Illustration of a connected cover that satisfies the three constraints of coverage, collection, and routing. Dots and triangles represent sensors and sinks respectively. Any two devices connected by a dash can directly communicate with each other.

Fig. 2. Illustration of fields in the target area. Each dot represents a sensor and the circle centered at the dot indicates the sensing range of the sensor. Each $F_i$ ($i$=1,2,...,15) denotes a field in the target area. A field in darker shade is covered by more sensors.

Fig. 3. An example of the construction graph with $|\boldsymbol{SE}|$=5, $|\boldsymbol{SI}|$=3, and $N_t$=5.

Fig. 4. Pseudocode of the local search procedure in ACO-MNCC.

Fig. 5. Flowcharts of the proposed ACO-MNCC. (a) The overall flowchart. (b) The detailed flowchart of an ant's solution construction procedure.

Fig. 6. Pseudocode of the greedy algorithm to compare with the ACO-MNCC.

Fig. 7. Convergence curves of ACO-MNCC, ACO-noLS, and ACO-noPhe on (a) C2, (b) C4, (c) C5, and (d) C8.

**Table Captions**

**Figures**

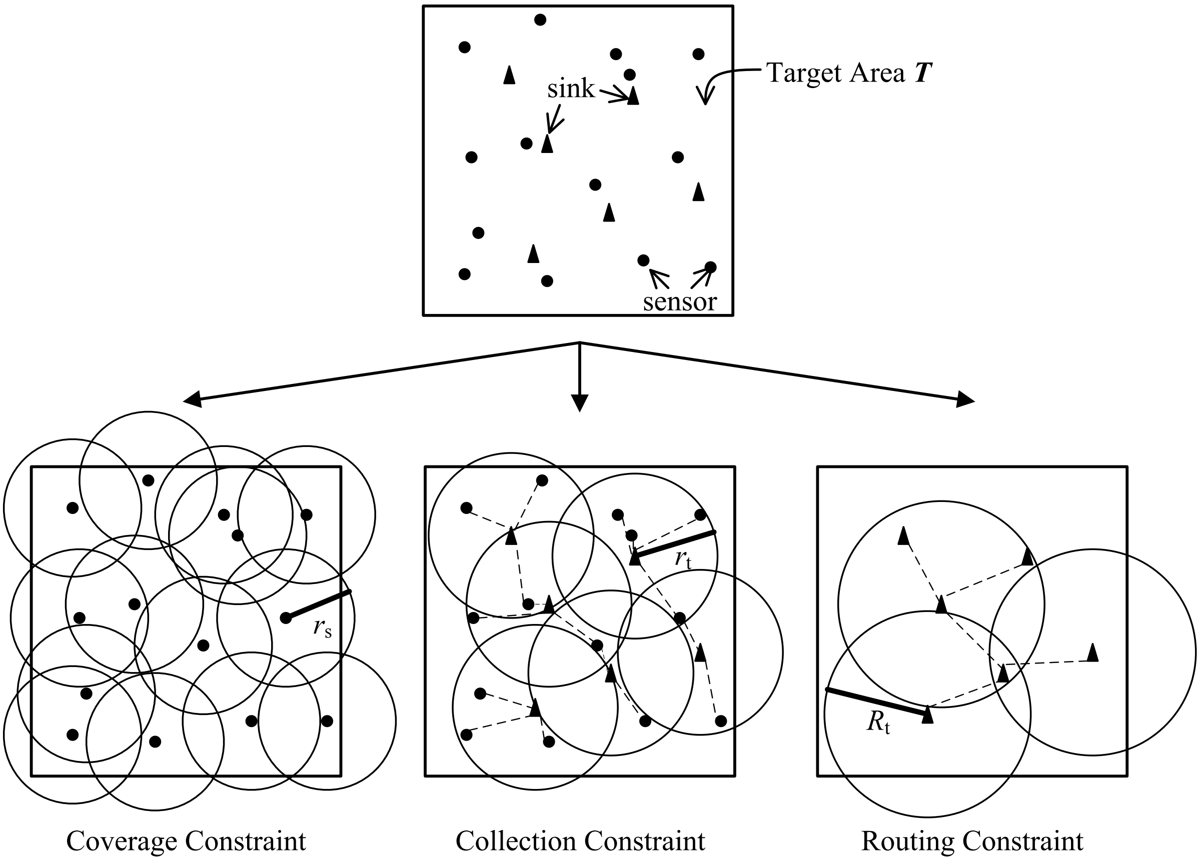


Fig. 1. Illustration of a connected cover that satisfies the three constraints of coverage, collection, and routing. Dots and triangles represent sensors and sinks respectively. Any two devices connected by a dash can directly communicate with each other.

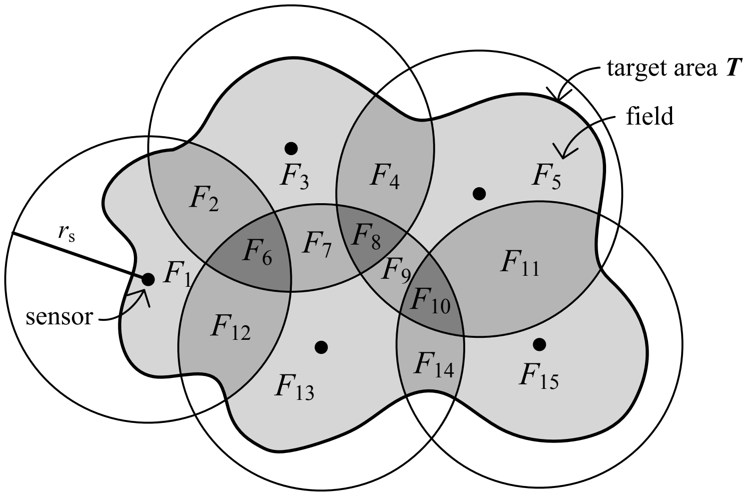


Fig. 2. Illustration of fields in the target area. Each dot represents a sensor and the circle centered at the dot indicates the sensing range of the sensor. Each $F_i$ ($i$=1,2,...,15) denotes a field in the target area. A field in darker shade is covered by more sensors.

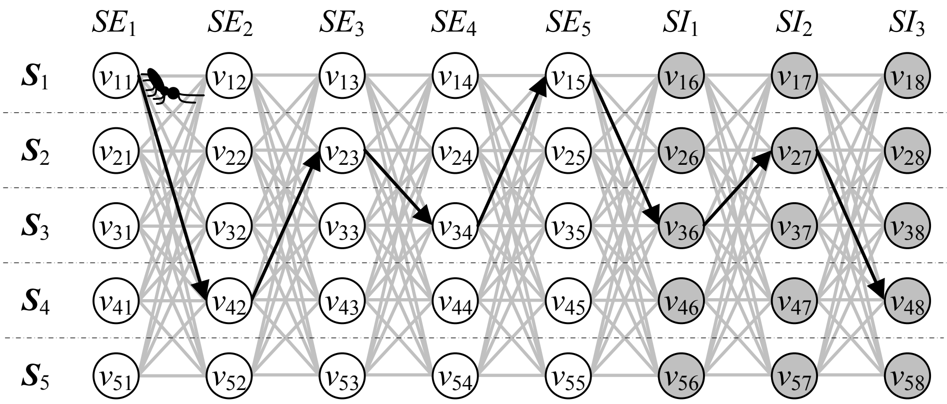


Fig. 3. An example of the construction graph with $|\boldsymbol{SE}|$=5, $|\boldsymbol{SI}|$=3, and $N_t$=5.

**Procedure LOCAL_SEARCH**
**Begin**
$\boldsymbol{S}' \leftarrow \boldsymbol{S}^{\mathrm{bs}}$;
**For** every critical field $F_v \in \boldsymbol{F}$
  **For** $i$=1 **to** $N^{\mathrm{bs}}$
    $\boldsymbol{C}_i \leftarrow \varnothing$;
    **For** every sensor $SE_j \in \boldsymbol{S}_i^{\mathrm{bs}}$
      **If** $SE_j$ covers $F_v$ **Then** $\boldsymbol{C}_i \leftarrow \boldsymbol{C}_i \cup SE_j$;
    **End For**
  **End For**
  **For** $i$=1 **to** $N^{\mathrm{bs}}$
    **If** $\boldsymbol{C}_i = \varnothing$ **Then**
      **For** $k$=1 **to** $N^{\mathrm{bs}}$
        **If** $|\boldsymbol{C}_k| \geq 2$ **Then**
          **For** every $SE_j \in \boldsymbol{C}_k$
            **If** $SE_j$ is redundant **Then**
              $\boldsymbol{S}_k^{\mathrm{bs}} \leftarrow \boldsymbol{S}_k^{\mathrm{bs}} - SE_j$;
              $\boldsymbol{C}_k \leftarrow \boldsymbol{C}_k - SE_j$;
              $\boldsymbol{S}_i^{\mathrm{bs}} \leftarrow \boldsymbol{S}_i^{\mathrm{bs}} \cup SE_j$;
              $\boldsymbol{C}_i \leftarrow \boldsymbol{C}_i \cup SE_j$
              **Break**;
          **End For**
        **If** $|\boldsymbol{C}_i|>0$ **Then Break**;
      **End For**
  **End For**
**End For**
**For** $i$=1 **to** $N^{\mathrm{bs}}$
  **If** $\kappa_i=\chi_i=\lambda_i=1$ **Then**
    **For** every sink $SI_j \in \boldsymbol{S}_i^{\mathrm{bs}}$
      **If** $SI_j$ is redundant **Then**
        $\boldsymbol{S}_i^{\mathrm{bs}} \leftarrow \boldsymbol{S}_i^{\mathrm{bs}} - SI_j$;
        **If** $C^{\mathrm{bs}} < N^{\mathrm{bs}}$ **Then**
          Randomly select a subset $k$ with at least one constraint unsatisfied;
        **Else** $k \leftarrow N^{\mathrm{bs}}+1$;
        $\boldsymbol{S}_k^{\mathrm{bs}} \leftarrow \boldsymbol{S}_k^{\mathrm{bs}} \cup SI_j$;
    **End For**
**End For**
**If** $\Phi(\boldsymbol{S}^{\mathrm{bs}})<\Phi(\boldsymbol{S}')$ **Then** $\boldsymbol{S}^{\mathrm{bs}} \leftarrow \boldsymbol{S}'$;
**End**

Fig. 4. Pseudocode of the local search procedure in ACO-MNCC.

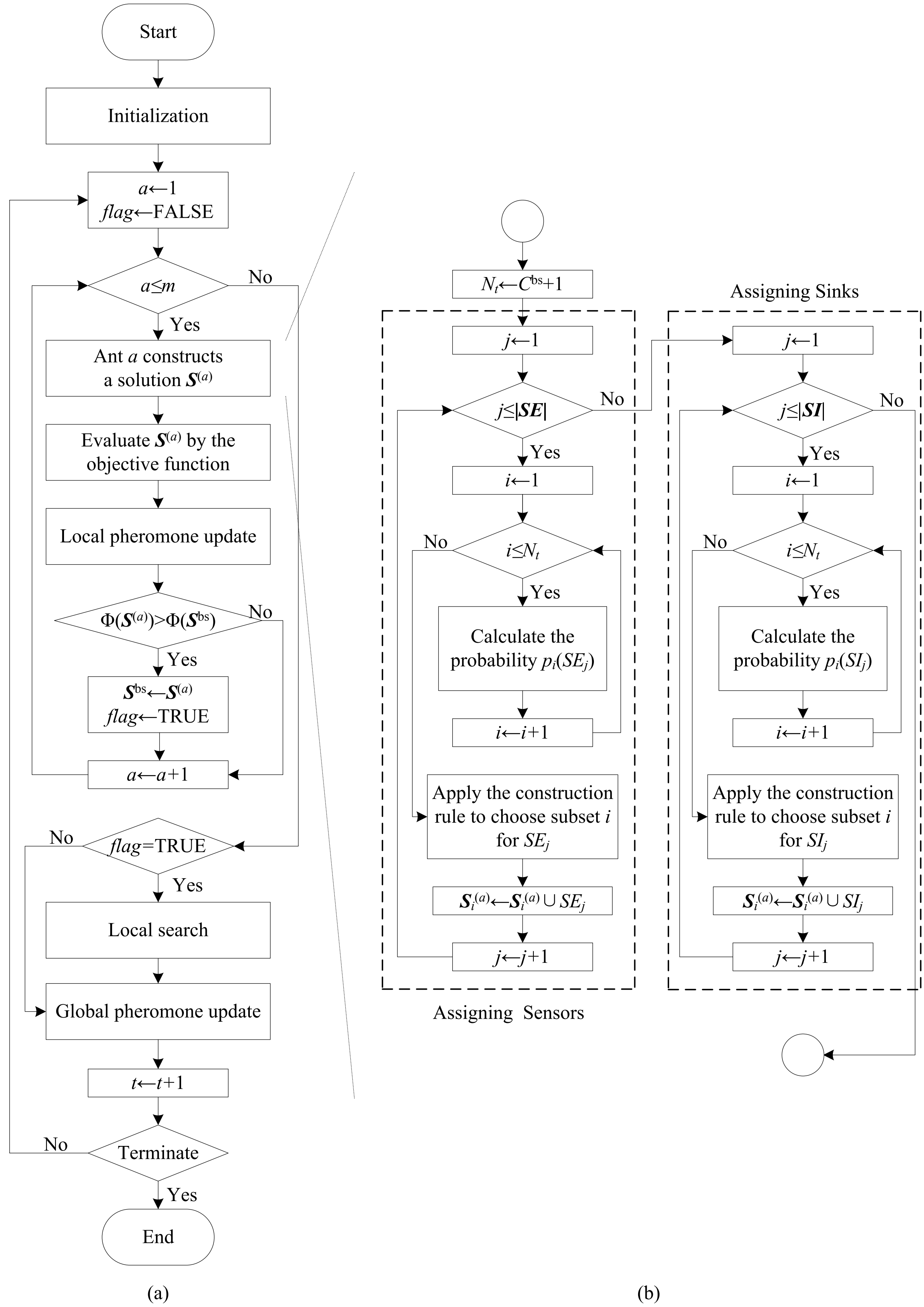


Fig. 5. Flowcharts of the proposed ACO-MNCC. (a) The overall flowchart. (b) The detailed flowchart of an ant's solution construction procedure.

**Procedure GREEDY**

**Begin**

**For** $i$=1 **to** $\hat{C}+1$

  $\boldsymbol{S}_i \leftarrow \varnothing$;

**End For**

Shuffle the set of sensors $\boldsymbol{SE}$;

Shuffle the set of sinks $\boldsymbol{SI}$;

**For** every device $J$ in $\boldsymbol{SE} \cup \boldsymbol{SI}$

  $flag \leftarrow$ FALSE;

  **For** $i$=1 **to** $\hat{C}+1$

    Calculate the heuristic information $\eta_i(J)$ according to (11);

    **If** $\eta_i(J)>0$ **Then** $flag \leftarrow$ TRUE;

  **End For**

  **If** $flag$=TRUE **Then** $i \leftarrow \arg\max_{1\le k\le \hat{C}+1} \eta_k(J)$;

  **Else** $i \leftarrow rand(1, \hat{C}+1)$;

  $\boldsymbol{S}_i \leftarrow \boldsymbol{S}_i \cup J$;

**End For**

Evaluate the solution $\boldsymbol{S}=\{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \mathbf{S}_{\hat{C}+1}\}$

**End**

Fig. 6. Pseudocode of the greedy algorithm to compare with the ACO-MNCC.

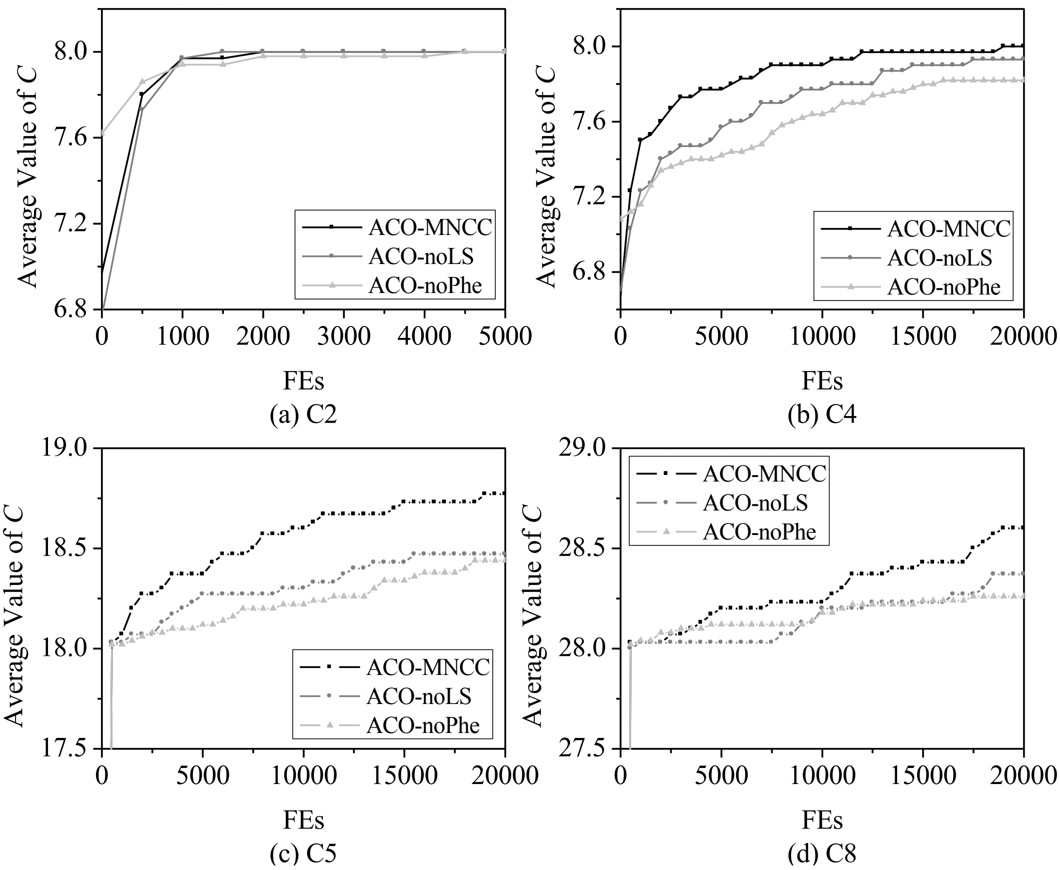


Fig. 7. Convergence curves of ACO-MNCC, ACO-noLS, and ACO-noPhe on (a) C2, (b) C4, (c) C5, and (d) C8.

## Tables

Table I Notations Used in Section III

| Symbol | Description |
|---|---|
| $\lvert\cdot\rvert$ | Operator returning the size of a set |
| $\lVert x-y \rVert$ | Operator returning the distance between two points $x$ and $y$ |
| $rand(a, b)$ | Function returning a random integer in $[a,b]$ |
| $\Phi(\cdot)$ | Objective function |
| $\omega_1$, $\omega_2$ | Weights used in the objective function |
| $L$, $W$ | Length and width of deploy area |
| $\boldsymbol{T}$ | Target area |
| $\boldsymbol{SE}=\{SE_1,SE_2,\ldots,SE_{\lvert\boldsymbol{SE}\rvert}\}$ | Set of sensors |
| $\boldsymbol{SI}=\{SI_1,SI_2,\ldots,SI_{\lvert\boldsymbol{SI}\rvert}\}$ | Set of sinks |
| $r_s$, $r_t$ | Sensing and transmission ranges of sensors |
| $R_t$ | Transmission range of sinks, $R_t>r_t$ |
| $\boldsymbol{F}=\{F_1,F_2,\ldots,F_{\lvert\boldsymbol{F}\rvert}\}$ | Set of fields in the target area |
| $\boldsymbol{F}_i$ | Set of fields covered by $SE_i$, $\boldsymbol{F}_i\subseteq\boldsymbol{F}$, $i=1,2,\ldots,\lvert\boldsymbol{SE}\rvert$ |
| $\hat{C}$, $C_{\max}$ | Upper bound and actual value of the maximum number of connected covers |
| $m$ | Number of ants in ACO-MNCC |
| $q_0$ | Parameter in the solution construction rule of ACO-MNCC, $q_0\in[0,1]$ |
| $q$ | Uniform random number, $q\in[0,1]$ |
| $\beta$ | Parameter controlling influence of heuristic information |
| $\rho$ | Evaporation rate in local pheromone update |
| $\xi$ | Evaporation rate in global pheromone update |
| $\tau_0$ | Initial pheromone value |
| $\Delta\tau$ | Pheromone increment in global pheromone update |
| $\boldsymbol{S}$, $\boldsymbol{S}^{(a)}$, $\boldsymbol{S}^{\mathrm{bs}}$ | Solution, solution built by ant $a$ ($a=1,2,\ldots,m$), best-so-far solution |
| $N$, $N^{(a)}$, $N^{\mathrm{bs}}$ | Number of subsets in $\boldsymbol{S}$, $\boldsymbol{S}^{(a)}$, $\boldsymbol{S}^{\mathrm{bs}}$ |
| $C$, $C^{(a)}$, $C^{\mathrm{bs}}$ | Number of connected covers in $\boldsymbol{S}$, $\boldsymbol{S}^{(a)}$, $\boldsymbol{S}^{\mathrm{bs}}$ |
| $\boldsymbol{S}_i$, $\boldsymbol{S}_i^{(a)}$, $\boldsymbol{S}_i^{\mathrm{bs}}$ | Subset $i$ in $\boldsymbol{S}$, $\boldsymbol{S}^{(a)}$, $\boldsymbol{S}^{\mathrm{bs}}(t)$, $i=1,2,\ldots,N/N^{(a)}/N^{\mathrm{bs}}$ |
| $U_i$, $V_i$ | Numbers of sensors and sinks in $\boldsymbol{S}_i$, $i=1,2,\ldots,N$ |
| $G_i$ | Communication graph derived from sinks in $\boldsymbol{S}_i$, $i=1,2,\ldots,N$ |
| $B_i$ | Size of the largest connected subgraph in $G_i$, $i=1,2,\ldots,N$ |
| $H_i$ | Number of collected sensors in $\boldsymbol{S}_i$, $i=1,2,\ldots,N$ |
| $\kappa_i$, $\chi_i$, $\lambda_i$ | Criterion for the coverage, collection, and routing constraints on $\boldsymbol{S}_i$, $i=1,2,\ldots,N$ |
| $N_t$ | Number of available subsets in iteration $t$, $t=1,2,\ldots$ |
| $v_{ij}$ | Vertex on row $i$ and column $j$ of the construction graph, $i=1,2,\ldots,N_t$, $j=1,2,\ldots,\lvert\boldsymbol{SE}\rvert+\lvert\boldsymbol{SI}\rvert$, |
| $\tau(J, K)$ | Pheromone between devices $J$ and $K$, $J$, $K\in\boldsymbol{SE}\cup\boldsymbol{SI}$ |

| Symbol | Description |
|---|---|
| $\eta_i(J)$ | Heuristic value for assigning an unassigned device $J$ to $\boldsymbol{S}_i$, $i$=1,2,…,$N_t$, $J \in \boldsymbol{SE} \cup \boldsymbol{SI}$ |
| $\mathrm{T}_i(J)$ | Average pheromone between an unassigned device $J$ and the existing devices in $\boldsymbol{S}_i$, $i$=1,2,…,$N_t$, $J \in \boldsymbol{SE} \cup \boldsymbol{SI}$ |
| $p_i(J)$ | Probability for assigning an unassigned device $J$ to $\boldsymbol{S}_i$, $i$=1,2,…,$N_t$, $J \in \boldsymbol{SE} \cup \boldsymbol{SI}$ |

Table II Test Cases

| Case No. | $\lvert\boldsymbol{SE}\rvert$ | $\lvert\boldsymbol{SI}\rvert$ | Case No. | $\lvert\boldsymbol{SE}\rvert$ | $\lvert\boldsymbol{SI}\rvert$ | Case No. | $\lvert\boldsymbol{SE}\rvert$ | $\lvert\boldsymbol{SI}\rvert$ | $r_s$ | $r_t$ | $R_t$ | $\hat{C}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A1 | 200 | 100 | B1 | 179 | 76 | C1 | 161 | 48 | 10 | 18 | 36 | 6 |
| A2 | 400 | 100 | B2 | 295 | 69 | C2 | 229 | 54 | 10 | 20 | 40 | 8 |
| A3 | 400 | 200 | B3 | 328 | 154 | C3 | 282 | 103 | 15 | 20 | 40 | 21 |
| A4 | 600 | 100 | B4 | 444 | 75 | C4 | 347 | 57 | 8 | 20 | 40 | 8 |
| A5 | 600 | 200 | B5 | 496 | 156 | C5 | 434 | 116 | 11 | 18 | 36 | 19 |
| A6 | 800 | 100 | B6 | 464 | 60 | C6 | 269 | 49 | 8 | 15 | 30 | 5 |
| A7 | 800 | 200 | B7 | 586 | 137 | C7 | 450 | 108 | 10 | 18 | 36 | 16 |
| A8 | 800 | 400 | B8 | 639 | 268 | C8 | 553 | 191 | 12 | 18 | 36 | 29 |
| A9 | 1000 | 100 | B9 | 773 | 71 | C9 | 624 | 54 | 5 | 18 | 36 | 6 |
| A10 | 1000 | 200 | B10 | 848 | 147 | C10 | 732 | 112 | 6 | 15 | 30 | 11 |
| A11 | 1000 | 400 | B11 | 883 | 301 | C11 | 763 | 224 | 9 | 16 | 32 | 25 |

Table III Comparison of ACO-MNCC and the Greedy Algorithm on Set A

| Case No. | ACO-MNCC | | | | Greedy Algorithm | | | |
|---|---|---|---|---|---|---|---|---|
| | Best | Worst | Avg | SR (%) | Best | Worst | Avg | SR (%) |
| A1 | 6 | 6 | 6 | 100 | 4 | 0 | 1 | 0 |
| A2 | 8 | 8 | 8 | 100 | 8 | 7 | 7.87 | 87 |
| A3 | 21 | 21 | 21 | 100 | 21 | 19 | 20.87 | 90 |
| A4 | 8 | 8 | 8 | 100 | 8 | 6 | 7.27 | 43 |
| A5 | 19 | 19 | 19 | 100 | 19 | 14 | 17.27 | 13 |
| A6 | 5 | 5 | 5 | 100 | 5 | 4 | 4.9 | 90 |
| A7 | 16 | 16 | 16 | 100 | 16 | 15 | 15.87 | 87 |
| A8 | 29 | 29 | 29 | 100 | 29 | 28 | 28.87 | 87 |
| A9 | 6 | 6 | 6 | 100 | 5 | 0 | 2.57 | 0 |
| A10 | 11 | 11 | 11 | 100 | 6 | 1 | 3.5 | 0 |
| A11 | 25 | 25 | 25 | 100 | 24 | 17 | 20.57 | 0 |

Table IV Comparison of ACO-MNCC and the Greedy Algorithm on Set B

| Case No. | ACO-MNCC | | | | Greedy Algorithm | | | |
|---|---|---|---|---|---|---|---|---|
| | Best | Worst | Avg | SR (%) | Best | Worst | Avg | SR (%) |
| B1 | **6** | **6** | **6** | **100** | 1 | 0 | 0.07 | 0 |
| B2 | **8** | **8** | **8** | **100** | 7 | 5 | 6.03 | 0 |
| B3 | **21** | **21** | **21** | **100** | 21 | 18 | 20 | 37 |
| B4 | **8** | **8** | **8** | **100** | 6 | 1 | 3.1 | 0 |
| B5 | **19** | **19** | **19** | **100** | 18 | 11 | 14.83 | 0 |
| B6 | **5** | **5** | **5** | **100** | 4 | 1 | 2.83 | 0 |
| B7 | **16** | **16** | **16** | **100** | 16 | 13 | 15.07 | 37 |
| B8 | **29** | **29** | **29** | **100** | 29 | 23 | 26.93 | 10 |
| B9 | **6** | **6** | **6** | **100** | 1 | 0 | 0.2 | 0 |
| B10 | **11** | **10** | **10.03** | **3** | 3 | 0 | 0.6 | 0 |
| B11 | **25** | **25** | **25** | **100** | 20 | 10 | 16.2 | 0 |

Table V Comparison of ACO-MNCC and the Greedy Algorithm on Set C

| Case No. | ACO-MNCC | | | | Greedy Algorithm | | | |
|---|---|---|---|---|---|---|---|---|
| | Best | Worst | Avg | SR (%) | Best | Worst | Avg | SR (%) |
| C1 | **6** | **5** | **5.1** | **10** | 0 | 0 | 0 | 0 |
| C2 | **8** | **8** | **8** | **100** | 3 | 0 | 0.93 | 0 |
| C3 | **21** | **21** | **21** | **100** | 19 | 8 | 14.23 | 0 |
| C4 | **8** | **8** | **8** | **100** | 2 | 0 | 0.33 | 0 |
| C5 | **19** | **18** | **18.80** | **80** | 11 | 4 | 7 | 0 |
| C6 | **5** | **5** | **5** | **100** | 1 | 0 | 0.27 | 0 |
| C7 | **16** | **16** | **16** | **100** | 11 | 2 | 7.33 | 0 |
| C8 | **29** | **28** | **28.63** | **63** | 22 | 13 | 17.2 | 0 |
| C9 | **5** | **5** | **5** | 0 | 0 | 0 | 0 | 0 |
| C10 | **10** | **9** | **9.33** | 0 | 0 | 0 | 0 | 0 |
| C11 | **23** | **23** | **23** | 0 | 10 | 1 | 3.5 | 0 |

Table VI Average CPU Time of ACO-MNCC

| Case No. | Time (sec.) | Case No. | Time (sec.) | Case No. | Time (sec.) |
|---|---|---|---|---|---|
| A1 | 0.06 | B1 | 2.01 | C1 | 0.95 |
| A2 | 0.03 | B2 | 0.03 | C2 | 1.01 |
| A3 | 0.13 | B3 | 0.12 | C3 | 0.42 |
| A4 | 0.07 | B4 | 0.06 | C4 | 14.36 |
| A5 | 0.26 | B5 | 0.41 | C5 | 54.59 |
| A6 | 0.07 | B6 | 0.03 | C6 | 0.05 |
| A7 | 0.25 | B7 | 0.18 | C7 | 2.80 |
| A8 | 0.59 | B8 | 0.57 | C8 | 164.59 |
| A9 | 0.18 | B9 | 33.85 | C9 | 0.23 |
| A10 | 10.41 | B10 | 1.00 | C10 | 47.40 |
| A11 | 1.78 | B11 | 26.37 | C11 | 5.11 |

Table VII Comparison of ACO-MNCC and Its Three Variants

| Case No. | ACO-noHeu | ACO-noPhe | ACO-noLS | ACO-MNCC |
|---|---|---|---|---|
| C2 | 2.6 | **8** | **8** | **8** |
| C4 | 2.1 | 7.94 | 7.93 | **8** |
| C5 | 3.68 | 18.44 | 18.37 | **18.77** |
| C8 | 5.8 | 28.28 | 28.37 | **28.63** |