Theoretical Treatment of Sink Scheduling Problem in Wireless Sensor Networks

Yu Gu*, Yusheng Ji*, Jie Li¶, and Baohua Zhao†‡
*Information Systems Architecture Science Research Division, National Institute of Informatics, Tokyo, Japan
†State Key Laboratory of Networking and Switching Technology, Beijing, 100876, China
‡School of Computer Science, Univ. of Science and Technology of China, Hefei, Anhui, 230027, China
¶Department of Computer Science, University of Tsukuba, Tsukuba Science City, Ibaraki 305-8573, Japan

***Abstract*—Sink Scheduling, in the form of scheduling multiple sinks among sink sites to leverage traffic burden, is an effective mechanism for the energy-efficiency of wireless sensor networks (WSNs). Due to the inherent difficulty (NP-hard in general), existing works on this topic mainly focus on heuristic/greedy algorithms and theoretic results remain unknown. In this paper, we fill in the research blank with two algorithms. The first one is based on the Column Generation (CG). It decomposes the original problem into two sub problems and solve them iteratively to approach the optimal solution. However, due to its high computational complexity, this algorithm is only suitable for small scale networks. The other one is a polynomial-time algorithm based on relaxation techniques to obtain an upperbound, which can serve as a performance benchmark for other algorithms on this problem. Through comprehensive simulations, we evaluate the efficiency of proposed algorithms.**

I. Introduction

Sink scheduling, in the form of scheduling multiple sinks among sink sites to leverage traffic burden and enhance energy efficiency, is shown to be a promising method in wireless sensor networks. However, the problem of optimizing the network lifetime via sink scheduling remains quite a challenge since routing issues are tightly coupled.

Gandham *et al.* first challenged this problem [1]. Due to the combinational complexity caused by sinks scheduling and data routing, instead of optimizing the network lifetime directly, they divided it into rounds, and an Integer Linear Programming (ILP) formulation has been constructed to minimize the maximum energy spent by a sensor in a round. Wang *et al.* also realized such complexity and relaxed the problem by considering scheduling and routing separately [2]. Performance of the proposed algorithms is questionable due to the lack of joint optimizations. Therefore, Luo *et al.* analyzed the problem based on joint optimizations [3]. They assumed that there is only one sink with unlimited mobility. With the same network model, Yi *et al.* developed the first approximation algorithm with performance guarantee [4]. Their work are quite enlightening. But considering that we use a different but also practical network model, which involves multiple sinks with limited mobility, their results may not be applied directly.

In summary, for the sink scheduling problem, the existing approaches either have no performance guarantee [1] [2] [5] or are under different network models [3] [6] [4]. Therefore, theoretic results remain unknown. In this paper, we aim to fill in the research blank.

First, by defining a novel notation Placement Pattern (PP) to bind routes with the placement of sinks, we mathematically formulate the problem in a cross-layer way. However, since the set of all patterns (P) is unknown, it cannot be solved directly. To meet the obstacle, a computationally efficient alternative is proposed based on the column generation. It decomposes the original optimization into two related sub optimizations - the master and sub problem. The master problem has a similar structure with the original optimization and can output a solution with a subset $P_0 (P_0 \subset P)$. Though P is unknown, P_0 is easy to get. The sub problem justifies optimality of the current solution. If the answer is yes, the procedure terminates. Otherwise, the sub problem will supplement current P_0 with another PP which would cause a maximum improvement over current solution. Then the master problem is solved again with the updated P_0. The iterative procedure goes on until the optimal solution has been found.

Due to inherent difficulties associated with the sink mobility problem, the CG-based algorithm has a high computational complexity. Thus, it is designed for small-scale networks. For large-scale networks, we develop a polynomial-time algorithm by relaxing mobility constraints to obtain an upperbound. This algorithm can serve as a performance benchmark for other algorithms.

To characterize the impact of sink mobility on network performance and study the efficiency of our proposed algorithms, simulations have been conducted. Simulation results substantiate the importance of using sink mobility for energy-constrained sensor networks, which echoes with other papers on this topic [1] [2] [3] [4].

For the CG-based algorithm, simulations show that it is quite effective for small-scale networks. However, as problem scale increases, computational complexity grows rapidly.

By comparing the bound with the lifetime achieved by the CG-based algorithm, we show that the proposed upperbound is quite tight in general. Therefore, it can serve as a performance benchmark for other algorithms.

The rest of the paper is organized as follows. We describe our problem statement in Section II. In Section III, we mathematically reformulate the problem. In Section IV, a column generation based approach is proposed to solve the

978-1-4244-9920-5/11/$26.00 ©2011 IEEE

pattern-based formulation. In Section V, a lifetime upperbound is developed. Simulation results are reported in Section VI. Finally, Section VII concludes the paper.

II. Problem Statement

We introduce following notations:

V_s = set of sinks, and $|V_s| = K$.

V = set of sensors, and $|V| = N$.

V_o = set of sink sites, and $|V_o| = M$.

$L = \{V \times V\}$, set of wireless links. $l_{i,j} = 1$, if sensor j is within the communication range (r_i) of sensor i, otherwise $l_{i,j} = 0$. The capacity of each link in L is $C_{i,j}$.

$L_o = \{V \times V_o\}$, set of potential wireless links between sensors and sink sites. $l_{i,o} = 1$, if sink site o is within the communication range of sensor i, otherwise $l_{i,o} = 0$. The capacity of each link in L_o is $C_{i,o}$.

e_i^T, e_i^R, e_i^S = energy required for transmitting, receiving, sensing per unit data for sensor i.

θ_i = data generation rate for node i.

E_i = initial residual energy for node i.

Like previous literatures on this topic [1] [2] [7], we do not explicitly consider radio interferences, i.e., we assume that underlay MACs like TDMA [8] have been used to eliminate the radio interference among communications.

The spatial distribution of sensor network can be modeled as an directed graph $G = \{V \bigcup V_o, L \bigcup L_o\}$.

Definition 1: *Network Lifetime (T).* The network lifetime (T) is defined as the elapsed time since the launch of this network till the instant that the first node dies.

Thus, the problem of our concern can be stated as:

Problem Statement *Optimal Sink Scheduling (OSS) Problem.* Given a network topology G, find out sink and sensor schedules to maximize network lifetime.

In the next section, we will mathematically formulate the OSS problem.

III. Problem Formulation

In this section, first we will define a Placement Pattern. Based on it, we mathematically formulate the problem.

A placement pattern p involves two issues: the placement of sinks as well as routes from sensors to sinks. If only one pattern is used, the network lifetime equals to the time duration until the first node in this pattern dies. However, due to spatial redundancy of sink sites, lifetime can be optimized by switching among different patterns to relieve traffic burden on a specific set of sensors, e.g. sensors around a sink.

To mathematically describe a p, we define $x_{s,o}^p$ as the binary variable that is set to 1 if sink s resides in site o in p and 0 otherwise and t_p as the sojourn time assigned to p. Define $g_{i,j}^p$ and $g_{i,o}^p$ as the data rates from sensor i to j and i to site o in pattern p. Define e_i^p as the energy consumption rate in p, to mathematically characterize a pattern p, we list the constraints as following,

$$\sum_{s \in V_s} x_{s,o}^p \leq 1, \forall o \in V_o \tag{1}$$

$$\sum_{o \in V_o} x_{s,o}^p \leq 1, \forall s \in V_s \tag{2}$$

$$\sum_{(i,o) \in L_o} g_{i,o}^p + \sum_{(i,j) \in L} g_{i,j}^p = \sum_{(j,i) \in L} g_{j,i}^p + \theta_i, \forall i \in V \tag{3}$$

$$g_{i,j}^p \leq C_{i,j}, \forall l_{i,j} \in L \tag{4}$$

$$g_{i,o}^p \leq C_{i,o} \cdot \sum_{s \in V_s} x_{s,o}, \forall (i,o) \in L_o \tag{5}$$

$$\begin{aligned} e_i^p = & \sum_{(i,o) \in L_o} g_{i,o}^p \cdot e_i^T + \sum_{(i,j) \in L_s} g_{i,j}^p \cdot e_i^T \\ & + \theta_i \cdot e_i^N + \sum_{(j,i) \in L_s} g_{j,i}^p \cdot e_i^R, \forall i \in V \end{aligned} \tag{6}$$

Eq. (1) denotes that one sink site can be occupied by at most one sink in pattern p.

Eq. (2) ensures that one sink can reside in at most one sink site in pattern p.

Eq. (3) guarantees flow balance, i.e. for sensor i, data transferred from i to its neighbors and sinks equals to data received from its neighbors plus data generated by itself.

Eqs. (4) and (5) ensure that data flow over one link can not exceed its capacity.

Eq. (6) denotes the energy consumption rate for sensor i in pattern p.

Based on the above constraints, we can define the placement pattern formally as following,

Definition 2: *Placement Pattern (PP).* A placement pattern is defined as $p = \{x_{s,o}^p, g_{i,j}^p, \forall s \in V_s, o \in V_0 \text{ and } i,j \in V\}$, where $x_{s,o}^p, g_{i,j}^p$ support Eqs.(1)-(6).

To optimize network lifetime, we can discretize time into time durations with different lengths, and for each duration we assign a placement pattern to the network. We name such a solution as a pattern-based solution, which can be formally defined as following,

Definition 3: *A Pattern-based Solution, PBS.* A pattern-based solution is defined as $PBS = \{< p_1, t_1 >, \cdots, < p_k, t_k >, t_k \geq 0\}$, where p_k is a placement pattern and t_k is the corresponding time duration assigned.

Therefore, the problem can be reformulated in a pattern-based way as follows:

$$\text{(PB)} \qquad Max(T = \sum_{p \in P} t_p)$$

subject to

$$\sum_{p \in P} e_i^p \cdot t_p \leq E_i, \forall i \in V \tag{7}$$

$$t_p \geq 0, \forall i \in V, p \in P \tag{8}$$

There are two major problems about this formulation, namely,

TABLE I
NETWORK PARAMETERS

Network Parameter	Value
θ_i	1
E_i	1000
$C_{i,j}$, $C_{i,p}$	5, 5
e_i^T, e_i^R, e_i^S	2,1,1

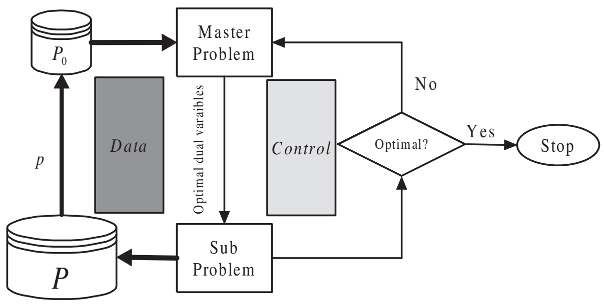


Fig. 1. The CG-based Approach

1) P, which contains all patterns, is unknown. Furthermore, its cardinality might be exponential to the number of wireless links.
2) This formulation is a mixed integer non-Linear programming (MINLP) [9].

Nevertheless, directly solving this *MINLP* formulation is quite difficult. In the next section, we will propose a CG-based algorithm to tackle the difficulties.

IV. A COLUMN GENERATION APPROACH

CG is a general purpose framework which has been often proposed as a computationally efficient alternative to standard integer optimization methods [10]. In our case, columns correspond to patterns, and the column generation based approach helps to reduce the complexity in constructing the whole set of patterns, by effectively selecting columns that make improvements to the optimization.

A. The Initial Set of Patterns P_0

The CG based approach works in feasible domain and requires some initial basic feasible solutions to start with. The effect of this approach can be enhanced by the quality of the initial Basic Feasible Solutions (BFS). In this paper, we propose an optimization with the objective of minimizing the summing of energy consumption rate of each sensor to generate initial BFS. The optimization will be called $Init$ hereafter.

$$\text{(Init)} \qquad Min(\sum_{i \in S} e_i^p)$$

subject to Eq. (1)- Eq. (6) and

$$g_{i,j}^p, g_{i,o}^p \geq 0, x_{s,o}^p = \{0,1\}, \forall i,j \in V, o \in V_o, s \in V_s \quad (9)$$

The $Init$ generates an energy-balanced pattern, therefore it tends to provide a high starting point for the CG-based algorithm and shorten the computational time. Simulations also confirm its efficiency.

TABLE II
SOLVE THE SAMPLE TOPOLOGY USING $INIT$

IT	CT	c_p^*
1	83.33	0.75
2	100.00	0.70
3	133.33	$-0.5 \times e^{-8}$

B. The Master Problem and The Sub Problem

The set of patterns P is quite difficult to enumerate, but a subset $P_0 (P_0 \subset P)$ is easy to get. For example, besides the $Init$, we can randomly assign sinks to sites and perform the shortest-path algorithm for each sensor to get routes to sinks. Therefore, assume that we have such a subset P_0, we can reformulate *PB* as a *Master* problem:

$$(\textit{Master}) \quad Max \sum_{p \in P_0} t_p \quad (10)$$

subject to

$$\sum_{p \in P_0} e_i^p \cdot t_p \leq E_i, \forall i \in V \quad (11)$$

$$t_p \geq 0, \quad \forall i \in V, p \in P_0 \quad (12)$$

The master problem is a linear programming problem and can be solved easily with a standard simplex algorithm. After solving the master problem, we verify its optimality by identifying whether it can be improved by adding new columns (patterns) to the current BFS. Denote $\tilde{B}_i$ as the optimal dual variables for the energy constraint (Eq.11) in the master problem, the reduced cost c_p for the variable t_p corresponding to pattern p is then:

$$c_p = 1 - \sum_{s_i \in S} \tilde{B}_i \cdot e_i^p \quad (13)$$

Clearly, we want to select the column that results in the maximum non-negative cost reduction c_p^* and join it into the current BFS, i.e., $P_0 = P_0 \bigcup p$, where c_p^* is obtained by solving the subproblem:

$$(\textit{Sub}) \quad Max(c_p) \quad (14)$$

subject to Eq. (1)- Eq. (6) and

$$g_{i,j}^p, g_{i,o}^p \geq 0, x_{s,o}^p = \{0,1\}, \forall i,j \in V, o \in V_o, s \in V_s \quad (15)$$

If the solution to the subproblem results in a negative reduced cost, then the previous result from the master problem is already optimal to the original problem and CG terminates. Otherwise, the master problem is solved again with the new BFS, and the whole procedure is repeated until the optimal solution has been found. The algorithm is shown in Fig.1.

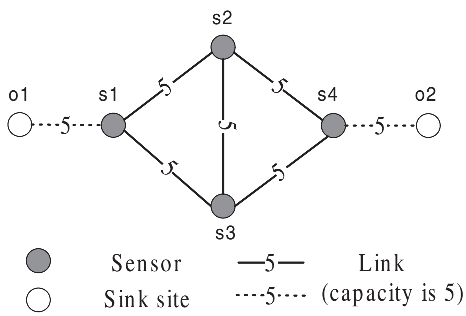


Fig. 2. The Sample Topology

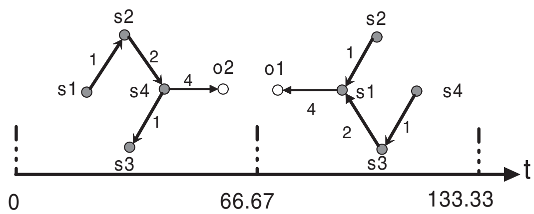


Fig. 3. The optimal solution that CG outputs with $INIT$

C. An Example

We solve the topology in Fig.2 to show how the proposed algorithm works. Network parameters are listed in Tab.I. Results are concluded in Tab.II.

It takes the proposed approach less than 1 second (3 iterations) to find the optimal solution (as shown in Fig.3. More specifically, first we use the *INIT* to generate a pattern where the sink resides in o_2, s_2 relays data for s_1 while s_4 relays data from s_2 and s_3. Then using this pattern as P_0, the master problem gets a lifetime 83.33 and the sub problem has a maximum non-negative cost reduction 0.75, which means that this solution is not optimal. Thus the sub problem extends the current P_0 with the pattern corresponding to the maximum non-negative cost reduction and the master problem is solve again with the updated P_0. The procedure goes on until in the $3rd$ iteration, the master problem obtains a lifetime 133.33 while the sub problem reports that the maximum cost reduction is negative, which means that current solution is optimal. As shown in Fig.3, the optimal solution contains two patterns, each is working for time 66.67. The first pattern is the initial pattern we generated. In the other pattern, the sink resides in o_1, s_3 relays data for s_4 while s_1 relays data from s_2 and s_3.

D. Algorithm Analysis

Though the CG-based algorithm can achieve optimality, it has a high computational complexity. The master problem is a LP formulation, which can be solved in polynomial time, however, the sub problem is an Integer Programming (IP) problem and relatively difficult to solve. Besides, number of iterations needed is unknown and somehow related to $|P|$, which may be exponential to number of wireless links. Simulations also confirm that though the proposed algorithm works quite well with small-scale networks, as problem scale grows, computational time grows rapidly. Therefore, the proposed algorithm fills in the theoretic blank but its usage in practical systems is limited, especially considering that sensor networks are resource-limited platforms.

Therefore, in the next section, we will propose an effective upper bound based on the relaxation technique. The major purpose is to set a performance benchmark for other algorithms on this topic. By using the CG-based algorithm to test the proposed upperbound, we show that this upperbound is quite tight in general.

V. An LP-based Upperbound

The major difficulty of the OSS problem is the combinational complexity caused by sink placement and routing issues. More specifically, we want to find the exact sink schedules as well as routes. However, if we only consider the sojourn time a sink resides in a site as well as the corresponding routes for sensors, the problem can be relaxed and formulated as a linear programming problem.

We define x_s^o as the total time sink s stays in site o, $f_{i,o}$ as the total data from sensor i to site o, $f_{i,j}$ as the total data from i to j. We have the following optimization to obtain the lifetime upperbound.

$$\text{(UPP)} \qquad Max(T) \tag{16}$$

subject to

$$\sum_{p \in V_p} x_s^o \le T, \forall s \in V_s \tag{17}$$

$$\sum_{b \in V_b} x_s^o \le T, \forall s \in V_s \tag{18}$$

$$\sum_{(i,o) \in L_o} f_{i,o} + \sum_{(i,j) \in L_s} f_{i,j} = \sum_{(j,i) \in L} f_{j,i} + \theta_i \cdot T, \forall i \in V \tag{19}$$

$$f_{i,j} \le C_{i,j} \cdot T, \forall (i,j) \in V \tag{20}$$

$$f_{i,o} \le C_{i,o} \cdot \sum_{(s,o) \in L_o} x_s^o, \forall (i,o) \in V_p \tag{21}$$

$$\begin{aligned} &\sum_{(i,o) \in L_o} f_{i,o} \cdot e_i^T + \sum_{(i,j) \in L} f_{i,j} \cdot e_i^T \\ &+\theta_i \cdot T \cdot e_i^S + \sum_{(j,i) \in L} f_{j,i} \cdot e_i^R \\ &\le E_i, \forall i \in V \end{aligned} \tag{22}$$

$$\begin{aligned} &f_{i,j}, f_{i,o}, x_s^o \ge 0, \\ &\forall s \in V_s, o \in V_o, (i,o) \in L_o; \forall (i,j) \in L \end{aligned} \tag{23}$$

Eq. (16) is referred to as UPP optimization problem hereafter.

Eq. (17) denotes that the total time a sink is active can not exceed the network lifetime T.

Eq. (18) denotes that the total time a sink site is occupied can not exceed the network lifetime T.

Eq. (19) guarantees flow balance, i.e. for sensor i, data from it to its neighbors and sinks equals to data received from its neighbors plus data generated by itself.

Eqs. (20) and (21) ensure that data flow over one link can not exceed its capacity.

Eq. (22) ensures that for sensor i, the total energy consumption would not exceed its initial energy E.

The computational complexity of UPP is $O((N^2+NM+KN)^3)$ [11]. Clearly, output of UPP may not be feasible since UPP ignores exact assignments of sensors and sinks. Therefore it may not be used directly to provide a feasible solution. However, as an upper bound, simulations show that UPP is quite tight in general.

VI. Numerical Results

To verify the efficiency of the proposed algorithms and study the impact of sink mobility, we have built a simulator using Visual Studio 2005 and LINGO 8.0 [12].

A. A Case study

As shown in Fig.4(a), 30 sensors and 5 sink sites are randomly distributed in a square area (100×100). Lines among sensors constitute L while lines between sink sites and sensors constitute L_o. First, we set K from 1 to 5. Using $INIT$, and the results are shown in Fig.4(b) and Fig.4(c).

First, this case study suggests that **for a given topology the increment on lifetime cannot be guaranteed by adding sinks.**. For example, if we set $K=1$, the optimal lifetime is 38.46. If given another sink, the lifetime will be prolonged to 107.00, corresponding to an increment of 178.19%. However, adding the third sink only improves the network lifetime to 111.11, corresponding to an increment of 3.84%. The fourth and fifth sink do not bring any improvement at all.

Second, there is another phenomenon worth mentioning, when we set $K=4$ and 5, it only needs 1 iteration and less than 1 second to achieve the optimality. We further study the optimal solutions and find out that **they all contain the same placement pattern. This pattern happens to be the output of the $INIT$.** In further experiments, we notice that this phenomenon happens occasionally.

To confirm investigations obtained in the case study, we conduct and implement more experiments in the next section.

B. Further Results

For network parameters $<N=10,M=3>$, $<N=20,M=3>$,$<N=20,M=5>$,$<N=30,M=5>$,$<N=30,M=10>$,$<N=40,M=10>$, we randomly generate 10 instances for each $<N,M>$. For each instance of $<N,M>$, we increase K from 1 to M. Other network parameters remain the same.

We study the impact of increasing sinks in a given network topology with fixed N and M. Intuitively, for a given network topology, the more sinks we have, the longer lifetime we can achieve. We evaluate this statement via simulations. As shown

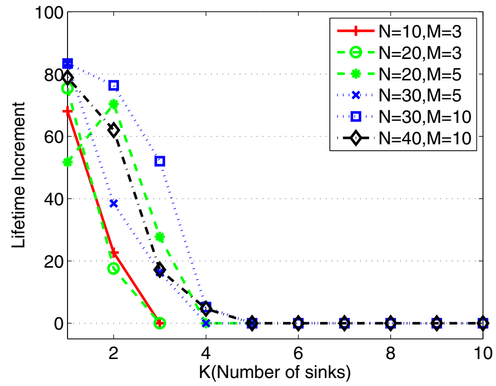


Fig. 5. Average lifetime increment of increasing sinks

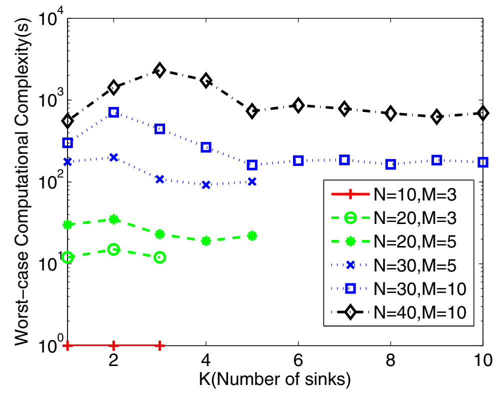


Fig. 6. Worst-case computational complexity

in Fig.5, we average the lifetime increment of increasing sinks on 10 randomly generated instances over each $<N,M>$. One importance observation is that **the first few sinks causes the most significant increment on network lifetime**. Then, the lifetime cannot be further improved. We take the simulations with $N=30,M=10$ as an example (The green line with square markers in Fig.5). Averagely, the first sink will bring a lifetime increment of 83.40, the second and third sink

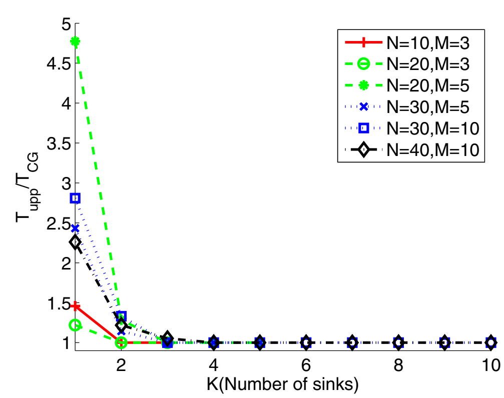
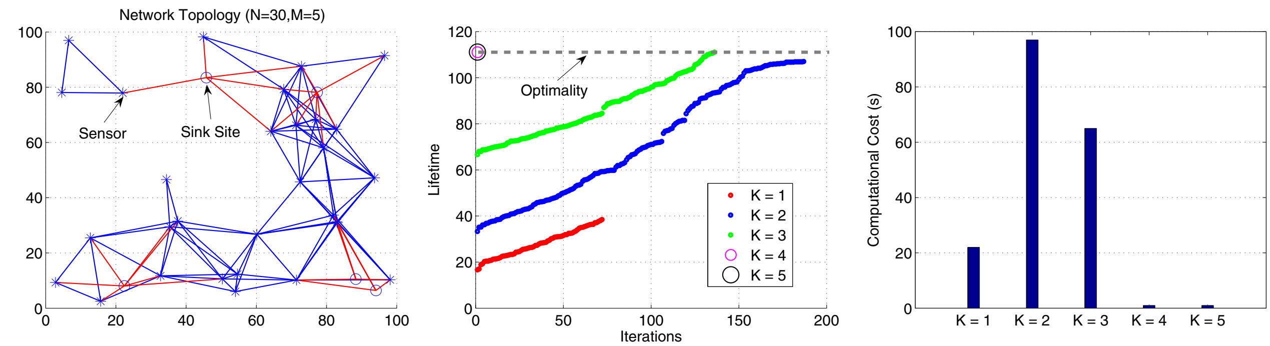


Fig. 7. Performance of the upperbound

Fig. 4. The case study with $N = 30, M = 5$



(a) Network Topology with $N = 30, M = 5$ (b) The iterative procedures with different K (c) The computational times for different K

will cause an increment of 76.37 and 52.03, and the fourth sink only improve the lifetime by 5.15. Then, the lifetime cannot be extended if we keep increasing sinks. This phenomenon echoes with the previous case study.

For the computational complexity, for each $< N, M, K >$, we average the computational time over 10 instances and show the results in Fig.6. The computational time grows exponentially with the problem scale and this backs up our conjecture that the multiple sink mobility problem belongs to NP-hard. For example, if $N = 10, M = 3$, the computational time is around 1 second, however, when we set $N = 40, M = 10$, the average computational time is around 1000 seconds.

For the UPP, all instances have been solved in less than 3 seconds. Compared to the CG-based algorithm, it is much faster. To study its performance, for each instance, we get a performance ratio by dividing the optimal solution (T_{CG}) the CG-based algorithm outputs by the upper bound (T_{UPP}). For each $< N, M, K >$, we obtain an average performance ratio over 10 instances and the results are shown in Fig.7. We see that the ratio is very close to 1 for any $< N, M, K >$ with $K \geq 2$, which means that UPP is quite tight in general. Therefore, we can conclude that UPP is a fast and effective performance benchmark.

VII. Conclusion

We studied the optimal sink scheduling problem in wireless sensor networks. The problem is inherently difficult since we need to consider both link layer (i.e. scheduling sinks) and network layer (i.e. routing issues). Based on a novel notation named Placement Pattern, the problem has been formulated in a cross-layer way and an iterative CG-based approach has been proposed to tackle the optimization. We also developed an effective upperbound by relaxing mobility constraints. Simulations not only confirmed efficiency of the proposed algorithms, but also showed interesting properties of the problem.

Acknowledgements

This work has been supported by the Grand-in-Aid for Scientific Research of Japan Society for Promotion of Science (JSPS).

References

[1] S. R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, "Energy efficient schemes for wireless sensor networks with multiple mobile base stations," in *Proceeding of IEEE GLOBECOM 2003*, vol. 22, San Francisco, CA, USA, Dec. 2003, pp. 377–381.

[2] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, "Exploiting sink mobility for maximizing sensor networks lifetime," in *Proc. of 38th Hawaii International Conference on System Sciences*, Big Island, Hawaii, USA, Jan. 2005.

[3] J. Luo and J. P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *Proceeding of IEEE INFOCOM 2005*, Miami, FL, USA, 2005.

[4] Y. Shi and Y. T. Hou, "Theoretical results on base station movement problem for sensor network," in *Proceeding of IEEE Infocom 2008*, Phoenix, AZ, USA, 2008.

[5] E. Guney, N. Aras, I. K. Altinel, and C. Ersoy, "Efficient integer programming formulations for optimum sink location and routing in heterogeneous wireless sensor networks," *Comput. Netw.*, vol. 54, no. 11, pp. 1805–1822, 2010.

[6] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and Z. M. Wang, "Controlled sink mobility for prolonging wireless sensor networks lifetime," *ACM/Springer Wireless Networks*, vol. 14, pp. 831–858, 2007.

[7] Y. Gu, Y. Ji, J. Li, H. Chen, B. Zhao, and F. Liu, "Towards an optimal sink placement in wireless sensor networks," in *Proc. of ICC'10*, Cape Town, South Africa, May 2010.

[8] G. Pei and C. Chien, "Low power tdma in large wireless sensor networks," in *Proceeding of IEEE MilCom 2001*, Washington, D.C., USA, 2001, pp. 347–351.

[9] H. T. R. Horst, *Global optimization: deterministic approaches*. Berlin: Springer-Verlag, 1996.

[10] H. Vance, C. Barnhart, E. Johnson, and GLAW, "Solving binary cutting stock problems by column generation and branch-and-bound," *Computational Optimization and Applications*, vol. 3, pp. 111–130, May 1994.

[11] Y. Ye, "An $O(n^3 \cdot l)$ potential reduction algorithm for linear programming," *Mathematical Programming*, vol. 50, pp. 239–258, 1991.

[12] [Online]. Available: http://www.lindo.com