Gu Y, Zhao BH, Ji YS *et al.* Theoretical treatment of target coverage in wireless sensor networks. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 26(1): 117–129 Jan. 2011. DOI 10.1007/s11390-011-1115-x

Theoretical Treatment of Target Coverage in Wireless Sensor Networks

Yu Gu[1] (谷 雨), Bao-Hua Zhao[1,2,*] (赵保华), Yu-Sheng Ji[3] (计宇生), *Member, IEEE*
and Jie Li[4] (李 颉), *Senior Member, ACM, IEEE*

[1] *School of Computer Science, University of Science and Technology of China, Hefei 230027, China*

[2] *State Key Laboratory of Networking and Switching Technology, Beijing 100876, China*

[3] *Information Systems Architecture Science Research Division, National Institute of Informatics, Tokyo, Japan*

[4] *Department of Computer Science, University of Tsukuba, Tsukuba Science City, Ibaraki, Japan*

E-mail: guyu@nii.ac.jp; bhzhao@ustc.edu.cn; kei@nii.ac.jp; lijie@cs.tsukuba.ac.jp

Received December 27, 2009; revised October 26, 2010.

Abstract The target coverage is an important yet challenging problem in wireless sensor networks, especially when both coverage and energy constraints should be taken into account. Due to its nonlinear nature, previous studies of this problem have mainly focused on heuristic algorithms; the theoretical bound remains unknown. Moreover, the most popular method used in the previous literature, i.e., discretization of continuous time, has yet to be justified. This paper fills in these gaps with two theoretical results. The first one is a formal justification for the method. We use a simple example to illustrate the procedure of transforming a solution in time domain into a corresponding solution in the pattern domain with the same network lifetime and obtain two key observations. After that, we formally prove these two observations and use them as the basis to justify the method. The second result is an algorithm that can guarantee the network lifetime to be at least $(1-\epsilon)$ of the optimal network lifetime, where ϵ can be made arbitrarily small depending on the required precision. The algorithm is based on the column generation (CG) theory, which decomposes the original problem into two sub-problems and iteratively solves them in a way that approaches the optimal solution. Moreover, we developed several constructive approaches to further optimize the algorithm. Numerical results verify the efficiency of our CG-based algorithm.

Keywords target coverage, wireless sensor networks, time-dependent solution, pattern-based solution, column generation

1 Introduction

In wireless sensor networks deployed for monitoring discrete physical targets with QoS requirements[1], considering that sensors are typically battery-driven and have energy constraints[2], so the problem of optimizing the network lifetime while fulfilling those QoS requirements becomes an interesting challenge. Numerous methods have been proposed to address target coverage problems with different network settings[3-10].

Among them [5] is one of the most important studies. The basic idea is that continuous time can be divided into discrete time slots with different lengths. In each time slot, only one coverage pattern, defined as a subset of sensors that can cover all targets, is activated while the remaining sensors are put into a sleep state to save energy. The problem has been mathematically formulated on the basis of this idea. However, this method has two weak points. The first is that the idea of discretization of continuous time has not been justified; a formal proof is missing. The other weak point, the more critical one, is that the formulation involves mixed integer nonlinear programming (MINLP), which makes it very difficult to tackle the formulation[11], especially for a resource limited sensor platform. In fact, the authors of that paper failed to solve it directly, but just proposed two heuristic algorithms instead. Their solutions are suboptimal and are likely to be far away from the optimum in certain cases, leaving the exact performance bound unknown.

This paper aims to fill in these theoretical gaps by providing both a justification of the method for discretization of continuous time and an efficient algorithm with a performance guarantee for this problem.

First, we justify the method of discretization of continuous time utilized in [5]. We notice that the target coverage is inherently involved with time issues,

---

Regular Paper

This work has been partially supported by the National Natural Science Foundation of China under Grant Nos. 60872009, 6002016, the Hi-Tech Research and Development 863 Program of China under Grant Nos. 2007AA01Z428, 2009AA01Z148, the Post Doctoral Fellowship (ID No. P10356) for Scientific Research of Japan Society for Promotion of Science (JSPS).

*Corresponding Author

e.g., sensors are active in different time spans and in charge of different targets. Therefore, it is more natural to formulate the problem in a time-dependent manner. However, the formulation belongs to the class of non-polynomial programming problems and is NP-hard in general. Therefore, before seeking an optimal solution to this formulation, we first gain critical insights into an investigation of a particular time-dependent target coverage problem. We will show how this problem is transformed into a pattern-based coverage with the same optimal network lifetime. Then, we prove these insights in general case and thus establish a formal justification for the method of discretization of continuous time.

Second, we try to develop an efficient solution to the MINLP formulation. Considering that the major difficulty in solving the original MINLP formulation in [5] is that P, containing all coverage patterns, might be extremely large so explicit enumeration becomes intractable as the size of the network grows. To deal with this problem, we employ a column generation approach to decompose the original MINLP formulation into master and sub-problems. We first solve the master formulation, using P_0, which is a subset of P. Note that P_0 can be easily obtained by a random selection algorithm as shown in Section 4. Moreover, the master problem belongs to the class of linear programming and is relatively easier to solve optimally[12]. Clearly, since we only use a subset of patterns, the solution to the master problem serves as a lower bound for the overall optimal lifetime. Then, we develop a novel method of obtaining an upper bound for the overall lifetime based on the sub-problem. If the difference between these two bounds satisfies a predefined precision, the algorithm terminates. Otherwise, we add a new pattern, which may have the most significant contribution according to the sub-problem, to P_0 and solve the master problem again. Therefore, by iteratively shrinking the distance between lower and upper bounds, the algorithm can guarantee the network lifetime is at least $(1-\epsilon)$ of the optimal network lifetime. A detailed analysis and formal proof are given.

We also develop several constructive methods in order to accelerate the iterative process. The first one is a novel random selection algorithm to generate P_0; it helps to speed up the convergence procedure of our approach. Note that the sub-problem is an integer programming (IP) problem and may be difficult to solve in a resource limited environment, e.g., a sensor platform. Therefore, instead of directly solving the IP formulation, we re-formulate and solve it by using a linear relaxation technique coupled with a rounding algorithm, which significantly reduces computation complexity while maintaining a certain level of performance. We also offer an in-depth study and give a formal proof to these methods.

The contributions of this paper are two-fold:

1) We formally justify the method of discretizing continuous time. In other words, we prove that as far as network lifetime is concerned, we can convert the time-dependent formulation to the pattern-based formulation.

2) We propose a CG-based approach with a performance guarantee to solve the pattern-based formulation. It provides a critical performance benchmark when evaluating other heuristic algorithms for target coverage problems in WSNs (wireless sensor networks), e.g., the algorithms proposed in [4-6].

The rest of the paper is organized as follows. We describe our system model and problem formulation in Section 2. We formally justify the method of discretizing continuous time in Section 3. Section 4 describes the column generation approach. Numerical simulation results are reported in Section 5. We introduce some related work in Section 6. Finally, Section 7 concludes the paper.

2 System Model and Time-Dependent Problem Formulation

In this section, we will introduce our system model and formulate the target coverage problem in a time-dependent manner.

2.1 System Model

We consider a set of n sensor nodes: $S = \{s_1, s_2, \ldots, s_n\}$ deployed to cover m targets: $R = \{r_1, r_2, \ldots, r_m\}$. The coverage requirement is that at any given moment, target r_k is covered by at least one sensor node. Sensor node s_i has an initial energy E_i ($i = 1, \ldots, n$). First, we define a notation named Target Coverage Graph.

Definition 1 (Target Coverage Graph). *A target coverage graph is a bipartite graph $TCG = \{S, L, R\}$ where there exists a link $l_{i,k} \in L$ if sensor node s_i can cover target r_k, for any $s_i \in S$ and $r_k \in R$. Note that for s_i, any target within its sensing range can be covered by itself.*

Considering that we do not take data delivery into consideration, without loss of generality, we assume that there are only two states for sensors: active and sleep. The energy consumption rate for a certain sensor s_i is e_i when it is in active state. Otherwise, the energy consumption is so small when it is asleep compared to when it is active that we can simply neglect it[2]. The network lifetime is defined as the elapsed time since the launch of the sensor network till the instant that there exists some target $r_k \in R$, to which no live sensor can be assigned.

Accordingly, the problem can be formally stated as:

Problem Statement: *Target Coverage* (*TC*) *Problem*. Given a target coverage graph $TCG = \{S, L, R\}$, maximize the network lifetime.

The target coverage problem, as proved in [5], belongs to the NP-complete.

Theorem 1[5]. *The TC problem is NP-complete.*

In the next subsection, we will formulate the problem in a time-dependent manner.

2.2 Time-Dependent (TD) Formulation

TC inherently involves time issues and thus should naturally be formulated in a time-dependent manner.

$x_i(t)$ denotes the indicator that should be set to 1 if sensor s_i is active at time instant t, and 0 otherwise.

The coverage constraint on each target $r_k \in R$ is

$$\sum_{s_i \in U_k} x_i(t) \geqslant 1, \tag{1}$$

where $U_k = \{s_i | s_i \text{ can cover } r_k\}$, i.e., for target r_k, at any time instant t, at least one sensor that can cover r_k should be active.

Similarly, the energy constraint on each sensor $s_i \in S$ is

$$\int_0^T x_i(t) \cdot e_i \leqslant E_i, \tag{2}$$

i.e., for sensor s_i, the total energy consumed due to covering targets over the lifetime T cannot exceed the initial energy E_i.

Therefore, the problem can be formulated as follows:

$$\text{(TD)} \quad \max(T) \tag{3}$$

subject to

$$\sum_{s_i \in U_k} x_i(t) \geqslant 1 \quad (\forall r_k \in R, 0 \leqslant t \leqslant T) \tag{4}$$

$$\int_0^T x_i(t) \cdot e_i \leqslant E_i \quad (\forall s_i \in S) \tag{5}$$

and

$$x_i(t) = \{0, 1\}, T \geqslant 0 \quad (\forall s_i \in S). \tag{6}$$

This formulation is in the form of a non-polynomial programming problem. Note that even a non-linear programming problem (a special case of a non-polynomial programming problem) belongs to the NP-complete[11], thus we conclude that the above non-polynomial programming problem is NP-complete. We define the following notation:

Definition 2 (Time-Dependent Solution). *A time-dependent solution is a solution* $\psi = \{T, \boldsymbol{x}(\boldsymbol{t}) = \{x_i(t), \forall s_i \in S\}\}$, *where* T *and* $\boldsymbol{x}(\boldsymbol{t})$ *satisfy* (4), (5) *and* (6).

In the next section, we will show that given a time-dependent formulation, we can always transform it into a pattern-based formulation.

3 Pattern-Based Problem Formulation

We will first gain two critical insights into a single example. Then, we will prove these insights in general cases and convert the time dependent (TD) formulation into a pattern-based (PB) one.

3.1 Observations via an Example

For convenience, we use the example topology presented in [5] as shown in Fig.1. There are four sensors $S = \{s_1, s_2, s_3, s_4\}$ deployed to cover three targets $R = \{r_1, r_2, r_3\}$. Each sensor has an identical residual energy 100 and energy consumption rate 1 when active.

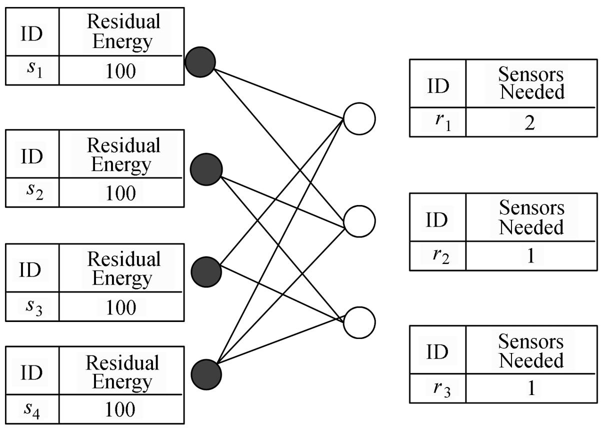


Fig.1. Example target coverage graph[5].

First, we need to define the meaning of coverage pattern. Similar to [5], we identify a subset of sensor nodes that meet target coverage constraint as a *coverage pattern*. Let x_i^p indicate whether a sensor node s_i is in coverage pattern p or not.

According to this definition, the coverage constraint should be characterized as follows/

$$\sum_{s_i \in U_k} x_i^p \geqslant 1 \quad (\forall r_k \in R, p \in P) \tag{7}$$

i.e., for target r_k, at least one sensor that can cover r_k should be in the pattern p.

We define another notation as follows.

Definition 3 (Pattern-Based Solution). *A pattern-based solution* $\omega = \{T, \langle \boldsymbol{p}, \boldsymbol{t} \rangle = \{\langle p_1, t_1 \rangle, \ldots, \langle p_u, t_u \rangle\}\}$ *is defined as a sequence of coverage patterns, each of*

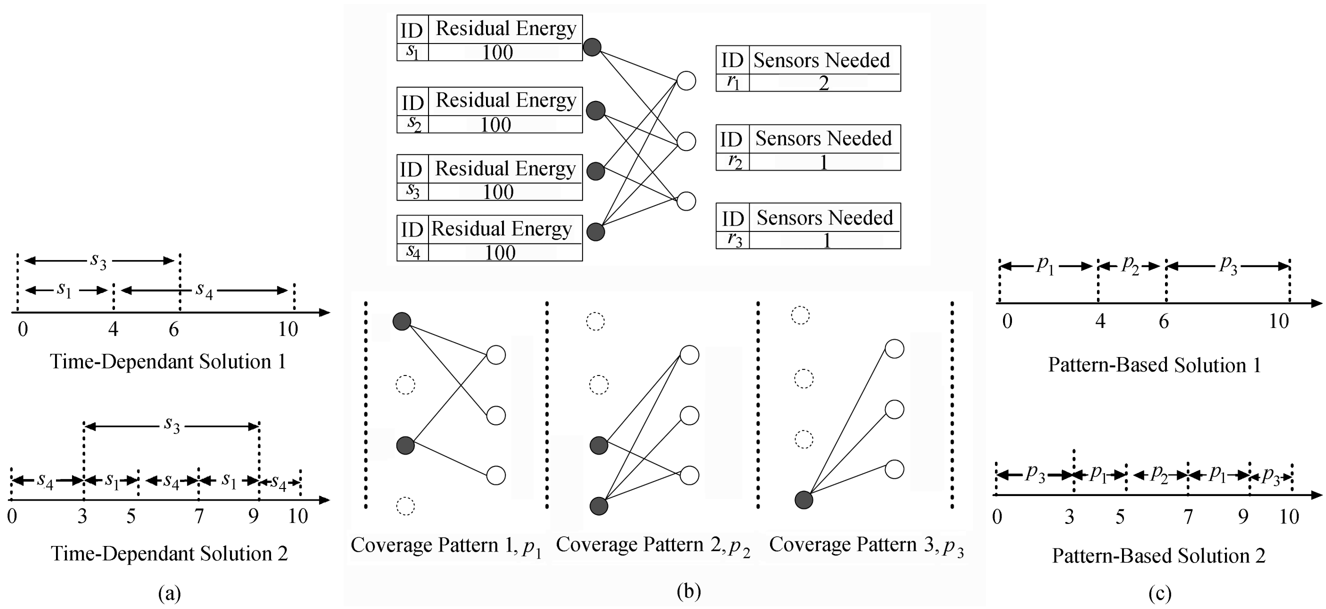


Fig.2. Example demonstrating that time-dependent solutions can be transformed into pattern-based solutions. (a) Time-dependent solutions. (b) Coverage patterns. (c) Pattern-based solutions.

which will be assigned a time duration. In this assigned time duration, only the corresponding coverage pattern is active and covering all targets; all the other sensors not in this pattern stay asleep.

Note that according to the above definition, any coverage pattern is capable of covering all targets. In this way, the coverage constraints can be fulfilled by using only one coverage pattern at a time.

As shown in Fig.2(a), assume that we have two different time-dependent solutions TDS_1 and TDS_2, which we show later can be transformed to corresponding pattern-based solutions.

More specifically, let us take a look at TDS_1, where s_1, s_3, and s_4 are active from time 0 to 4, 0 to 6, and 4 to 10, respectively. It is easy to verify that TDS_1 satisfies the coverage constraints. In other words, at any time span, any target is covered by at least one active sensor. Now, we take another look at TDS_1. We notice that from time 0 to 4, there are two active sensors that cover three targets: s_1 and s_4, which constitute a coverage pattern $p_1 = \{s_1, s_3\}$ according to the definition. Similarly, we have coverage pattern $p_2 = \{s_3, s_4\}$ and $p_3 = \{s_4\}$ active in time durations 4 to 6 and 6 to 10, respectively. These three patterns are shown in Fig.2(b). Therefore, a time-dependent solution TDS_1 can be transformed to a pattern-based solution PBS_1, as shown in Fig.2(c). Moreover, the network lifetime, 10, remains the same. The same procedure can be applied to another time-dependent solution TDS_1 (see Fig.2(a)) and the corresponding pattern-based solution is shown in Fig.2(c).

Now, we further study the pattern-based solutions PBS_1 and PBS_2. At first glance, there are two totally different solutions in the sense of ordering and frequency of the patterns' presence. However, we notice that actually for p_1, p_2 and p_3, they have been used for exactly 4, 2 and 4 time units for PBS_1 and PBS_2, respectively. It is quite straightforward that PBS_1 and PBS_2 share the same network lifetime. In other words, given a pattern-based solution, as long as the sojourn time assigned to patterns remains the same, the network lifetime remains the same regardless of the ordering and frequency of the patterns' presence.

In summary, there are two important observations to be derived, namely:

1) Given a time-dependent solution, we can constitute a pattern-based solution with the same network lifetime.

2) Given a pattern-based solution, as long as the sojourn time assigned to the patterns remains the same, the same network lifetimes remain the same regardless of the ordering and frequency of the patterns' presence.

In the next subsection, we will prove that these two observations hold not only in the example here, but also in general cases.

3.2 Theoretical Analysis: From Time-Dependent to Pattern-Based

First, we theoretically prove that the first observation obtained in the previous subsection is true in a more general case. After that, we give a brief analysis of the second observation.

Lemma 1. *Given a time-dependent solution, we* *can always constitute a pattern-based solution with the same network lifetime.*

Proof. The proof of this lemma follows the idea used in the previous subsection when we try to transform TDS_1 into PBS_1.

Formally, we assume that we have a time-dependent solution $\psi = \{T, \boldsymbol{x(t)}\}$ and do the following to transform it into a pattern-based solution.

Transformation. For time instant j, $0 \leqslant j \leqslant T$, find all the s_i that are active, e.g., $x_i(j) = 1$; they constitute a pattern p_j.

Merge. For time instant j and $j + \Delta j$ with active pattern p_j and $p_{j+\Delta j}$ $(0 \leqslant j \leqslant T, \Delta j \to 0)$, if p_j and $p_{j+\Delta j}$ are identical, merge $p_{j+\Delta j}$ into p_j and increase the time duration t_j for p_j by Δj.

After the above two actions, we now have a pattern-based solution $\omega = \{T, \langle \boldsymbol{p}, \boldsymbol{t} \rangle = \{\cdots, \langle p_j, t_j \rangle, \cdots\}\}$.

Note that at any time instant i, the time-dependent solution ψ obeys,

$$\sum_{s_i \in U_k} x_i(t) \geqslant 1 \quad (\forall r_k \in R, 0 \leqslant t \leqslant T) \tag{8}$$

Therefore, according to the definition, the sensors that are active at t constitute a coverage pattern.

Regarding the energy constraints, since the above transformation does not change the status of a sensor at t, the pattern-based solution has the same network lifetime as the time-dependent solution. Clearly, the lifetimes for both ψ and ω are the same. □

We prove the second observation as follows.

Lemma 2. *Given a pattern-based solution, as long as the sojourn time assigned to the patterns remains the same, the network lifetime remains the same regardless of the ordering and frequency of patterns' presence.*

Proof. The proof is based on contradiction. Assume we have two different pattern-based solutions: $PBS_1 = \{\langle p_{11}, t_{11} \rangle, \cdots, \langle p_{1k_1}, t_{1k_1} \rangle\}$ with lifetime T_1 and $PBS_2 = \{\langle p_{21}, t_{21} \rangle, \cdots, \langle p_{2k_2}, t_{2k_2} \rangle\}$ with lifetime $T_2 \neq T_1$, where the patterns and corresponding sum of assigned time duration are the same but the ordering and frequency of the patterns' presence are different. In such a case, the following equation must hold:

$$T_1 = \sum_{i=1}^{k_1} t_{1i} = \sum_{j=1}^{k_2} t_{2j} = T_2. \tag{9}$$

As a result, we have $T_1 = T_2$, which introduces a contradiction and the proof is finished. □

Clearly, by combining the above two lemmas, we reach the following conclusion.

Theorem 2. *The time-dependent solution corresponds to the pattern-based solution in the sense of having the same optimal network lifetime.*

Therefore, based on the above theorem, we define a notation P, which contains all coverage patterns, and the pattern-based PB formulation transformed from the time-dependent formulation TD can be as follows:

$$\text{(PB)} \quad Max\Big(\sum_{p \in P} t_p\Big) \tag{10}$$

subject to

$$\sum_{p \in P} e_i^p \cdot t_p \leqslant E_i \quad (\forall s_i \in S, p \in P) \tag{11}$$

and

$$e_i^p, t_p \geqslant 0 \quad (\forall s_i \in S, p \in P) \tag{12}$$

where t_p denotes the time duration assigned to a pattern p, and P is the set containing all patterns. (10) is referred to as PB optimization problem hereafter.

(11) guarantees that for every sensor node $s_i \in S$, its total energy consumption would not exceed the initial energy E_i.

Note that, in TD, the integral symbol $\int_0^T \cdot$ indicates it is time-dependent while in PB, the summation symbol $\sum_{p \in P} \cdot$ shows it is pattern-based. Till now, we show that the fundamental formulation (i.e., TD) of target coverage problem can be seamlessly transformed to a pattern-based one (i.e., PB), which exactly verifies the idea of discretization of continuous time in [5] and fills in the first theoretical gap.

Clearly, the major difficulty in solving the PB formulation is that P, containing all coverage patterns, might be extremely large so explicit enumeration becomes intractable as the size of the network grows. Moreover, its cardinality might be exponential to the number of links between the sensors and targets, e.g., a full connection TCG, where each sensor can cover all targets.

To overcome this obstacle, based on the column generation approach, we develop a novel optimization framework with the idea of iteratively shrinking the distance between the upper and lower bounds, which could guarantee that the network lifetime is at least $(1 - \epsilon)$ of the optimal network lifetime, where ϵ can be made arbitrarily small depending on the required precision.

We will cover details of this method in the next section.

4 $(1 - \epsilon)$ Algorithm for TC Problem

4.1 Algorithm Overview

The framework is illustrated as follows.

1) Generate an initial basic feasible set (BFS) P_0, which follows $P_0 \subset P$, e.g., randomly generate some coverage patterns that constitute P_0.

2) Solve the master formulation with BFS, whose solution ($T_{\rm low}$) serves as a lower bound for the optimal lifetime T.

3) Solve the sub-problem with optimal dual variables $\boldsymbol{B}$ in 2) and generate a new coverage pattern p.

4) Obtain current lifetime upper bound ($T_{\rm upp}$).

5) If $\frac{T_{\rm low}}{T_{\rm upp}} < 1 - \epsilon$, return $T_{\rm low}$, else, update the current BFS by adding p into it. Go to 2).

Column generation (CG) is a general-purpose framework that has often been used as a computationally efficient alternative to standard integer optimization methods and as a modeling tool when a direct approach is infeasible[13-14]. In our case, the columns correspond to patterns, and the column generation approach helps to reduce the complexity of constructing the whole set of patterns by effectively selecting columns that make improvements in the optimization.

In the next subsection, we illustrate details of this algorithm.

4.2 Initial Basic Feasible Solutions

The CG approach works in a feasible domain and requires initial feasible solutions to start with. Its effect can be enhanced by increasing the quality of the initial BFS. Therefore, for fast convergence, it is important to develop methods of obtaining a good initial BFS. Here, we use the random selection algorithm proposed in our previous paper[15]. The complexity of the random selection algorithm is $O(n)$, where n is the number of sensors. The more the patterns, the faster this CG approach converges. Thus it is preferable to use RSA to generate multiple initial coverage patterns.

4.3 Upper and Lower Bounds

Assume we have an initial BFS P_0 derived from the above algorithm. We can reformulate the PB optimization problem as a Master problem:

$$\text{(Master)} \quad Max \sum_{p \in P_0} t_p \tag{13}$$

subject to

$$\sum_{p \in P_0} e_i^p \cdot t_p \leqslant E_i \quad (\forall s_i \in S) \tag{14}$$

and

$$t_p \geqslant 0 \quad (\forall s_i \in S, p \in P). \tag{15}$$

Master is a restriction of the PB problem, whose optimal solution $T_{\rm low}$ serves as a lower bound of the PB problem. Note that Master is a classical LP problem and can be efficiently solved with the standard simplex algorithm.

$\tilde{B}_i$ denotes the optimal dual variables for the energy constraint (14) in the Master problem, the reduced cost c_p for the variable t_p corresponding to coverage pattern p is then:

$$c_p = 1 - \sum_{s_i \in S} \tilde{B}_i \cdot e_i^p. \tag{16}$$

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 Sub-problem:

$$\text{(Sub)} \quad Max(c_p) \tag{17}$$

subject to (7) and (12), where we set λ_i as the optimal dual variables for the energy constraint (14) in Master.

The following conclusion helps us achieve a lifetime upper bound.

Lemma 3[16]**.** *For the column generation approach, if there exists T_{con} and it follows,*

$$T_{con} \geqslant \sum_{p \in P} t_p \tag{18}$$

the upper bound T_{upp} for this problem would hold,

$$T_{upp} = c_p \cdot T_{con} + T_{low}. \tag{19}$$

Thus, it is reasonable to obtain another lifetime upper bound by relaxing the convergence constraint that every target should be covered by at least one sensor at any given moment to one that every target should be covered by at least one sensor on average.

First, we define T as the network lifetime and y_i as the total time when sensor s_i is active. We formulate the following linear programming formulation named CON to achieve T_{con}.

$$\text{(CON)} \quad Max(T) \tag{20}$$

subject to

$$\sum_{s_i \in U_k} y_i \geqslant \cdot T \quad (\forall r_k \in R), \tag{21}$$

$$y_i \cdot e_i \leqslant E_i \quad (\forall s_i \in S), \tag{22}$$

$$y_i \leqslant T \quad (\forall s_i \in S), \tag{23}$$

$$y_i \geqslant 0, \ T \geqslant 0 \quad (\forall s_i \in S, \ r_k \in R). \tag{24}$$

(21) ensures that for every target $r_k \in R$, there would be at least one sensor covering it on average.

(22) guarantees that for every sensor node $s_i \in S$, the total energy consumption does not exceed the initial energy E_i.

(23) makes sure that for each $s_i \in U_k$, its time when covers some target $r_k \in R$ does not exceed the network lifetime, i.e., T.

Theorem 3. *The optimal solution T_{con} to the above CON formulation satisfies*:

$$T_{con} \geqslant \sum_{p \in P} t_p. \tag{25}$$

Thus, given an initial basic feasible set P_0, the optimal lifetime T^* to PB lies between the above $T_{\rm upp}$ and $T_{\rm low}$.

The following relationship holds:

$$\frac{T_{\rm low}}{T^*} \leqslant \frac{T_{\rm low}}{T_{\rm upp}}. \tag{26}$$

Our algorithm terminates and outputs $T_{\rm low}$ if the following relationship is satisfied:

$$\frac{T_{\rm low}}{T_{\rm upp}} < 1 - \epsilon \tag{27}$$

where ϵ can be made arbitrarily small depending on the required precision.

Otherwise, we conclude that P_0 is not good enough to characterize P, therefore, a new coverage pattern should be added to the current BFS. In particular, we find such a pattern by solving the Sub-problem.

The above considerations lead to the following conclusion.

Theorem 4. *The CG algorithm guarantees the network lifetime is at least $(1-\epsilon)$ of the optimal network lifetime, where ϵ can be made arbitrarily small depending on the required precision.*

4.4 Further Optimization of the Sub-Problem

Notice that the Sub-problem is an integer programming (IP) problem belonging to the NP-complete. It is relatively easy to solve this IP formulation in a simulation environment by using standard techniques like the branch-and-bound algorithm[11]; however, such resource-consuming techniques are usually unavailable in a real-world sensor system. The algorithms developed for solving linear programming problems, like the revised simplex method, can be better implemented in a sensor node since they work efficiently on a resource-limited platform.

Thus, we shall develop an $O(m^3n^3)$ ρ-approximation algorithm for the Sub-problem where $\rho = 1 + \max |U_k|$. First, the original Sub-problem is relaxed to a Linear Programming formulation named as LP-Sub. Then we propose a novel LP-Sub-based rounding algorithm and prove that it is a ρ-approximation algorithm for the Sub-problem.

We can relax the Sub-problem as follows (we use x_i instead of x_i^p hereafter for convenience):

$$\text{(LP-Sub)}\ Min\Big(\sum_{s_i \in S} \tilde{B}_i \cdot e_i\Big) \tag{28}$$

subject to

$$\sum_{s_i \in U_k} x_i \geqslant 1 \quad (\forall s_i \in S, r_k \in R) \tag{29}$$

$$0 \leqslant x_i \leqslant 1. \tag{30}$$

Table 1. CG with $P_0 = \{s_4\}$

Iterations	$\tilde{B}_1$	$\tilde{B}_2$	$\tilde{B}_3$	$\tilde{B}_4$	c_p	$T_{\rm low}$	$T_{\rm upp}$	$\boldsymbol{T}$
1	0	0	0	1	1	100	350	$\langle\{s_4\},100\rangle$
2	1	0	0	1	1	200	450	$\langle\{s_4\},100\rangle, \langle\{s_1,s_2,s_3\},100\rangle$
3	0	1	0	1	1	200	450	$\langle\{s_4\},100\rangle$, $\langle\{s_2,s_3\},100\rangle$
4	0	0	1	1	1	200	450	$\langle\{s_4\},100\rangle$, $\langle\{s_1,s_3\},100\rangle$
5	0.5	0.5	0.5	1	0	250	250	$\langle\{s_4\},100\rangle, \langle\{s_2,s_3\},50\rangle$
								$\langle\{s_1,s_3\},50\rangle, \langle\{s_1,s_2\},50\rangle$

Table 2. CG with $P_0^* = \{s_1, s_2\}$

Iterations	$\tilde{B}_1$	$\tilde{B}_2$	$\tilde{B}_3$	$\tilde{B}_4$	c_p	$T_{\rm low}$	$T_{\rm upp}$	$\boldsymbol{T}$
1	1	0	0	0	1	100	350	$\langle\{s_1,s_2\},100\rangle$
2	0	1	0	0	1	100	350	$\langle\{s_2,s_3,s_4\},100\rangle$
3	0.5	0.5	0.5	0	1	150	400	$\langle\{s_1,s_2\},50\rangle$, $\langle\{s_2,s_3,s_4\},50\rangle$, $\langle\{s_1,s_3,s_4\},50\rangle$
4	1	0	0	1	1	200	450	$\langle\{s_4\},100\rangle$, $\langle\{s_1,s_2\},100\rangle$
5	0	1	0	1	1	200	450	$\langle\{s_4\},100\rangle$, $\langle\{s_2,s_3\},100\rangle$
6	0.5	0.5	0.5	1	0	250	250	$\langle\{s_4\},100\rangle$, $\langle\{s_2,s_3\},50\rangle$
								$\langle\{s_1,s_3\},50\rangle$, $\langle\{s_1,s_2\},50\rangle$

Algorithm 1. LP-Sub-Based Rounding Algorithm
Input: Target Coverage Graph, priority vector
Output: coverage pattern p

```
Begin
    p = ∅;
    Solve LP-Sub formulation: x_i^*;
    for ∀i do
        if x_i^* ⩾ 1/ρ then
            add s_i to pattern p;
            set x̄_i = 1;
        else
            set x̄_i = 0
        end
    end
    Return p;
End
```

An approximation algorithm, Algorithm 1, gives an integer solution based on the optimal solution to the above LP-Sub formulation. For this algorithm, we have the theorem below.

Theorem 5. *The LP-Sub-based rounding algorithm is a ρ-approximation $O(m^3n^3)$ algorithm for the Sub-problem.*

Proof. According to the way we set $\overline{x}_i$, clearly, $\overline{x}_i \leqslant \rho \cdot x_i^*$. Thus,

$$\sum_{s_i \in S} \tilde{B}_i \cdot \overline{e}_i \leqslant \rho \cdot \sum_{s_i \in S} \tilde{B}_i \cdot e_i^*. \tag{31}$$

Since the solution to the relaxed linear programming problem is the lower bound of the original Sub-problem, our algorithm is a ρ-approximation if we can prove that $\overline{x}$ is also a feasible solution to the original IP formulation.

First, we divide $\overline{x}$ into two subsets:

$$S_1 = \Big\{ i | x_i^* \langle \frac{1}{\rho} \Big\} \tag{32}$$

$$S_2 = \Big\{ i | x_i^* \geqslant \frac{1}{\rho} \Big\}. \tag{33}$$

Thus,

$$\sum_{i \in S_1} x_i^* < \frac{1}{\rho} \cdot \sum_{i \in S_1} 1 \leqslant 1. \tag{34}$$

Furthermore, according to how we set $\overline{x}_i$, we can also show that

$$\sum_{i=1}^{n} \overline{x}_i \geqslant \sum_{i \in S_2} x_i^* \geqslant 1 - \sum_{i \in S_1} x_i^* > 1. \tag{35}$$

Since $\sum_{i=1}^{n} \overline{x}_i$ is an integer, it follows that $\sum_{i=1}^{n} \overline{x}_i \geqslant 1$ which means that $\overline{x}$ is also a feasible solution to the original IP formulation. Therefore, our LP-Sub-based rounding algorithm is a ρ-approximation algorithm for the Sub-problem. The time complexity of this algorithm is determined by the steps needed to solve the LP-Sub formulation, which is $O(m^3n^3)$ according to [12]. □

Hereafter, we call CG with the revised Sub-problem the revised CG. In each iteration of the revised CG, instead of using values output by LP-Sub, we verify the optimality by using the pattern generated by the rounding algorithm. These considerations lead us to the following theorem.

Theorem 6. *For the same instance of the TC problem, if CG outputs a lifetime T, the revised CG also outputs the lifetime T.*

Proof. The proof is based on the fact that only when the strict termination criteria are fulfilled does the column generation converge to the optimal solution. For both CG and the revised CG, they terminate only when the maximum reduced cost is non-positive; therefore they both have the optimal T. □

4.5 Computational Complexity Analysis

Note that for the column-generation-based approach, since in an iteration we need to solve the Sub-problem, which belongs to IP problem, the complexity remains unknown[11]. However, for the revised column-generation-based approach, in an iteration, there are only two LP formulations needed to be solved. Therefore, for the complexity of the revised column-generation-based approach, we have following conclusion.

Theorem 7. *For the revised column-generation based-approach, the computation complexity would be: $O\big(\frac{1}{4}|P|^2(|P|+1)^2 + |P|m^3n^3\big)$.*

Proof. For one iteration, since the revised CG-based approach needs to solve two LP formulations, the computational complexity depends on variables in those two formulations. Therefore, considering that in the worst case we need to visit all the patterns to determine the optimal schedule, for i-th iteration, the number of variables in the LP formulations would be

$$\begin{aligned} O &= O(1^3 + 2^3 + \ldots + |P|^3 + |P|m^3n^3) \\ &= O\Big(\sum_{i=1}^{|P|} i^3 + |P|m^3n^3\Big) \\ &= O\Big(\frac{1}{4}|P|^2(|P|+1)^2 + |P|m^3n^3\Big). \end{aligned}$$ □

4.6 From a Pattern-Based Solution to a Feasible Schedule

In this part, we will discuss the relationship between pattern-based solutions and corresponding feasible schedules. More specifically, we will show the procedure of generating a feasible schedule for the sensors based on the solution found by the CG-based approach. As stated above, after the termination of the CG-based approach, we find such a pattern-based solution. However, since that solution does not consider the ordering and frequency of patterns' presence, it does not automatically constitute a feasible schedule, defined as a time table specifying that from what time up to what time which sensor watches which targets. Therefore, we need to address the problem of transforming a pattern-based solution to a feasible schedule. To achieve the objective, we first study the interesting property of pattern-based solutions.

As shown in Fig.3, we use the solution of the example topology presented in Fig.2(b) as a pattern-based solution PBS_1. We also give another pattern-based solution PBS_2. Though PBS_1 and PBS_2 look different in the sense of ordering and frequency of patterns' presence, we know that for p_1, p_2 and p_3, they have been used in exactly 4, 2 and 4 time units for PBS_1 and PBS_2, respectively. Thus PBS_1 and PBS_2 have the same network lifetime. In other words, given a pattern-based solution, as long as sojourn time assigned to patterns remains the same, network lifetime remains the same regardless of the ordering and frequency of patterns' presence (proved in Lemma 2).

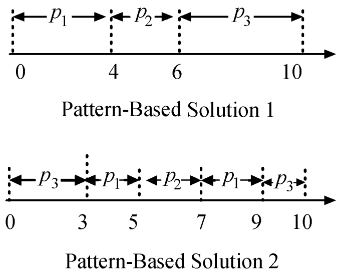


Fig.3. Pattern-based solutions.

Algorithm 2. Sensors Scheduling Algorithm
Input: Pattern-based solution, $\omega = \{T, \langle \boldsymbol{p}, \boldsymbol{t} \rangle = \{\langle p_1, t_1 \rangle, \cdots, \langle p_u, t_u \rangle\}\}$
Output: Feasible schedule
Begin
  $t = 0$;
  for $i = 1$; $i \leqslant u$; $i++$ **do**
    Activate sensors in pattern p_i in time duration $[t, t + t_i)$ while setting the other sensors to sleep;
    $t = t + t_i$;
  end
End

Therefore, we know that for a given pattern-based solution, we can find numerous different schedules by adjusting the ordering and frequency of patterns' presence while the network lifetimes remain the same. So we use a simple algorithm with time complexity $O(un)$, as presented in Algorithm 2, to generate a feasible schedule. In the next section, we use a simple example to illustrate our CG approach.

5 Numerical Results

5.1 Special Case Study

We built a simulator using Visual Studio 2005 and LINGO 9.0.

To demonstrate the algorithm, we solve the example topology proposed in [5], where $S = \{s_1, s_2, s_3, s_4\}$, $R = \{r_1, r_2, r_3\}$ and $L = \{l_{1,1}, l_{1,2}, l_{2,2}, l_{2,3}, l_{3,1}, l_{3,3}, l_{4,1}, l_{4,2}, l_{4,3}\}$. The target coverage graph is shown in Fig.1. We use two different BFSs: $P_0 = \{s_4\}$ and $P_0^* = \{s_1, s_2\}$ and set ϵ to 0.05. The details are listed in Table 1 and Table 2.

As expected, both P_0 and P_0^*, resulted in the same optimal assignment and the optimal network lifetime, where equal to 250. Moreover, it takes less than 1 second for the algorithm to solve both cases. Fig.4 shows the upper and lower bounds on lifetime calculated by our CG algorithm. For both BFSs, the upper and lower bounds quickly converge to the stable value of 250, which indicates that the optimal lifetime for this special case is 250. The result echoes to the conclusion of [5] and is further verifies our algorithm.

More specifically, let us take the instance with P_0 as an example. In the first iteration, the Master

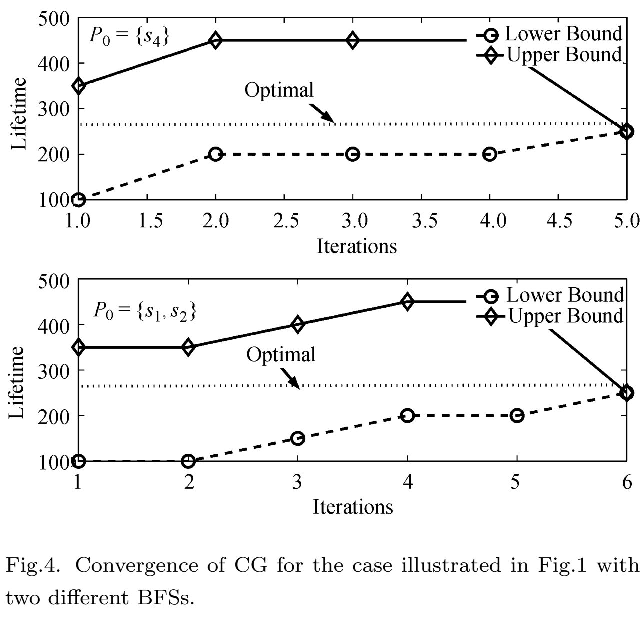


Fig.4. Convergence of CG for the case illustrated in Fig.1 with two different BFSs.

formulation gives a lifetime of 100 with optimal dual variables $\{\tilde{B}_1, \tilde{B}_2, \tilde{B}_3, \tilde{B}_4\} = \{0, 0, 0, 1\}$. By solving the Sub-problem, we know that $T_{\rm upp}$ equals 200. Therefore the algorithm selects the coverage pattern $\{s_1, s_2, s_3\}$, which responds to the maximum reduced cost according to Sub-problem. In the second iteration, the lifetime achieved by the Master formulation using updated BFS is 200 and $\{\tilde{B}_1, \tilde{B}_2, \tilde{B}_3, \tilde{B}_4\} = \{1, 0, 0, 1\}$. Still it is not optimal and a new coverage pattern $\{s_2, s_3\}$ is selected. In the third and fourth iterations, coverage pattern $\{s_1, s_3\}$ and $\{s_1, s_2\}$ are selected. In the last iteration, we have $T_{\rm low} = T_{\rm upp} = 250$, which indicates that the optimal assignment has been obtained: coverage patterns $\{s_1, s_3\}$, $\{s_1, s_2\}$, $\{s_2, s_3\}$ are active for 50 seconds and $\{s_4\}$ is active for 100 seconds and the optimal lifetime equals 250. This assignment echoes with the results in [5].

5.2 Performance Comparison

In this part, in order to verify the efficacy of the proposed approach, we compare the running performance of the following three different approaches.

1) Directly solving the MINLP formulation by the LINGO software[18] (for short MINLP).

2) The Column-Generation-based approach (for short CG).

3) The Column-Generation-based approach with the revised Sub-problem (for short CGR).

We used eight test networks of various sizes. The number of the nodes and targets range from 10 to 50, and from 5 to 10, respectively. For each of the test network, the following computations have been conducted. First, we solve the MINLP formulation directly by using the LINGO software. Considering that for large networks, MINLP will take excessive computational time. We therefore set a time limit of 5 hours. Then, we apply the proposed CG-based approach as well as the revised-CG-based approach to solve these test networks. For these two approaches, ϵ is set to 0.05.

Simulation results are concluded in Table 3, from which we can obtain several observations as follows.

1) For small-scale networks, MINLP may achieve the optimal solutions. But when the scale of problems becomes larger, this approach becomes unbearably slow.

2) Both CG-approaches can efficiently solve the TC instances for small-scale networks. But they are clearly not computationally efficient for large networks.

3) For CGR, it outperforms the CG in terms of computational complexity and the number of iterations needed.

For the first observation, the reason is quite straightforward. Actually, directly solving the MINLP formulation implies a brutal search, which becomes very inefficient or even helpless on an NP-complete problem due to the explosive search space. For example, though MINLP can solve the instance with 25 sensors and 5 targets in 487 seconds, when the scale of the network grows, it fails to solve any instance with more than 30 nodes within the pre-set time limit, i.e., 5 hours. Clearly, due to its unacceptable computational complexity, MINLP is a very bad candidate algorithm for the TC problem, even for a network in very small scale.

For the second observation, we notice that both CG approaches can be efficient algorithms for a small or moderate scale network. For a same-scale network, e.g., $n = 20$ and $m = 5$, the optimal solution can be achieved within 46 seconds. Because in each iteration CG approaches greedily choose the pattern with the largest payoff. However, when n increases from 20 to 50, both CG approaches need nearly 1 hour to solve the instance. This is because the TC problem is inherently difficulty due to its NP-complete property. The phenomenon indicates that heuristic approaches like LP-MSC and Greedy-MSC in [6] are of some value. Because in a real-world system, heuristic approaches can efficiently adapt themselves to large networks.

For the third observation, as shown in Fig.5, we notice that CGR needs less iterations to reach the termination criterion compared to the CG approach for the same instance. For example, when we set $n = 40$ and $m = 10$, CGR converges after 86 iterations while CG needs 109 iterations. Moreover, for the same instance,

Table 3. Performance Comparison

n	m	MINLP (LINGO)		CG			Revised-CG		
		Lifetime	Time	Lifetime	Iterations	Time	Lifetime	Iterations	Time
10	5	26.78	15 s	33.33	3	3 s	33.33	3	2 s
20	5	38.73	219 s	50.00	11	46 s	50.00	14	31 s
25	5	87.12	487 s	100.00	35	4 m 25 s	100.00	31	98 s
30	5	–	5 h	143.67	47	15 m 43 s	143.67	54	10 m 17 s
40	5	–	5 h	200.00	97	33m18 s	200.00	89	21 m 56 s
40	10	–	5 h	150.00	109	42 m 11 s	150.00	86	28 m 32 s
50	5	–	5 h	250.00	176	56 m 24 s	250.00	152	37 m 19 s
50	10	–	5 h	250.00	183	1 h 15 m 23 s	250.00	162	45 m 45 s

it takes longer time to solve an IP formulation (CG) than an LP formulation (CGR)[12]. Actually, an LP formulation can be always solved in polynomial time using certain approaches, e.g., the algorithm proposed in [13]. Considering that the CG approach iteratively solves master and slave problems, CGR always deals with two LP formulations while the original CG approach always needs to solve an IP formulation, i.e., the Sub-problem. Therefore, as shown in Fig.6, the revised CG approach outperforms the original CG approach.

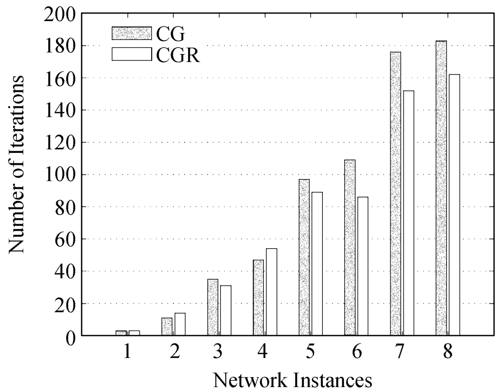


Fig.5. Iteration comparison between CG and CGR.

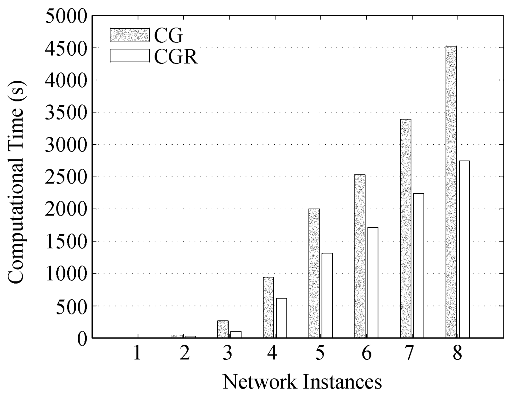


Fig.6. Computational time comparison between CG and CGR.

6 Related Work

Previous literatures about this topic mostly focus on a full target coverage problem, where each target is covered by at least one sensor at any time[3-10].

In [4], assuming a fixed sensing range for all the sensors and an identical coverage requirement for all the targets, a heuristic method is proposed to extend the network lifetime by organizing sensors into a maximal number of disjoint set covers that are activated successively. The authors further relaxed the stringent constraint of disjoint set by allowing sensors to participate in multiple sets, and designed two heuristics (one based on linear programming and one greedy) to compute the Maximum Set Cover (MSC) problem[5]. It is still an open problem to decide cardinality of set covers, which is simply set to a fixed value in [5]. The authors also investigated the case where the sensing range for sensors is adjustable and a similar iterative method is developed in [6]. Note that in above literatures only heuristic methods have been proposed and no theoretical result, e.g., an lifetime upperbound, has been introduced.

A similar coverage scenario is proposed in [7]. The objective is to maximize the network lifetime for k to 1 sensor-target surveillance networks. This work has been further extended in [8] to accommodate the same k to 1 sensor-target problem with an extra routing requirement. They made an assumption: a sensor can watch only one target at a time. Note that this assumption significantly reduces the complexity in solving such a kind of target coverage problems. This assumption however may not be suitable for some applications like the multiple targets tracking system[17] or SensorWeb project[18], where heavy load should be distributed to limited number of sensor nodes and a sensor node needs to be in charge of different targets simultaneously.

There are also several articles addressing the coverage breach (coverage breach occurs when a subset of sensors fails to cover all targets) problem, which is defined as how to minimum coverage breach while making efficient use of both energy and bandwidth. The problem is somehow similar to PTC problem since both of them consider the case when some targets lose coverage. But in PTC problem, every target has a coverage requirement, which is absent in the coverage breach problem. Also note that this coverage breach problem belongs to NP-complete and only a set of heuristic algorithms have been proposed[19-21].

Clearly, previous literatures either focus on heuristic algorithms, which remain difficult to characterize and have no performance guarantee[4-6,9-10], or employ a strict constraint to reduce complexity[7-8], which cannot be extended to accommodate the general case. Therefore no theoretical result has been reported yet.

In our recent work[15], we proposed a column-generation-based approach to optimally solve the target coverage problem. However, this approach suffers from shortcomings like high computational complexity.

7 Conclusion

In this paper, we addressed the problem of achieving an optimal network lifetime in surveillance sensor networks, which is an NP-complete problem. First, we formulated the optimization problem in a non-linear programming form. Directly solving this optimization problem would be extremely complicated because of its combinatorial complexity. Thus, we developed a novel column-generation-based approach that decomposes the original formulation into sub-formulations and solves them iteratively. The proposed algorithm can guarantee the network lifetime is at least $(1 - \epsilon)$ of the optimal network lifetime, where ϵ can be made arbitrarily small depending on the required precision.

The proposed CG-based approach can achieve nearly optimal or even optimal solutions and can serve as benchmark for other algorithms on this problem when the scale of networks is small. However, as shown in the simulations, the computational time grows much fast as the scale of problems increases. Thus polynomial-time heuristic or greedy algorithms are much more desired, especially those with performance guarantees.

References

[1] He T, Krishnamurthy S, Luo L *et al.* VigilNet: An integrated sensor network system for energy-efficient surveillance. *ACM Transactions on Sensor Networks*, 2006, 2(1): 1-38.

[2] Akyildiz I F, Su W, Sankarasubramaniam Y, Cayirci E. A survey on sensor networks. *IEEE Communications Magazine*, 2002, 40(8): 102-114.

[3] Cardei M, Du J. Energy-efficient coverage problems in wireless ad hoc sensor networks. *The Computer Communications Journal*, 2006, 29(4): 413-420.

[4] Cardei M, Du D Z. Improving wireless sensor network lifetime through power aware organization. *ACM Wireless Networks*, 2005, 11(3): 333-340.

[5] Cardei M, Thai M, Li Y, Wu W. Energy-efficient target coverage in wireless sensor networks. *In Proc. IEEE INFOCOM 2005*, Miami, USA, Mar. 13-15, 2005, pp.1976-1984.

[6] Cardei M, Wu J, Lu M, Pervaiz M O. Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. In *Proc. IEEE Int. Conf. Wireless and Mobile Computing, Networking and Communications* (*WiMob 2005*), Montreal, Canada, Aug. 22-24, 2005, pp.438-445.

[7] Liu H, Wan P, Jia X. Maximal lifetime scheduling for k to 1 sensor-target surveillance networks. *Computer Networks*, 2006, 50(15): 2839-2854.

[8] Liu H, Wan P, Jia X. Maximal lifetime scheduling for sensor surveillance systems with k sensors to 1 target. *IEEE Transactions on Parallel and Distributed Systems*, 2007, 17(12): 1526-1536.

[9] Lu M, Wu J, Cardei M, Li M. Energy-efficient connected coverage of discrete targets in wireless sensor networks. In *Proc. IEEE International Conference on Computer Networks and Mobile Computing* (*ICCNMC 2005*), Zhangjiajie, China, Aug. 2-4, 2005, pp.43-52.

[10] Jaggi N, Abouzeid A A. Energy-efficient connected coverage in wireless sensor networks. In *Proc. 4th Asian International Mobile Computing Conference* (*AMOC*), Kolkota, India, Jan., 2006.

[11] Horst R, Tuy H. Global Optimization: Deterministic Approaches. 3rd Ed. Berlin: Springer-Verlag, 1996.

[12] Ye Y. An $O(n^3 l)$ potential reduction algorithm for linear programming. *Mathematical Programming*, 1991, 50: 239-258.

[13] Vance H, Barnhart C, Johnson E, Glaw A. Solving binary cutting stock problems by column generation and branch-and-bound. *Computational Optimization and applications*, 1994, 3(2): 111-130.

[14] Bjorklund P, Varbrand P, Yuan D. Resource optimization of spatial TDMA in ad hoc radio networks: A column generation approach. In *Proc. IEEE INFOCOM 2003*, San Francisco, USA, Mar. 30-Apr. 3, 2003, pp.818-824.

[15] Gu Y, Ji Y S, Li J, Zhao B H. QoS-aware target coverage in wireless sensor networks. *Wireless Communications and Mobile Computing*, 2009, 9(12): 1645-1659.

[16] Lubbecke M E, Desrosiers J. Selected topics in column generation. *Operations Research*, 2002, 53(6): 1007-1023.

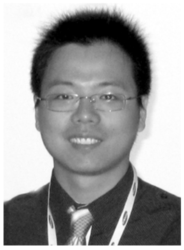
[17] He T, Vicaire P A, Yan T *et al.* Achieving real-time target tracking using wireless sensor networks. In *Proc. IEEE RTAS 2006*, San Jose, California, Apr. 4-7, 2006, pp.37-48.

[18] Nath S, Liu J, Zhao F. SensorMap for wide-area sensor webs. *IEEE Computer Magazine*, 2007, 40(7): 90-93.

[19] Maggie X C, Ruan L, Wu W L. Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks. In *Proc. IEEE INFOCOM 2005*, Miami, USA, Mar. 13-17, 2003, pp.2638-2645.

[20] Maggie X C, Ruan L, Wu W L. Coverage beach problems in bandwidth constrained sensor networks. *ACM Transactions on Sensor Networks*, 2007, 3(2): Article No.12.

[21] Wang C, Thai M T, Li Y *et al.* Minimum coverage breach and maximum network lifetime in wireless sensor networks. In *Proc. IEEE Global Telecommunications Conference* (*GLOBECOM*), Washington DC, USA, Nov. 26-30, 2007, pp.1118-1123.



Yu Gu received his Ph.D. degree from the Department of Computer Science, University of Science and Technology of China, in 2010. He received his B.Eng. degree in 2004, from the Special Classes for the Gifted Young, University of Science and Technology of China. From Feb. 2006 to Aug. 2006, he has been an visiting student in Wireless Network Group, Microsoft Research Asia, Beijing, China. From Dec. 2007 to Dec. 2008, he has been with the Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba, Japan, where he was a visiting scholar. Now he is working in the National Institute of Informatics, Japan, as a JSPS research fellow. He received the Excellent Paper Award in the IEEE Scalcom2009. His research interests include network protocol verification, Ad Hoc network, especially wireless sensor network.

Bao-Hua Zhao graduated from Chengdu Institute of Telecommunication Engineering (renamed as University of Electronic Science and Technology of China now) in 1969, and in 1976 he began to teach in the Department of Computer Science of University of Science and Technology of China. He is a professor and Ph.D. supervisor of the University of Science and Technology of China. He currently is vice-director of Software Research Center of the Department of Computer Science, vice chairperson of CIC Communication Software Technical Committee, member of Professional Committee of Software Engineering of Chinese Computer Federation, and editor of Journal of Computer Information and Technology. His research interests include protocols theory and engineering and algorithms of wireless sensor network.

Yu-Sheng Ji received the B.E., M.E., and Ph.D. degrees in electrical engineering from the University of Tokyo in 1984, 1986, and 1989, respectively. She joined the National Center for Science Information Systems, Japan (NACSIS) in 1990. Currently, she is an associate professor at the National Institute of Informatics, Japan (NII), and the Graduate University for Advanced Studies (SOKENDAI). Her research interests include network architecture, traffic control, and performance analysis for quality of service provisioning in communication networks. She is on the board of trustees of IEICE since 2006, and an associate editor of IEICE Transactions since 2007.

Jie Li received his B.E. degree in computer science from Zhejiang University, China, M.E. degree in electronic engineering and communication systems from Chinese Academy of Posts and Telecommunications, China. He received his Dr. Eng. degree from the University of Electro-Communications, Tokyo, Japan. He has been with the Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba, Japan and is a full professor. His research interests are in mobile distributed multimedia computing and networking, OS, network security, modeling and performance evaluation of information systems. He is a senior member of IEEE, and a senior member of ACM. He has served as a secretary for Study Group on System Evaluation of the Information Processing Society of Japan (IPSJ) and on the editorial boards for IPSJ (Information Processing Society of Japan) Journal, and International Journal of High Performance Computing and Networking. He is also serving on Steering Committees of the SIG of System EVAluation (EVA) of IPSJ, the SIG of DataBase System (DBS) of IPSJ, and the SIG of MoBiLe Computing and Ubiquitous Communications of IPSJ. He has also served on the program committees for several international conferences such as IEEE INFOCOM, IEEE GLOBECOM, and IEEE MASS.