Ant Colony Optimization Algorithm for Lifetime Maximization in Wireless Sensor Network with Mobile Sink

Jing-hui Zhong and Jun Zhang (Corresponding Author)
Department of Computer Science, Sun Yat-sen University, P.R. China
Key Laboratory of Digital Life, Ministry of Education, P.R. China
Key Laboratory of Software Technology, Education Dept. of Guangdong Province, P.R. China
junzhang@ieee.org

ABSTRACT
In wireless sensor networks (WSNs), sensors near the sink can be burdened with a large amount of traffic, because they have to transmit data generated by themselves and those far away from the sink. Hence the sensors near the sink would deplete their energy much faster than the others, which results in a short network lifetime. Using mobile sink is an effective way to tackle this issue. This paper explores the problem of determining the optimal movements of the mobile sink to maximize the network lifetime. A novel ant colony optimization algorithm (ACO), namely the ACO-MSS, is developed to solve the problem. The proposed ACO-MSS takes advantage of the global search ability of ACO and adopts effective heuristic information to find a near globally optimal solution. Multiple practical factors such as the forbidden regions and the maximum moving distance of the sink are taken into account to facilitate the real applications. The proposed ACO-MSS is validated by a series of simulations on WSNs with different characteristics. The simulation results demonstrate the effectiveness of the proposed algorithms.

Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]:Problem Solving, Control Methods and search-Heuristic methods; G.1.6 [Numerical Analysis]: Optimization-Global optimization

General Terms
Algorithms, Design, Experimentation.

Keywords
Ant colony optimization, Lifetime maximization, Mobile sink scheduling, Wireless sensor network

1. INTRODUCTION
Wireless sensor networks (WSNs) are comprised of a larger number of small and inexpensive sensor nodes (SNs), which are deployed for collecting information from environment and communicating with each other via wireless signal. Applications of WSNs include traffic monitoring, environment monitoring and home education [1]-[3], etc. In WSNs, the SNs are usually battery-powered and are often deployed in large quantities, sometimes in not easily accessible environment. It is generally impossible to recharge SNs. Therefore, conserving the energy of SNs, so as to prolong the network lifetime becomes one of the most important issues in the research field of WSNs [3]-[7].

In WSNs, SNs near the sink tend to consume much more energy, because they have to transmit packets generated by themselves and those far away from the sink. Hence they would deplete their energy much faster than the others, which results in a short network lifetime. This is the “energy-hole” problem in WSNs. One effective way to tackle this issue is to use mobile sink [8]-[15]. The key idea is to allow the sink to move though the network and collect data from SNs at different locations (i.e., sink sites), so that the energy consumption of the network can be more balanced.

Determining the movements of the mobile sink to maximize the lifetime of a WSN, or the mobile sink scheduling (MSS) problem, has drawn increasing attentions over the past few years. In [8], Shah et al. introduced the basic idea of using mobile nodes to conserve the energy of SNs. In their network architecture, the mobile nodes, termed mules, move arbitrarily thought the network and collect data from nearby SNs via single-hop communication. However, this approach may suffer from long latency and cannot balance the energy consumption among SNs effectively. In [10], Bi et al suggested moving the sink in rounds. At the end of each round, the sink makes a constant movement towards the node that has the most residual energy. Works on this line can be found in [11][12][17]. Nevertheless, these approaches can only obtain a local optimal solution, because they only use local information to search solutions. In [15], wang et al. tried to optimize the entire moving path of the sink before the sink starts working, with the global information of the network, such as the positions and residual energy of all SN known in advanced. They presented a linear programming (LP) optimization model to solve the MSS problem. But the sink sites and the flow routing are fixed. Paradimitriou et al. [16]showed that taking into account the flow routing to solve the MSS problem can lead to better network lifetime. They formulated an LP optimization model to solve the joint optimization problem. Similar work that uses LP methods to solve the MSS problem can be found in [13] [14][17]. The LP based methods are able to obtain the globally optimal solution, but they are only suitable for small-scale WSNs because of their large memory requirement.

Instead of using LP methods, this paper proposes an ant colony optimization (ACO) algorithm, namely, ACO-MSS, to solve the MSS problem. ACO is a new paradigm of swarm intelligence algorithm which is proposed by Dorigo in the early 1990s [19] [20]. It simulates the foraging behaviors of the ant species. So far, ACO has been shown particularly good at solving problems of graph or topology types, such as the travelling sale man problem [20], cash flows scheduling [24], aircraft arrival scheduling [23], and etc [24]-[28]. Because the moving behavior of the sink in a WSN can be transformed into a graph type combinational problem, the search mechanism of ACO is applicable to the MSS problem. To the best of our knowledge, this paper is the first attempt to apply ACO to the MSS problem.

In ACO-MSS, the monitoring region is divided into discrete grids. Those grids with their center points out of the forbidden regions are regarded as feasible grids. By regarding the feasible grids as candidate sink sites, the MSS problem can be converted into a combinational optimization problem, which is very suitable for the ACO to solve. In order to improve the algorithm efficiency, new hybrid heuristic information is defined and utilized. The proposed ACO-MSS would be validated by performing simulations on networks with various characteristics. Two mobile sink algorithms, i.e., the GMRE approach [18] and the LP based approach [14], are used for comparison.

The rest of the paper is organized as follows. Section 2 defines the MSS problem. Section 3 illustrates the implementation details of the ACO-MSS. Simulation studies are presented in Section 4, followed by conclusions in Section 5.

2. NETWORK MODEL AND PROBLEM DEFINITION

In this section, the network model is presented firstly, followed by the definition of the MSS problem. The notations for problem definition are tabulated in Table 1.

Table 1. Notations for problem definition

Symbol	Definition
n	Number of SNs
m	Number of sink sites
r_i	The sensing data rate of z_i
R_v	The maximum step distance of the mobile sink
R	The maximum transmission range of SNs
$N(z_i)$	The set of nodes which can communicate with z_i
$u_{i,j}^k$	The data rate from z_i to z_j when the sink is at s_k
π_i	The sojourn time of the sink at s_i
T	The network lifetime
E_i	The initial energy of z_i
t_i	The lifetime of z_i
e^{rec}	The energy consumption of receiving one bit of data
e^{tran}	The energy consumption of transmitting one bit of data
$\omega_{i,j}$	The energy consumption of transmitting one bit of data from z_i to z_j
$\{s_1,\ldots, s_m\}$	The set of sink sites
$\{z_0,z_1,\ldots, z_n\}$	The set of nodes, where z_0 is the sink and $z_1\ldots z_n$ are the SNs

2.1 Network Model

We consider a WSN containing a sink and n battery-powered sensor nodes (SNs). SNs are assumed able to adjust their transmission range with an upper bound of R, so as to conserve energy. Each SN generates sensing data with a rate of r_i. The sensing date is forwarded to the sink via single-hop or multi-hop manner. In order to alleviate the traffic burden of SNs near the sink, the sink is allowed to move though the monitoring region. The sink collects data when it stops and it does not collect data during moving.

In this paper, the energy consumption model in [21] is used to compute the network lifetime. Specifically, the energy consumption of transmitting one bit of data over a distance d is

$$e^{\text{tran}} = a + b \cdot d^v \tag{1}$$

where a and b are constants related to the transmission media properties, $v \in [2,4]$ is the path loss index; the energy consumption for each SN receiving u bits of data is

$$e^{\text{rec}} = \delta \tag{2}$$

where ρ is a constant. Denote the set of all nodes in the network as

$$\{z_0, z_1, z_2, ..., z_n\} \tag{3}$$

where z_0 represents the sink, $z_1,\ldots,z_n$ represent the SNs. The lifetime of each SN is computed by

$$t_i = \frac{E_i}{e_i^{\text{rec}} + e_i^{\text{tran}}} \tag{4}$$

where E_i is the initial energy of z_i, e_i^{rec}, and e_i^{tran} are the energy consumption rates of z_i for receiving data and transmitting data respectively. We define the network lifetime T to be the duration of the network when it starts working until the first node dies due to energy exhaustion, i.e.,

$$T = \min\{t_i \mid i = 1, 2, ..., n\} \tag{5}$$

2.2 MSS Problem Definition

The MSS problem addressed in this paper aims to schedule the movements of the sink and the flow routing, so that the network lifetime can be maximized. To achieve this goal, three issues need to be addressed: 1) the positions and visiting order of sink sites, 2) the sojourn time of the sink at each sink site, 3) the flow routing associated with each sink site. Moreover, to facilitate practical applications, we consider that the sink cannot collect data at some forbidden regions and the step distance of the sink should be smaller than or equal to R_v.

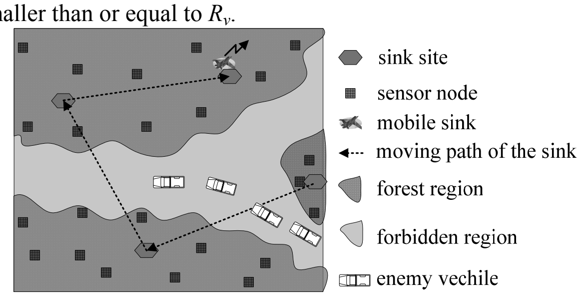

Figure 1. An application example of the proposed algorithm.

Fig .1 shows one application example of the MSS problem. In this instance, a large number of SNs are deployed in a battle field to monitor the movements of enemy vehicles or troops. A mobile sink flies over the monitoring region to collect data from SNs at several sink sites. The sensing data collected by the sink will be sent to the data management center immediately. The mobile sink does not collect data when flying. Hence, the step distance of the mobile sink should be small enough to avoid long latency. Due to some physical constraints or security considerations, the mobile sink is not allowed to collect data at some forbidden regions.

Formally, the MSS problem can be formulated as follows.

$$\text{Maximize } \sum_{i=1}^{m} \pi_i \tag{6}$$

subject to

$$\sum_{i\in N(s_k)} u_{i,0}^k = \sum_{i=1}^{n} r_i,\ \forall k \in [1,m] \tag{7}$$

$$\sum_{j\in N(z_i)} u_{i,j}^k - r_i - \sum_{j\in N(z_i)} u_{j,i}^k = 0,\ \forall i \in [1,n], \forall k \in [1,m] \tag{8}$$

$$\sum_{k=1}^{m} ((\sum_{j\in N(z_i)} (\omega_{i,j} \cdot u_{i,j}^k) + \sum_{j\in N(z_i)} (\delta \cdot u_{j,i}^k)) \cdot \pi_k) \le E_i,\ \forall i \in [1,n] \tag{9}$$

$$\left|\overline{s_i s_{i+1}}\right| \le R_v,\ \forall i \in [1, m-1] \tag{10}$$

$$s_i \in \boldsymbol{F},\ \forall i \in [1,m] \tag{11}$$

$$m \ge 0 \tag{12}$$

$$\pi_i \ge 0,\ \forall i \in [1,m] \tag{13}$$

$$u_{i,j}^k \ge 0,\ \forall i \in [1,n], \forall j \in [1,n], \forall k \in [1,m] \tag{14}$$

where π_i is the sojourn time of the sink at s_i, $\left|\overline{s_i s_j}\right|$ is the Euclidean distance between s_i and s_j, $\omega_{i,j}$ is the energy consumption of transmitting one bit of data from s_i to s_j, $\boldsymbol{F}$ represents the feasible regions. Eq. (7) means that, at each sink site, the data received by the sink equals to the data sensed by SNs. Eq. (8) represents that, at each sink site, the outgoing data of each SN equals to the sum of its sensed data and received data. Inequality (9) means that the total energy consumed by each SN cannot exceed its initial energy. Inequality (10) denotes that the step distance of the sink should not exceed R_v. Eq. (11) states that all sink sites should be in feasible regions. Inequalities (12) to (14) are the non-negative constraints for all variables. Inequalities (9) to (10) indicate that the optimization problem is nonlinear and difficult to solve by traditional deterministic algorithms.

3. THE PROPOSED ACO-MSS

3.1 Ant Colony Optimization

In nature, ants work collaboratively to search good paths between their nest and food sources. They deposit pheromones on the ground to mark certain favorable paths when traveling. Subsequent ants are more likely to follow the paths with a larger amount of pheromone. In this way, ants can endeavor to find the shortest path efficiently. ACO mimics this approach to solve optimization problems.

In an ACO algorithm, a group of artificial ants search solutions by traveling a connected graph whose nodes represent the solution components and edges represent the possible moving directions. The connected graph, namely, the construction graph, expresses the structure of the problem. Each solution can be represented by a path on the graph. The pheromone trails record the historical exploring experience of the entire colony, while the heuristic information forms problem-specific knowledge.

3.2 Heuristic Information Definition

We consider following three factors to guide the movements of the sink.

1) *Average residual energy*: The average residual energy evaluates the residual energy of SNs near a candidate sink site. This factor is used to guide the sink to regions containing SNs with more residual energy. Let $N(c_i)$ be the set of SNs which are within the communication range of a candidate sink site c_i. The *average residual energy* of c_i is defined as

$$U_{c_i} = \frac{\sum_{z_j \in N(c_i)} E_j}{|N(c_i)|} \tag{15}$$

2) *Average communication hop*: In order to reduce the energy consumption, it is desirable to place the sink at regions with a smaller number of communication hops. Denote the distance between c_i and z_j as $d(c_i, z_j)$ and the communication range of z_j as R. The lower bound of communication hops for transmitting data from z_i to c_i can be computed by

$$k_{ij} = \left\lceil \frac{d(c_i, z_j)}{R} \right\rceil \tag{16}$$

Accordingly, the *average communication hops* of c_i can be approximated by

$$V_{c_i} = \frac{\sum_{j=1}^{n} \left\lceil \frac{d(c_i, z_j)}{R} \right\rceil}{n} \tag{17}$$

3) *Maximum Step Distance*: As the sink does not receive data when moving, the step distance of the sink should be smaller than or equal to a maximum step distance R_v, so as to avoid long latency. We describe this factor by

$$D_{c_i} = \begin{cases} 1, \text{ if } d(c_i, z_0) \le R_v \\ 0, \text{ otherwise} \end{cases} \tag{18}$$

where z_0 is the current position of the sink.

With regard to the above factors, the heuristic information to guide the movement of the sink is defined as

$$\eta_{c_i} = U_{c_i} \cdot V_{c_i}^{-\alpha} \cdot D_{c_i} \tag{19}$$

where α is a predefined constant.

3.3 Implementation of ACO-MSS

In ACO-MSS, a solution to the MSS problem consists of two parts. As expressed in Eq. (14), the first part encodes the positions and visiting orders of sink sites, while the second part encodes the corresponding routing strategies of sink sites.

$$\left\{ \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \ldots, \begin{bmatrix} x_m \\ y_m \end{bmatrix} \begin{bmatrix} \begin{bmatrix} u_{0,0}^1 & \cdots & u_{0,n}^1 \\ \cdots & \cdots & \cdots \\ u_{n,0}^1 & \cdots & u_{n,n}^1 \end{bmatrix}, \begin{bmatrix} u_{0,0}^2 & \cdots & u_{0,n}^2 \\ \cdots & \cdots & \cdots \\ u_{n,0}^2 & \cdots & u_{n,n}^2 \end{bmatrix} \ldots, \begin{bmatrix} u_{0,0}^m & \cdots & u_{0,n}^m \\ \cdots & \cdots & \cdots \\ u_{n,0}^m & \cdots & u_{n,n}^m \end{bmatrix} \end{bmatrix} \right\} \tag{20}$$

where $\begin{bmatrix} x_i \\ y_i \end{bmatrix}$ represents the positions of z_i and $u_{i,j}^k$ represents the data rate from z_i to z_j when the sink is at s_k. The first part contains the positions and visiting order of sink sites, while the second part contains the flow routing associated with each sink site.

In order to avoid setting the sink sites at forbidden regions, the whole monitoring region is divided into $W \times H$ discrete grids. Those grids with their center points in the forbidden regions are regarded as infeasible grids, whereas the others are feasible grids. The center points of all feasible grids comprise the candidate sink sites. These candidate sink sites form a connected graph, where the nodes represent candidate sink sites and the edges represent the possible moving directions of the sink.

The algorithm framework of ACO-MSS consists of three main steps, i.e., the initialization, solution construction, and global pheromone updating, are involved. The implementation details of ACO-MSS are described as follows.

Step1: Initialization

The initialization firstly obtains a best-so-far solution P_{best} by a greedy algorithm. In the greedy algorithm, the sink is initially placed at a feasible grid at the center regions. Then it makes movements in rounds. At the beginning of each round, the sink moves to a feasible candidate sink site with the largest heuristic value. The flow routing associated with the chosen sink site is obtained by the FA routing strategy [4]. Then the sink keeps collecting data at the sink site for a period of Δt . When the duration of Δt is reached, the sink stops collecting data and continues to choose a next sink site. The above process is repeated until at least one SN has not enough residual energy to work for the next period. After obtaining P_{best} by the greedy algorithm, pheromone of edges on the construction graph is initialized by

$$\zeta_{i,j} = T_0, \quad \forall i \in [1,n], \forall j \in [1,n] \tag{21}$$

where $\zeta_{i,j}$ is the pheromone on the edge connecting c_i and c_j, T_0 is the network lifetime of P_{best}.

Step2: Solutions Construction

In this step, each of the *NP* artificial ants constructs a solution by the following process. First, the artificial ant is placed at an initial candidate sink site such as the one in the center region. Second, the ant chooses a next sink site s_k by

$$s_k = \begin{cases} \max\limits_{c_j}\{p_{c_j}\}, \text{ if } rand(0,1) < q_0 \\ \text{proportion - selection, otherwise} \end{cases} \tag{22}$$

where p_{c_j} is the selection probability of c_j. The value of p_{c_j} is computed by

$$p_{c_j} = \frac{\zeta_{i,j} \cdot \eta_{c_j}^{\beta}}{\sum_{k=1}^{M} (\zeta_{i,k} \cdot \eta_{c_k}^{\beta})} \tag{23}$$

where M is the number of candidate sink sites, i represents the index of the candidate sink site where the sink is currently at, β is a predefined constant, η_{c_j} is the heuristic information of c_j. The proportion-selection chooses a sink site based on the selection probabilities of candidate sink sites, using the roulette wheel selection method [22]. Third, when the next sink site s_k is determined, the artificial ant moves to s_k, updates the flow routing (i.e., $u_{i,j}^k, \forall i \in [0,n], \forall j \in [0,n]$) using the FA method [4] and keeps the sink collecting data at s_k for a period of Δt . Fourth, when the working period of Δt is reached, the heuristic information of all candidate sink sites is updated and the artificial ant continues to choose a next sink sites. The ant terminates the solution construction process when at least one SN has not enough energy to work for a period of Δt .

Once an artificial ant finishes the solution construction process, the pheromone values are updated by

$$\zeta_{i,j} = (1-\rho) \cdot \zeta_{i,j} + \rho \cdot T_k, \forall \overline{c_i c_j} \in P_k \tag{24}$$

where P_k is the solution found by the artificial ant, T_k is the network lifetime of P_k. In this way, pheromone on the traveling paths of better ants would be increased, while pheromone on the traveling paths of worse ants would be decreased.

Step3: Global Pheromone Update

The global pheromone update operation takes place when all ants finish constructing solutions. The goal is to increase the pheromone deposited along the traveling path of P_{best}, so that subsequent ants can exploit promising solutions more efficiently. The global pheromone update operation can be expressed as

$$\zeta_{i,j} = (1-\rho) \cdot \zeta_{i,j} + \rho \cdot T_{best}, \forall \overline{c_i c_j} \in P_{best} \tag{25}$$

where T_{best} is the lifetime found by the best-so-far ant.

There is a repetition from the second step to the third step until a termination condition is met.

4. SIMULATIONS AND COMPARISONS

4.1 Simulation Settings

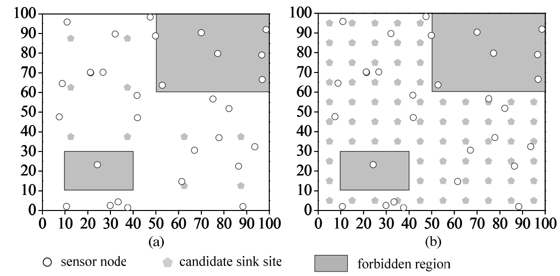

Figure 2. Monitoring region and the candidate sink sites of compared algorithms ((a) candidate sites of the LP(C-MB), (b) candidate sites of the GMRE and ACO-MSS).

In this section, we apply the ACO-MSS on seven random WSNs, where SNs are deployed in a 100m × 100m rectangle monitoring region with two regular forbidden regions, as shown in Fig. 2. We compare ACO-MSS with two other algorithms, namely, the LP (C-MB) [14] and the GMRE [18]. The LP (C-MB) is a linear programming method which aims to optimize the sojourn time and the data flow of each sink site [14]. The GMRE is a greedy-based mobile sink algorithm which finds a solution by applying a step-by-step selection policy. At each step, the GMRE greedily chooses a next location which contains the SNs with the most residual energy. Since the GMRE does not optimize the flow routing of the network, we use the FA routing strategy in GMRE as in ACO-MSS.

The candidate sink sites of all compared algorithms are shown in Fig. 2. To reduce the memory requirement of LP (C-MB), we divide the monitoring region into 4 × 4 discrete grids, and the 10 feasible grids are used as the candidate sink sits of the LP (C-MB). As in [21], parameters of the energy consumption are set to be δ = 50nJ/bit, a = 50nJ/bit, b = 100pJ/bit/m^2, and v = 2. The initial energy, the sensing data rate, and the communication range of each SN, are set to be 20J, 1bit/s, and 30m respectively. The maximum moving distance of the sink is set to be 30m. The communication range and Δt are set to be 30m and 3600s respectively.

The parameter settings of ACO-MSS are listed in Table 2. Since ACO-MSS is a stochastic algorithm, we perform it for 30 independent runs on each test case and use the mathematical mean for comparison. All algorithms are programmed using

Visual C++ 6.0 and are run on a PC with Intel (R) Core ™2 Quad CPU Q6600, 2.4 GHz, 1.96GB of RAM.

Table 2. Parameter settings of ACO-MSS

name	value	Summary
NP	10	Number of artificial ants
q_0	0.95	Exploitation rate
α	10	The weight of the *average communication hops*
ρ	0.5	Pheromone reinforcement rate
β	0.1	The weight of heuristic information
FES	1000	Maximum number of fitness evaluations

4.2 Result Comparisons

Table 3. Comparison results on the seven test cases

n	LP(C-MB)		GMRE		ACO-MSS	
	T	*CPU*	*T*	*CPU*	*T* (*Std.*)	*CPU*(*Std.*)
30	**36.7**	0.624	19	0.005	35.9 (0.55)	1.97(0.04)
50	**49.3**	4.084	18	0.006	44.0(0.26)	3.71(0.06)
100	N/A	N/A	39	0.020	**57.6(0.81)**	9.49(0.14)
200	N/A	N/A	67	0.109	**76.8(1.72)**	40.57(0.60)
300	N/A	N/A	73	0.256	**82.73(1.48)**	101.71(1.56)
400	N/A	N/A	84	0.585	**97.37(2.06)**	239.53(5.45)
500	N/A	N/A	76	0.843	**101.2(2.92)**	393.8(14.63)

The comparison results are listed in Table 3, where *T* represents the average network lifetime found by the algorithm, and *CPU* represents the average running time (in second) of each algorithm. It can be observed that LP(C-MB) obtains the best network lifetime on the 1st and 2nd cases. However, due to the large memory requirement, LP(C-MB) fails to work on the other test cases. According to the network lifetime, the proposed ACO-MSS performs better than GMRE on all test cases. With regard to the CPU time, GMRE requires the least CPU times, while ACO-MSS requires the largest but still acceptable.

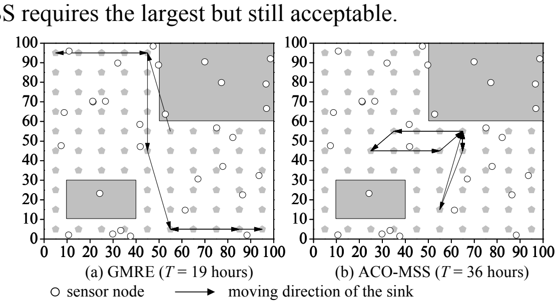

Figure 3. Moving paths of the sink found by GMRE and ACO-MSS in the 1st test case.

Fig. 3 illustrates the moving paths of the sink found by GMRE and ACO-MSS in the 1st test case. It can be observed that the sink would move to the edge regions in the solution found by GMRE, while the sink mainly visits the center regions in the solution found by ACO-MSS. According to the final network lifetime, the solution found by ACO-MSS is much better than that found by the GMRE.

5. CONCLUSIONS

This paper studied the mobile sink scheduling (MSS) problem in WSNs, aiming to maximize the lifetime of WSNs by exploiting the sink mobility. A mathematic formulation is presented to describe the MSS problem, in which multiple practical factors are considered, including the positions of sink sites, the flow routing and the maximum moving distance of the sink. An ant colony optimization algorithm, namely the ACO-MSS, is developed to tackle the problem. The proposed ACO-MSS makes use of the global search ability of ACO and newly designed heuristic information to search for near globally optimal solutions. We validated the proposed ACO-MSS by conducting simulations on WSN with forbidden regions. Simulation results demonstrate that the proposed ACO-MSS have a very promising performance for solving the MSS problem. Future research work includes the following: 1) considering more objectives such as minimizing the total moving distance of the mobile sink, 2) relaxing some simplified assumptions of the formulation, e.g., considering multiple mobile sinks or bringing the finite link transmission rate into the formulation, and 3) apply adaptive parameter control strategies or other discrete evolutionary algorithms [31]-[34] to solve the MSS problem.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Science Fund for Distinguished Young Scholars No.61125205, National Natural Science Foundation of China No.61070004 and NSFC Joint Fund with Guangdong under Key Project U0835002.

7. REFERENCES

[1] D. Puccinelli, M. Haenggi, "Wireless sensor networks: applications and challenges of ubiquitous sensing," *IEEE Circuits and Systems Magazine*, vol.5, no.3, pp. 19- 31, 2005.

[2] A. Pantelopoulos, N. G. Bourbakis, "A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol.40, no.1, pp.1-12, Jan. 2010.

[3] M. Haenggi, "Opportunities and challenges in Wireless sensor Networks," in *Handbook of Sensor Network: Compact Wireless and Wired Sensing Systems*, M. llyas and I. Mahgoub, eds., Boca Raton, FL, pp. 1.1-1.14, CRC Press, 2004.

[4] J. H. Chang, L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol.12, no.4, pp. 609- 619, Aug. 2004.

[5] Y. Lin, J. ZHANG, et al., "An Ant Colony Optimization Approach for Maximizing the Lifetime of Heterogeneous Wireless Sensor Networks", *IEEE Transactions on Systems, Man, and Cybernetics--Part C: Applications and and Reviews* (In press)

[6] J. H. Zhong and J. Zhang, "Energy-efficient local wake-up scheduling in wireless sensor networks," *2011 IEEE Congress on Evolutionary Computation (CEC),* pp.2280-2284, 5-8 June 2011.

[7] X. M. Hu, J. Zhang, and *et al*., "Hybrid Genetic Algorithm Using a Forward Encoding Scheme for Lifetime Maximization of Wireless Sensor Networks," *IEEE Transactions on Evolutionary Computation*, vol.14, no.5, pp.766-781, Oct. 2010.

[8] R. C. Shah, S. Roy, S. Jain, W. Brunette, "Data MULEs: modeling a three-tier architecture for sparse sensor networks," *Proc. 1th IEEE International Workshop on Sensor Network Protocols and Applications 2003*, pp. 30- 41, May 2003.

[9] K. H. S. Kim, T. F. Abdelzaher, and W. H. Kwon, "Minimum energy asynchronous dissemination to mobile sinks in wireless sensor networks," in *Proceedings of the First International Conference on Embedded Networked Sensor Systems, SenSys 2003*, Los Angeles, CA, pp. 193–204, November, 5–7 2003.

[10] Y.Z. Bi, L.M. Sun, J. Ma, N. Li, I.A. Khan, and C.F. Chen, "HUMS: An Autonomous Moving Strategy for Mobile Sinks in Data-Gathering Sensor Networks," *EURASIP J. Wireless Comm. and Networking*, pp. 1-15, 2007.

[11] M. Marta, M. Cardei, "Improved sensor network lifetime with multiple mobile sinks", *Elsevier Journal of Pervasive and Mobile Computing*, vol. 5 no. 5, pp. 542– 555, October 2009.

[12] Y. Z. Bi, and et al., “Moving Schemes for Mobile Sinks in Wireless Sensor Networks,” *Performance, Computing, and Communications Conference, 2007. IPCCC 2007. IEEE Internationa* , pp.101-108, 11-13 April 2007.

[13] Y. S. Yun, Y. Xia, “Maximizing the Lifetime of Wireless Sensor Networks with Mobile Sink in Delay-Tolerant Applications,” *IEEE Transactions on Mobile Computing*, vol.9, no.9, pp.1308-1318, Sept. 2010.

[14] Y. Shi and Y. T. Hou, “Theoretical Results on Base Station Movement Problem for Sensor Network,” *Proc. IEEE INFOCOM, 2008.*

[15] Z. M. Wang, S. Basagni, E. Melachrinoudis, C. Petrioli, “Exploiting Sink Mobility for Maximizing Sensor Networks Lifetime,” *Proc. 38th Annual Hawaii International Conference on System Sciences 2005 (HICSS '05)*, pp. 287a, 03-06 Jan. 2005.

[16] I. Papadimitriou and L. Georgiadis, “Maximum Lifetime Routing to Mobile Sink in Wireless Sensor Networks,” *Proc. 13th IEEE International Conference onsoftware, Telecommunications and Computer Networks, SoftCOM 2005, Split, Croatia, September 2005.*

[17] M. Gatzianas, L. Georgiadis, “A Distributed Algorithm for Maximum Lifetime Routing in Sensor Networks with Mobile Sink,” *IEEE Transactions on Wireless Communications*, vol.7, no.3, pp.984-994, March 2008.

[18] S. Basagni, A. Carosi, and et al., “A New MILP Formulation and Distributed Protocols for Wireless Sensor Networks Lifetime Maximization,” *Proc. IEEE International Conference on Communications 2006* , pp.3517-3524, June 2006

[19] M. Dorigo and T. Stutzle, *Ant Colony Optimization*. Cambridge, MA: MIT Press, 2004.

[20] M. Dorigo, L. M. Gambardella, “Ant colony system: a cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on Evolutionary Computation*, vol.1, no.1, pp.53-66, Apr 1997.

[21] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy efficient communication protocols for wireless microsensor networks,” in *Proc, 33rd International Conference on System Sciences(HICSS 2000),* vol. 8, pp. 8020-8029, January 2000.

[22] J. H. Zhong, X. M. Hu, J. Zhang and M. Gu, “Comparison of Performance between Different Selection Strategies on Simple Genetic Algorithms,” in *Proc. IEEE Computational Intelligence for Modeling, Control and Automation 2005,* vol.2, pp. 1115-1121, Nov. 2005.

[23] Z. H. Zhan, J. Zhang, and Yun Li, “An Efficient Ant Colony System Based on Receding Horizon Control for the Aircraft Arrival Sequencing and Scheduling Problem”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 399-412, June 2010.

[24] W. N. Chen, J. Zhang, and H. Chung, “Optimizing Discounted Cash Flows in Project Scheduling--An Ant Colony Optimization Approach”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40 No. 5 pp. 64-77, Jan. 2010.

[25] W. N. Chen, J. ZHANG, and et al., “Optimizing Discounted Cash Flows in Project Scheduling--An Ant Colony Optimization Approach”, *IEEE Transactions on Systems, Man, and Cybernetics--Part C: Applications and Reviews* Vol.40 No.1 pp.64-77, Jan. 2010

[26] X. M. Hu, J. ZHANG, and et al., “An Intelligent Testing System Embedded with an Ant Colony Optimization Based Test Composition Method”, *IEEE Transactions on Systems, Man, and Cybernetics--Part C: Applications and Reviews,* Vol. 39, No. 6, pp. 659-669, Dec 2009.

[27] J. ZHANG, H. Chung, W. L. Lo, and T. Huang, “Extended Ant Colony Optimization Algorithm for Power Electronic Circuit Design”, *IEEE Transactions on Power Electronic*. Vol.24,No.1, pp.147-162, Jan 2009.

[28] W. N. Chen and J. ZHANG, “Ant Colony Optimization Approach to Grid Workflow Scheduling Problem with Various QoS Requirements”, *IEEE Transactions on Systems, Man, and Cybernetics--Part C: Applications and Reviews,* Vol. 31, No. 1,pp.29-43,Jan 2009.

[29] X. M. Hu, J. ZHANG and et al., “Protein Folding in Hydrophobic-Polar Lattice Model: A Flexible Ant- Colony Optimization Approach ”, *Protein and Peptide Letters,* Volume 15, Number 5, 2008, Pp. 469-477

[30] J. ZHANG, X. M. Hu, and et al., “Implementation of an Ant Colony optimization technique for Job Scheduling Problem ”, *Transactions of the Institute of Measurement and Control* 28 ,1(2006) pp. 93-108.

[31] J. ZHANG, and et al., “Evolutionary Computation Meets Machine Learning: A Survey”, *IEEE Computational Intelligence Magazine*, pp.68-75, NOVEMBER 2011.

[32] Z. H. Zhan, and et al., “Adaptive Particle Swarm Optimization”, *IEEE Transactions on Systems, Man, and Cybernetics--Part B*. VOL. 39, NO. 6, Dec 2009, Page 1362-1381.

[33] J. ZHANG, H. Chung and W. L. LO, “Clustering-Based Adaptive Crossover and Mutation Probabilities for Genetic Algorithms”, *IEEE Transactions on Evolutionary Computation* Vol.11, No.3, June 2007 ，Page. 326-335.

[34] W. N. Chen, and et al., “A Novel Set-Based Particle Swarm Optimization Method for Discrete Optimization Problems”, *IEEE Transactions on Evolutionary Computation*, Vol.14,No.2, pp.278-300, April 2010.