Target Coverage Problem in Wireless Sensor Networks: A Column Generation Based Approach

Yu Gu*†, Jie Li‡, Baohua Zhao*†, and Yusheng Ji§

* Dept. of Computer Science and Technology, Univ. of Science and Technology of China, Hefei, Anhui, 230027, China

† State Key Laboratory of Networking and Switching Technology, Beijing, 100876, China

‡ Dept. of Computer Science, Univ. of Tsukuba, Tsukuba Science City, Ibaraki 305-8573, Japan

§ Information Systems Architecture Science Research Division, National Institute of Informatics, Tokyo, Japan

***Abstract*—Target coverage problem in wireless sensor networks remains a challenge. Due to nonlinear nature, previous work has mainly focused on heuristic algorithms, which remain difficult to characterize and have no performance guarantee. To solve the problem, this paper offers two important contributions.**

The first contribution is to have two lifetime upper bounds, which could be used to justify performance of previously proposed heuristic algorithms. One upper bound is based on the relaxation and reformulation technique while the other is derived by relaxing coverage constraints. We study the interesting connection between those two bounds and thus endow them with physical meanings.

The second contribution is proposing a column generation based (CG) approach. The objective is to find an optimal schedule, defined as a time table specifying from what time up to what time which sensor watches which targets while the maximum lifetime has been obtained. We also offer an in-depth theoretic analysis as well as several novel techniques to further optimize the approach. Numerical results not only demonstrate that the lifetime upper bounds are very tight, but also verify that the proposed CG based approach constantly yields the optimal or near optimal solution.

I. Introduction

Wireless sensor networks [1] have emerged recently as an effective way of monitoring remote or inhospitable physical targets [2], which usually have different Quality of Service (QoS) requirements, i.e., different targets may need different sensing quality in terms of the number of transducers, sampling rate, etc. Considering that sensors are typically battery-driven and have energy constraints [3], the problem of optimizing network lifetime while fulfilling those QoS requirements rises as an interesting but challenging question. Numerous methods have already been proposed to address a set of target coverage problems with different network settings [4] [5] [6] [7] [8] [9] [10] [11].

The target coverage problem has been proposed in [5] [6] (also named the Maximum Set Cover (MSC) problem), where each target needs one sensor at any time instant and the objective is to maximize network lifetime under energy constraints. The authors first proved that MSC belongs to NP-complete, and then developed a key idea, based on which MSC problem has been mathematically formulated. The idea is that the continuous time could be divided into discrete time slots with different lengths. In each slot, only one coverage pattern, defined as a subset of sensors that can cover all targets, is activated while setting remaining sensors into sleep state to save energy. However, there are two weak points about this paper. The first one is that the idea of discretization of continuous time has not been justified and a formal proof is missing. The other one, also the critical one, is that their formulation belongs to a Mixed Integer Nonlinear Programming (*MINLP*) and is very difficult to be tackled [12], especially in a resource limited sensor platform. Actually, authors failed to solve it directly and proposed two heuristic algorithms instead, whose solutions are suboptimal and likely to be far away from the optimum in certain cases. Moreover, the performance of their heuristic algorithms remains unknown due to lack of comparisons, e.g. comparisons with lifetime upper bound or the exact optimal solution.

This problem has been further extended in [10] with the same objective of maximizing the network lifetime in a so called k to 1 sensor-target surveillance networks, where each target should be watched by at least k sensors at any time instant. The critical difference between this problem and MSC problem is an additional assumption, that is, a sensor can watch only one target at a time. This assumption significantly reduces problem complexity. Actually, based on this assumption, authors are able to develop a polynomial-time algorithm, which guarantees an optimal solution. However, since the optimal algorithm is directly based on the assumption, it cannot be extended to solve the target coverage problems without this assumption. Moreover, this assumption may not be suitable for some applications like the multiple targets tracking system [13] or SensorWeb project [14], where data load should be distributed to a limited number of sensor nodes and each sensor node needs to be in charge of different targets simultaneously.

Therefore, previous literatures on target coverage problems either focus on heuristic algorithms, which remain difficult to characterize and have no performance guarantee [5] [6] [7] [8] [9], or relax the problem with an additional assumption to reduce complexity [10] [11], which cannot be extended to address problems without the assumption. To the best of our knowledge, no theoretical result has been reported yet. To fill in the blank, this paper aims to design a general optimization architecture that can be applied to a set of similar target coverage problems. The architecture we develop in this paper contains two modules. The first module employs a general idea of relaxations to build Linear Programming formulations,

978-1-4244-5113-5/09/$25.00

by solving which we could obtain a lifetime upper bound. The second module comprises an efficient Column Generation (CG) based algorithm to achieve solutions with a performance guarantee.

Specifically, we investigate an extension of the MSC problem by taking QoS requirements for targets into consideration and name this new problem as a Target $\mathcal{Q}$-Coverage problem (T$\mathcal{Q}$C) in this paper. For T$\mathcal{Q}$C, first we define a subset of sensors which meets target coverage constraints as a *coverage pattern*, in order to capture relationship between targets and sensors. If only one such *coverage pattern* is used, the network is down when the first node in this pattern dies. However, due to spatial redundancy of sensors, lifetime can be extended by switching among different patterns. Then, based on this definition, we formulate the problem as a Mixed Integer Nonlinear Programming (*MINLP*) problem. Clearly, directly solving this MINLP formulation is prohibitively hard due to inherent complexity. Therefore, we propose an optimization architecture containing two modules to address this issue.

The first module provides two lifetime upper bounds, which could be used as a comparison with heuristic algorithms to justify the overall performance. One is based on the relaxation and reformulation technique on MINLP formulation while the other one is derived via a linear programming (LP) formulation, by relaxing coverage constraints, e.g. from that each target should be covered by at least $K(K > 0)$ sensors at any given moment, to that each target should be covered by at least K sensors on average during network lifetime, in order to establish an objective we can compare with. We also investigate the connection between two bounds and thus provide physical meanings for them. Note that the relaxation techniques we use to develop those two bounds are general and could also to be applied to other similar target coverage problems like MSC. Therefore, the bounds are of great meaning in a way that they could be used to justify the efficiency of heuristic algorithms developed for other target coverage problems besides T$\mathcal{Q}$C.

The second module contains a column generation (CG) [15] based approach, which seeks an optimal feasible schedule, defined as a time table specifying from what time up to what time which sensor watches which targets. The approach decomposes the MINLP formulation into a mater problem as well as a sub problem and solve them iteratively. Here a column corresponds to a feasible coverage pattern and our idea is to find a column with steepest ascent in lifetime, based on which we iteratively search for the maximum lifetime solution. Since CG needs an initial set of feasible coverage patterns to start, we design a random selection algorithm, which helps to speed up the converge procedure of our approach. Furthermore, we note that the sub problem is an integer programming (IP) problem and maybe difficult to be tackled 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 employing a linear relaxation technique coupled with a rounding algorithm, which significantly reduces computation complexity while maintaining certain level of performance.

To verify the efficiency of our proposed methods, extensive experiments have been conducted. By comparing with the lifetime upper bound we develop, we show that the CG based approach constantly yields optimal or near-optimal solutions and thus provides a critical performance benchmark when evaluating other heuristic algorithms for target coverage problems in WSN.

Furthermore, in order to show that the proposed optimization architecture could be used for a set of similar target coverage problems, we optimally solve an important sample topology presented in Reference [6], where it has been used as an basis to demonstrate the definition of target coverage problem as well as the idea to solve the problem. The numerical results confirmed our conclusion.

In our previous paper on this topic [16], we also considered how to optimize network lifetime under QoS coverage constraints. We proposed a lifetime upper bound based on relaxing coverage constraints and a column generation based approach. To identify the original contributions of this paper, we list three major improvements over our previous paper [16] as following:

1) We propose a new lifetime upper bound via the reformulation and relaxation technique, connect it with the upper bound presented in [16] and endow them with physical meanings.
2) By relaxing the sub problem, we revise the CG based approach in [16] in order to reduce computational complexity. Theoretical analysis shows that, while the relaxed sub problem can be solved much more efficiently, performance still can be guaranteed. Furthermore, the revised CG based approach has been analyzed.
3) We conduct numerous experiments to justify performance of the revised CG based approach. Numerical results confirm its superiority over the CG based approach presented in [16].

The rest of the paper is organized as follows. In the next section, we briefly review some related works. We describe our system model and formulate the problem in Section II. we develop two lifetime upper bounds in Section III. In Section IV, a column generation based approach is proposed. Numerical simulation results are reported in Section V. Finally, Section VI concludes the paper.

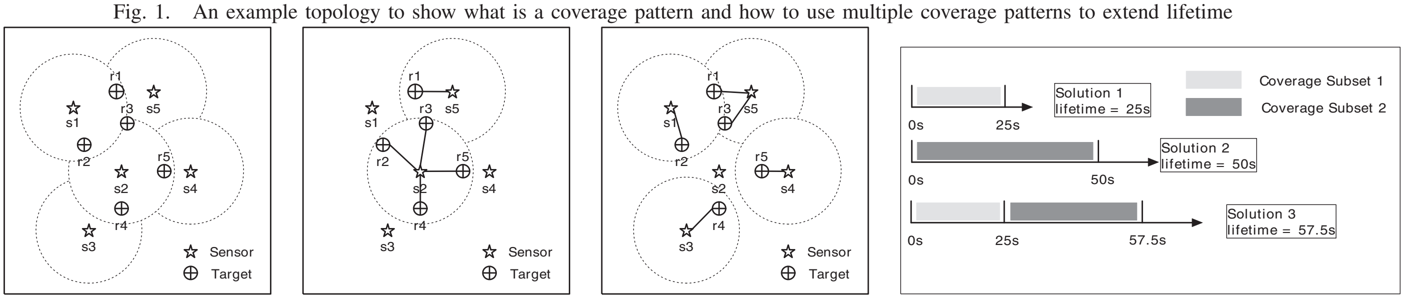
II. Problem Statement

A. System Model

We consider a set of n sensor nodes: $S = \{s_1, s_2 \ldots s_n\}$ deployed (randomly or deterministically) to cover m targets: $R = \{r_1, r_2 \ldots r_m\}$ with a priority vector $Q = \{q_1, q_2 \ldots q_m\}$. The priority vector denotes that at any given moment, target r_k is covered by at least q_k ($q_k \geq 1$) active sensor nodes. For sensor node s_i, it has an initial energy e_i ($i = 1 \cdots n$). First, in order to illustrate the coverage relationship between sensor nodes and targets, we define a notation named Target Coverage Graph as following,

Definition 1: *Target Coverage Graph*. A target coverage graph is a bipartite graph $TCG = \{S, L, R\}$ where there exists a link

Fig. 1. An example topology to show what is a coverage pattern and how to use multiple coverage patterns to extend lifetime



(a) Network Topology (b) Coverage pattern 1 (c) Coverage pattern 2 (d) Using Multiple Coverage patterns to Extend Network Lifetime

$l_{i,k} \in L$ 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.

The target $\mathcal{Q}$-Coverage requirement is defined as follows:

Definition 2: *Target $\mathcal{Q}$-Coverage Requirement*. At any given moment target r_k is covered by at least $q_k(q_k \geq 1)$ sensor nodes $(k = 1 \ldots m)$.

For energy model, we use the same model presented in our previous paper [16]. Without loss of generality, we assume that the energy that one sensor spends on covering one target per time unit is 1. According to the above analysis, if one sensor is assigned to cover K targets simultaneously, the energy cost will be $K * 1 = K$.

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 less than q_k alive sensors can be assigned.

Accordingly, the problem of our concern can be formally stated as:

Problem Statement: *Target $\mathcal{Q}$-Coverage (TQC) Problem* . Given a target coverage graph $TCG = \{S, L, R\}$ with a priority vector Q, maximize the network lifetime.

In this next section, we will give an upper bound and then formally formulate the problem.

B. Mathematical Formulation

We identify a subset of sensor nodes such that they meet target coverage constraint as a *coverage pattern*. Due to spatial redundancy of sensors, lifetime can be maximized by switching among different coverage patterns subject to energy constraints.

Here we use an example to illustrate the concept of the coverage pattern and how to use multiple coverage patterns to prolong network lifetime. As shown in Figure 1(a), there are five sensors deployed to cover five targets with a priority vector $Q = \{1, 1, 1, 1, 1\}$. The initial energy is set to 100 units for all five sensors and covering one target for one second costs a sensor one unit energy. First we select s_2, s_5 as the coverage pattern 1 (Figure 1(b)), where s_2 watches r_2,r_3, r_4 and $r5$ while s_5 covers r_1. Clearly, if we are only allowed to use coverage pattern 1, s_2 will die after 25 seconds and targets r_2,r_3, r_4 and $r5$ will lose coverage. Therefore the network lifetime will be 25 seconds. Similarly, we can select s_1, s_3, s_4, s_5 as the coverage pattern 2 (Figure 1(c)) where s_1 covers r_2, s_3 covers r_4, s_4 covers r_5, and s_5 covers r_1 and r_3,, respectively. If only coverage pattern 2 is used, after 50 seconds, s_5 will die and targets r_1 and r_3 lose coverage. Therefore the network lifetime will be 50 seconds.

However, as shown in Figure 1(d), we can use both coverage patterns alternatively to extend network lifetime as following: From 0 second to 25 second, we use coverage pattern 1. s_2 will die and s_5 still have 75 units energy left. Since coverage pattern 2 does not include s_2, we can use it from 25 second to 57.5 second. Then s_5 runs out of energy and r_1, r_3 lose coverage. Therefore, we could extend lifetime to 57.5 seconds if these two coverage patterns are used one by one. Actually, we could still prolong network lifetime after s_5 is dead since r_1 and r_3 can be monitored by s_1, which still have 62.5 units energy left after 57.5 second. Therefore, we show that the network lifetime can be maximized by scheduling among coverage patterns.

Let $x_{i,k}^p$ indicate if or not sensor node s_i covers target r_k in coverage pattern p. t_p denotes the time duration assigned to a pattern p, and P is the set containing all patterns. We define $U_k = \{s_i \mid s_i \text{ can cover } r_k\}$ and $V_i = \{r_k \mid r_k \text{ can be covered by } s_i\}$. We can mathematical formulate the problem as following,

$$\text{(MINLP)} \qquad Max(\sum_{p \in P} t_p) \tag{1}$$

subject to

$$\sum_{s_i \in U_k} x_{i,k}^p \geq q_k \quad (\forall r_k \in R, p \in P) \tag{2}$$

$$e_i^p = \sum_{r_k \in V_i} x_{i,k}^p \quad (\forall s_i \in S, p \in P) \tag{3}$$

$$\sum_{p \in P} e_i^p \cdot t_p \leq e_i \quad (\forall s_i \in S, p \in P) \tag{4}$$

and

$$x_{i,k}^p = \{0, 1\}, t_p \geq 0 \quad (\forall s_i \in S, p \in P) \tag{5}$$

Eq.(1) is referred to as *MINLP* optimization problem hereafter.

Eq.(2) ensures that for every target $r_k \in R$, there would be at least q_k sensors covering it at any given moment.

Eq.(3) denotes the amount of energy s_i consumes in pattern p.

Eq.(4) guarantees that for every sensor node $s_i \in S$, its total energy consumption would not exceed the initial energy e_i.

There are two major problems about this formulation, namely,

1) P, which contains all coverage patterns, is unknown. Furthermore, its cardinality might be exponential to number of links between sensors and targets.
2) This formulation belongs a non-linear programming problem and is very difficult to solve [12].

Nevertheless, directly solving this *MINLP* formulation is quite difficult. The same difficulty happened in Reference [6] and authors failed to address it. Instead, they proposed two heuristic algorithms (LP-MSC, Greedy-MSC), whose performance has not been justified.

In the next two sections, we will propose two theoretic results, whose key ideas could be applied to a set of similar target coverage problems including MSC and TQC. The first one is lifetime upper bound, which can be used to justify heuristic approaches proposed for the problem. The other one is a column generation based approach, which could optimally solve the problem. Note that though this approach can guarantee the optimal solution, it is still an exponential time algorithm. Therefore heuristic algorithms may be still needed.

III. The First Theoretic Result: Lifetime Upper Bound

In this section, we try to obtain the first theoretic result for the target coverage problem: lifetime upper bound. We develop the upper bound through two different ways. One is achieved by relaxing *MINLP* formulation, which is also used in some related works, e.g. [5] [6]. However, due to lack of physical meaning, it is not clear whether or not this upper bound is suitable. Thus, we develop another lifetime upper bound by relaxing coverage constraints, e.g. from that every target should be covered by at least q_k sensors at any given moment, to that every target should be covered by at least q_k sensors on average. Clearly, this upper bound has an explicit physical meaning and is easy to understand. It is quite interesting to investigate whether there exists a relationship between those two upper bounds. By utilizing another transformation technique, we show that the first upper bound achieved by relaxing *MINLP* shares exactly the same form with the second upper bound obtained by relaxing coverage constraints. This investigation establishes the connection between two upper bounds and thus offers an explicit physical meaning for the first upper bound. In other words, those two upper bounds actually describe the same idea in different ways.

A. Lifetime Upper Bound By Relaxing MINLP

In this part, we will derive the first lifetime upper bound by relaxing *MINLP*. First, we show that by adopting a reformulation and relaxation technique, we can convert *MINLP* to a linear programming (*LP*) formulation, whose optimal solution servers as a lifetime upper bound.

Specifically, we define $z^p_{i,k} = x^p_{i,k} * t_p$, thus we have:

$$\text{(IP)} \qquad Max(\sum_{p \in P} t_p) \tag{6}$$

subject to

$$\sum_{s_i \in U_k} z^p_{i,k} \geq q_k \cdot t_p \quad (\forall r_k \in R, p = 1, \ldots, n) \tag{7}$$

$$\sum_{p \in P} \sum_{r_k \in V_k} z^p_{i,k} \leq e_i \quad (\forall s_i \in S, p = 1, \ldots, n) \tag{8}$$

and

$$z^p_{i,k} = \{0, t_p\}, t_p \geq 0 \quad (\forall s_i \in S, p = 1, \ldots, n) \tag{9}$$

Considering that the above formulation belongs to an Integer Programming (IP) problem and is still hard to address according to Reference [12], we further relax Eqn.9 and obtain a linear formulation as following,

$$\text{(LP)} \qquad Max(\sum_{p \in P} t_p) \tag{10}$$

subject to Eqn.7, Eqn.8 and

$$0 \leq z^p_{i,k} \leq t_p, t_p \geq 0 \quad (\forall s_i \in S, p = 1, \ldots, n) \tag{11}$$

Clearly, according to Reference [12], we have following conclusion for the above formulation,

Theorem 1. *For a given instance of TQC problem, the optimal solution obtained by* LP *formulation is an upper bound.*

However, since the transformation technique used has no physical meaning, it is not clear whether or not the upper bound is appropriate. Therefore, in the next part, we will develop another lifetime upper bound by explicitly relaxing coverage constraints.

B. Lifetime Upper Bound By Relaxing Coverage Constraints

In this part, we develop a lifetime upper bound by relaxing the coverage constraints, that every target should be covered by at least q_k sensors at any given moment, to that every target should be covered by at least q_k sensors on average.

Define T, $y_{i,k}$ as the network lifetime and the total time duration sensor s_i spends on covering target r_k, respectively.

$$\text{(UPP)} \qquad Max(T) \tag{12}$$

subject to

$$\sum_{s_i \in U_k} y_{i,k} \geq q_k \cdot T, \quad \forall r_k \in R \tag{13}$$

$$\sum_{r_k \in V_i} y_{i,k} \leq e_i, \quad \forall s_i \in S \tag{14}$$

$$0 \leq y_{i,k} \leq T, \forall s_i \in S, r_k \in R \tag{15}$$

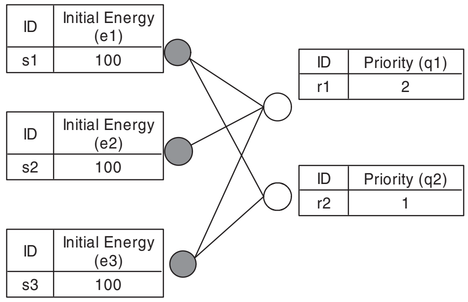


Fig. 2. An example topology

TABLE I
OPTIMAL ASSIGNMENT FROM THE *UPP* FORMULATION

Sensor	Time for r_1 (s)	Time for r_2 (s)
s_1	0.00	100.00
s_2	100.00	0.00
s_3	100.00	0.00
Opt.	T = 100.00	

Eqn.13 ensures that for every target $r_k \in R$, there would be at least q_k sensors covering it on average.

Eqn.14 guarantees that for every sensor node $s_i \in S$, its total energy consumption would not exceed the initial energy e_i.

Eqn.15 makes sure that for each $s_i \in U_k$, its time covering some target $r_k \in R$ would not exceed network lifetime, e.g. T.

The *UPP* is a typical linear programming formulation, thus can be solved in polynomial time. Actually, according to the algorithm proposed by Ye *et al.* [17], it could be solved in $O(m^3n^3)$ time. For *UPP*, we have the following conclusion,

Theorem 2. *For a given instance of TQC problem, the optimal solution obtained by* UPP *formulation is an upper bound.*

Proof. For a given instance of T$\mathcal{Q}C$ problem, for any assignment of sensors that can support the coverage constraints, it holds for all constraints in *UPP* formulation according to the way we build it. Therefore, it is within the solution space of *UPP* formulation. In other words, the actual optimal solution for the given instance also must be a solution of *UPP* formulation. On the other hand, the optimal solution by solving *UPP* formulation is an upper bound for all feasible solutions in the solution space (note that the solution space for *UPP* formulation may also includes infeasible solutions for the given instance). Therefore, the optimal solution must be an upper bound. $\diamond End \diamond$

As shown in Tab. I, *UPP* solves the instance posed in Fig.2 and outputs an optimal T which equals to 100 within 0.1 second using Lingo 9.0 [18]. Clearly, *UPP* is tight for this special instance.

However, since the above formulation only indicates the total time a sensor should cover some target, if we want to achieve the optimal T, we should find a schedule, defined as a time table specifying from what time up to what time which coverage pattern should be used, to satisfy each $y_{i,k}$. This schedule however may not exist or need much extra effort to find. Liu *et al.* in [11] proposed an efficient algorithm to construct such a schedule with an extra constraint that a sensor can watch only one target at a time. In our model, a sensor can be used to cover several targets at a time and a target may need several sensors at a time, thus their algorithm can not be extended to address our problem.

Though *UPP* cannot establish an exact schedule for the T$\mathcal{Q}$C problem, it serves as an objective and demonstrates the performance gap between the optimal solution and the one obtained by proposed approaches.

C. Theoretic Analysis of Upper Bounds

As claimed in the previous context, due to lack of physical meaning of *LP*, we cannot identify whether or not it is tight or suitable for T$\mathcal{Q}C$ problem. Therefore, it is quite interesting to study the relationship between those two upper bounds. Actually, after analysis, we find out that there exists an important connection between them. In order to demonstrate the connection, we further transform the *LP* into another form, names as *UPP-MINLP*, which seems to be exactly the same as *UPP* formulation except for symbols used , e.g. variables used in the formulation. Therefore, *UPP-MINLP* indeed has physical meanings in a way that the *UPP* formulation has been built.

By defining $T^* = \sum_{p \in P} t_p$, $y^*_{i,k} = \sum_{p \in P} \sum_{s_i \in U_k} z^p_{i,k}$ and summing t_p for Eqn.13 and Eqn.15, we could change *LP* formulation into a similar form (named *UPP-MINLP*) with *UPP* as following,

$$\text{(UPP-MINLP)} \qquad Max(T^*) \tag{16}$$

subject to

$$\sum_{s_i \in U_k} y^*_{i,k} \geq q_k \cdot T^*, \quad \forall r_k \in R \tag{17}$$

$$\sum_{r_k \in V_i} y^*_{i,k} \leq e_i, \quad \forall s_i \in S \tag{18}$$

$$0 \leq y^*_{i,k} \leq T^*, \forall s_i \in S, r_k \in R \tag{19}$$

Except for names of variables used in the formulations, *UPP* and *UPP-MINLP* are exactly the same. This establishes a novel contribution: we give the *UPP-MINLP* a physical meaning and background, which makes it much sounder and stronger. In Reference [5] [6], similar reformulation and relaxation techniques have been used to deal with their formulations, however, due to lack of physical meanings, it is unclear that whether this kind of relaxed formulations is suitable or not. Therefore, the theoretic analysis in this section fills in the blank and tells us the physical meaning between the relaxation.

In the following parts, we only use the second upper bound since those two upper bounds are identical.

In this section, we develop two different techniques to obtain lifetime upper bound, which could be used to identify performance bound between the proposed heuristic algorithms. For example, we could use the same techniques to develop lifetime upper bound for the MSC problem in Reference [6] and use it as a performance comparison with heuristic algorithms like LP-MSC, Greedy-MSC, in order to demonstrate their performance gain. However, we also care about how to optimally solve the T$\mathcal{Q}$C problem. In the next section, a column generation based approach is proposed to address this issue.

IV. The Second Theoretic Result: A Column Generation Based Approach

Column generation (CG) is a general purpose framework which has been often proposed either as a computationally efficient alternative to standard integer optimization methods or as a modeling tool when a direct approach is infeasible [19] [20]. 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 Basic Feasible Solutions (BFS)

The CG based approach works in the 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 BFS. Therefore, to achieve a fast convergence speed, it is important to develop methods to obtain a good initial BFS. Here, we use the random selection algorithm (Algorithm 1) proposed in our previous paper [16]. The complexity of the Random Selection Algorithm (RSA) is $O(n)$, where n is the number of sensors. The more the number of patterns, the faster this CG approach converges. Thus it is preferable to use RSA to generate multiple initial coverage patterns.

B. The Master and Sub Problem

Assume we have an initial BFS P_0 derived from the algorithm described in the above section, we can reformulate the *MINLP* optimization problem as a *Master* problem:

$$(\textit{Master}) \quad Max \sum_{p \in P_0} t_p \tag{20}$$

subject to

$$\sum_{p \in P_0} e_i^p \cdot t_p \leq e_i \quad \forall s_i \in S \tag{21}$$

$$t_p \geq 0 \quad \forall p \in P_0 \tag{22}$$

The master problem is a classical LP problem and can be solved easily with standard simplex algorithm. As stated before, after solving the master problem, we should verify its optimality. In other words, we need to identify whether it can be improved by adding new columns (coverage patterns) to the current BFS. Denote $\tilde{B}_i$ as the optimal dual variables for the energy constraint (21) in the master problem, the reduced cost c_p for the variable t_p corresponding to coverage pattern p is then:

$$c_p = 1 - \sum_{s_i \in S} \tilde{B}_i \cdot e_i^p \tag{23}$$

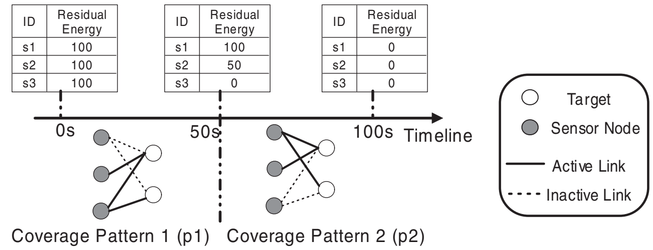


Fig. 3. An example demonstrating how CG works

TABLE II
Illustration of CG Procedure

Iteration	$\tilde{B}_1$	$\tilde{B}_2$	$\tilde{B}_3$	*Master*	*Sub*
1	0.00	0.00	0.50	50.00	1.00
2	0.50	0.00	0.50	100.00	0.00

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:

$$(\textit{Sub}) \quad Max(c_p) \tag{24}$$

subject to Eq.(2), Eq.(3) and Eq.(5).

If the solution to the sub problem results in a negative reduced cost, then the previous value from the master problem is already optimal to the original problem and CG procedure terminates. Otherwise, master problem is re-calculated with the new BFS, and the whole procedure is repeated.

To demonstrate the CG approach, we throw the instance proposed in Fig.2 into this iterative procedure. It outputs an optimal lifetime 100 in 2 iterations, the details are concluded in Tab. II and Fig.3. First, RSA randomly generates the initial BFS containing pattern p_1. In iteration 1, lifetime achieved is 50 by the *Master* formulation, optimal dural variables $\{\tilde{B}_1, \tilde{B}_2, \tilde{B}_3\}$ = {0.00,0.00,0.50}, and the *Sub* problem tells us that the maximum reduced cost equals to 1.00 and the corresponding coverage pattern is p_2, which means that there is still some room for further improvement if we add p_2 into current BFS. Therefore, in iteration 2, lifetime achieved by the *Master* formulation using both p_1 and p_2 is 100 and $\{\tilde{B}_1, \tilde{B}_2, \tilde{B}_3\}$ = {0.50,0.00,0.50}. At this time, the *Sub* problem outputs 0.00 and validate the optimality of current solution.

C. Further Optimization for the Sub Problem

We notice that the sub problem is an Integer Programming (IP) problem which belongs to class NP-hard. In a simulation environment, it is relatively easy to solve this IP formulation by using standard techniques like the Branch-and-bound algorithm [12], however, such kind of resource-consuming techniques is usually unavailable in a real-world sensor system. But algorithms developed for the Linear Programming problem, like the revised simplex method, can be implemented in a sensor node since they can work efficiently in a resource limited platform.

Thus, we develop a $O(m^3n^3)$ 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 *LP-Sub*. Then we propose a novel *LP-Sub* based rounding algorithm and prove that it is an ρ-approximation algorithm for the *Sub* problem.

We relax the *Sub* problem as following: (we use $x_{i,k}$, e_i instead of $x^p_{i,k}$ and e^p_i hereafter for convenience)

$$(\textit{LP-Sub})\ Min(\sum_{s_i \in S} \tilde{B}_i \cdot e_i) \quad (25)$$

subject to

$$\sum_{s_i \in U_k} x_{i,k} \geq q_k \quad (\forall s_i \in S, r_k \in R) \quad (26)$$

$$e_i = \sum_{k=1}^{m} x_{i,k} \quad (27)$$

$$0 \leq x_{i,k} \leq 1 \quad (28)$$

Algorithm 1: LP-Sub Based Rounding Algorithm

Input: Target Coverage Graph, priority vector
Output: coverage pattern p
begin
  $p = \emptyset$;
  Solve *LP-Sub* formulation: $x^*_{i,k}$;
  for $\forall i, k$ **do**
    if $x^*_{i,k} \geq \frac{1}{\rho}$ **then**
      add s_i to pattern p ;
      set $\overline{x}_{i,k} = 1$;
    else
      set $\overline{x}_{i,k} = 0$
  Return p ;
end

An approximation algorithm, as shown in Alg.1, is proposed to gain an integer solution based on the optimal solution of the above *LP-Sub* formulation. For this algorithm, we have:

Theorem 3. *The Sub-LP based Rounding Algorithm is an ρ-approximation $O(m^3n^3)$ algorithm for the Sub problem.*

Proof. According to the way we set $\overline{x}_{i,k}$, clearly, $\overline{x}_{i,k} \leq \rho \cdot x^*_{i,k}$, thus,

$$\sum_{s_i \in S} \tilde{B}_i \cdot \overline{e}_i \leq \rho \cdot \sum_{s_i \in S} \tilde{B}_i \cdot e^*_i \quad (29)$$

Since solution of Relaxed Linear Programming is the lower bound of the original *Sub* problem, therefore, our algorithm is 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 patterns:

$$S_1 = \{i | x^*_{i,k} < \frac{1}{\rho}\} \quad (30)$$

$$S_2 = \{i | x^*_{i,k} \geq \frac{1}{\rho}\} \quad (31)$$

Thus,

$$\sum_{i \in S_1} x^*_{i,k} < \frac{1}{\rho} \cdot \sum_{i \in S_1} 1 \leq 1 \quad (32)$$

And, according to how we set $\overline{x}$ as well as Eqn(32), we also can claim that:

$$\sum_{i=1}^{n} \overline{x}_{i,k} \geq \sum_{i \in S_2} x^*_{i,k} \geq q_k - \sum_{i \in S_1} x^*_{i,k} > q_k \quad (33)$$

Since $\sum_{i=1}^{n} \overline{x}_{i,k}$ is integer, it follows $\sum_{i=1}^{n} \overline{x}_{i,k} \geq q_k$ which means that $\overline{x}$ is also a feasible solution to the original IP formulation. Therefore, our *LP-Sub* based rounding algorithm is an ρ-approximation algorithm for the *Sub* problem. Time complexity of this algorithm is decided by the step of solving *LP-Sub* formulation which is $O(m^3n^3)$ according to [17]. $\diamond End \diamond$

Hereafter, we name CG with the revised Sub problem as revised CG. In each iteration of the revised CG, instead of using values output by *LP-Sub*, we verify the optimality using the pattern generated by the rounding algorithm. We have,

Proposition 1. *For the same instance of TQC problem, if CG outputs a lifetime T, the revised CG would also achieve the lifetime T.*

Proof. The proof is based on a fact: only when the strict termination criteria is fulfilled, column generation converges to the optimal solution. For both CG and revised CG, they terminate only when the maximum reduced cost is non-positive, therefore they both achieve the optimal T. $\diamond End \diamond$

Our revised CG performs quite well in our extensive experiments, as shown in the Section V.

D. 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 [12]. 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 4. *For the revised column generation based approach, the computation complexity would be $O(\frac{1}{4}|P|^2(|P|+1)^2 + |P|m^3n^3)$.*

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 the ith iteration, number of variables in the LP formulations would be $(i^3 + m^3n^3)$. Thus, the time complexity would be,

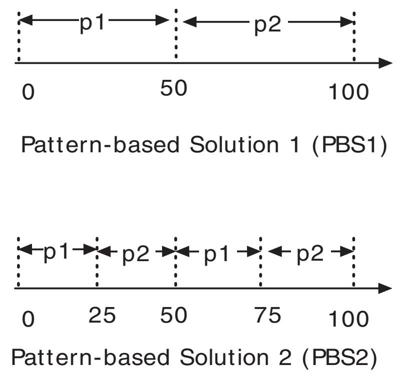


Fig. 4. Pattern-based Solutions

$$\begin{aligned} O = & \quad O(1^3 + 2^3 + \cdots + |P|^3 + |P|m^3n^3) \\ = & \quad O(\textstyle\sum_{i=1}^{|P|} i^3 + |P|m^3n^3) \\ = & \quad O(\tfrac{1}{4}|P|^2(|P|+1)^2 + |P|m^3n^3) \end{aligned}$$

$\diamond End \diamond$

E. From a Pattern-based Solution to A Feasible Schedule

In this section, we will discuss details about the procedure of generating a feasible schedule for the sensors based on the solution found by the CG based approach.

First, we define a notation as following:

Definition 3: *A pattern-based solution (PBS).* A pattern-based solution $\omega = \{T, <\vec{p}, t> = \{<p_1, t_1>, \cdots, <p_u, t_u>\}\}$ is defined as a sequence of coverage patterns, each of 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.

Clearly, 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 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.4, we use the solution for the example topology presented in Fig.3 as a pattern-based solutions PBS_1. We also give another pattern-based solutions PBS_2. For the first glance, there are two totally different solutions in the sense of ordering and frequency of patterns‘ presence. However, we notice that actually for p_1, p_2, they have been used for exactly 50 time units for both PBS_1 and PBS_2, respectively. It is quite straightforward that PBS_1 and PBS_2 share a same network lifetime. In other words, given a pattern-based solution, as long as sojourn time assigned for patterns remains the same, network lifetime remains the same regardless of the ordering and frequency of patterns' presence.

Actually, the above investigation works for the general case, and we can prove the following conclusion:

Theorem 5. *Given a pattern-based solution, as long as sojourn time assigned for patterns remains the same, the same 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 solution: $PBS_1 = \{<p_{11}, t_{11}>, \cdots, <p_{1k_1}, t_{1k_1}>\}$ with lifetime T_1 and $PBS_2 = \{<p_{21}, t_{21}>, \cdots, <p_{2k_2}, t_{2k_2}>\}$ with lifetime $T_2 \neq T_1$, where patterns and corresponding sum of assigned time duration are the same but the ordering and frequency of patterns' presence are different. Therefore the following equation must hold,

$$T_1 = \sum_{i=1}^{k_1} t_{1i} = \sum_{j=1}^{k_2} t_{2j} = T_2 \tag{34}$$

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

Algorithm 2: Sensors Scheduling Algorithm

Input: A pattern-based solution, $\omega = \{T, <\vec{p}, t> = \{<p_1, t_1>, \cdots, <p_u, t_u>\}\}$
Output: A feasible schedule
begin
  t = 0;
  for $i = 1;\ i \leq u;\ i++$ **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

Thus, 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 $O(|P|)$, as presented in Alg.2, to generate a feasible schedule.

V. Numerical Results

To verify the efficiency of our CG-based approach, we have built a simulator using Visual Studio 2005 [21] and LINGO 9.0 [18] in a Lenovo T60 laptop with 1G memory. We mainly focus on how fast it can solve T$\mathcal{Q}$C problems and how it performs when the problem scale grows. For the following experiments, we assume that all targets and nodes are randomly deployed in a $100m \times 100m$ area.

Using network parameters listed in Tab. III, in order to study performance of our proposed approach, we run the CG approach and vary the number of sensors from 60 to 100 with an increment of 10 to monitor 10 targets. Sensing range is set to 25. The corresponding results are concluded in Tab. IV. We manually select a positive performance factor $PF, PF \leq 1$. Only when CG outputs a lifetime T and $T \leq PF \cdot UPP\}$, we stop the iterative procedure. This mechanism is named as CG-*PF* and we will explain the reason later.

TABLE IV
PERFORMANCE EVALUATION. **CG-STC**: CG WITH STRICT TERMINATION CRITERIA; **CG-PF**: CG OBTAINING PF OF *UPP*; **#SEN**: NUMBER OF SENSORS;**#ITE**: NUMBER OF ITERATIONS; **#PAT**: NUMBER OF PATTERNS USED IN THE FINAL ITERATION; **CT**: COMPUTATIONAL TIME; **T**: NETWORK LIFETIME

#Sen	*UPP*	CG-50%				CG-80%				CG-STC				Revised-CG-STC			
		#Ite	#Pat	CT	T	#Ite	#Pat	CT	T	#Ite	#Pat	CT	T	#Ite	#Pat	CT	T
60	150.00	5	1	1	50.00	44	11	5	80.73	83	15	10	100.00	40	16	4	100.00
70	150.00	45	19	16	75.00	77	31	31	100.00	218	43	46	150.00	130	29	19	150.00
80	200.00	91	21	65	100.00	258	40	91	160.56	383	53	137	200.00	215	48	50	200.00
90	250.00	124	34	91	125.00	323	53	131	200.20	439	65	204	250.00	308	59	104	250.00
100	250.00	171	30	144	125.00	398	50	235	200.01	544	62	323	250.00	414	59	219	250.00

TABLE III
NETWORK PARAMETERS

Parameter	Value
m	10
SR	25
e_i	100
Q	{2,2,2,2,2,2,2,2,2,2}

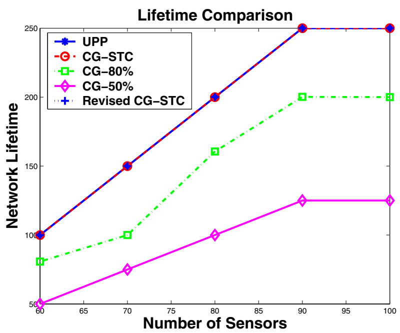


Fig. 5. Illustration of network lifetime achieved

Several key observations can be obtained from experimental data, namely,

1) *UPP* works quite well, and could be used by other approaches for the problems similar to the T$\mathcal{Q}$C problem.
2) Our CG approach performs quite well, e.g. achieving optimal solutions. Moreover, CG-*PF* can guarantee a performance bound for the T$\mathcal{Q}$C problem.
3) Revised CG also performs well, e.g. network lifetime achieved by revised CG is exactly the same as CG outputs while revised CG needs shorter time.

For the first observation, as shown in Fig.5, *UPP* is very tight and achievable. We also conduct numerous similar experiments with different network parameters and the results also confirm this observation. The central idea of *UPP* is simple but effective, thus it can be applied to other problems similar to the T$\mathcal{Q}$C problem, e.g. for the *MSC* problem [6], we can also relax the coverage constraints and get a tight upper bound. Therefore, others can utilize it as a performance bound and identify the exact performance gain their approach obtains.

The reason we develop so called CG-*PF* is also based on this observation. Convergence speed of CG based approach is a critical concern. According to observations through numerous experiments, in most cases, the termination criterion of negative reduced cost given by sub problem works efficiently; however in certain cases, the proposed approach will keep on iterating with a very small improvement in the objective function of the master problem. This phenomenon is known as the *tailing off effect* [22]. An intuitive idea is to manually terminate the CG process to cut off the long tail and get a near optimal solution. However, assume that if we do not know a tight upper bound, it is difficult to decide when we should stop the CG approach, since the exact performance gap between current solution and the optimal one is unknown. Besides, in this case, no performance guarantee can be obtained. In our case, since we know that *UPP* is quite an efficient upper bound, it could be used to guide when we should stop the iteration to save computational time while maintaining performance by a predefined factor (e.g. *PF*).

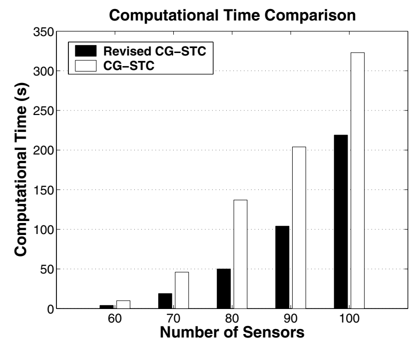


Fig. 6. Illustration of computational time comparison between CG and revised CG

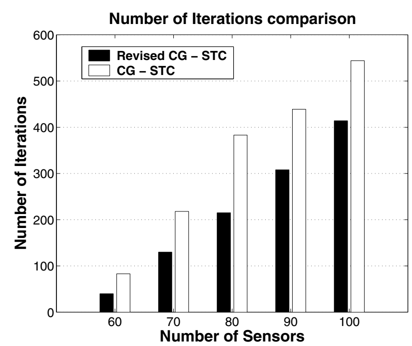


Fig. 7. Illustration of iterations comparison between CG and revised CG

For the third observation, as illustrated in Fig.6, revised CG outperforms CG in a way that it obtains the optimal solution in a shorter time compared to CG. This phenomenon can be explained when we check the number of iterations needed. Clearly, for the same problem scale, the revised CG runs much faster for one iteration compared to CG, since the time consuming sub problem (IP) in CG is replaced by a linear programming problem (LP) in the revised CG. Moreover, for the same instance, experimental data demonstrates that revised CG needs fewer iterations than CG does, as shown in Fig.7. Consequently, computational time for the revised CG is much less than CG. For example, $n = 90$, number of iterations the revised CG needs to converge is 308, corresponding to 70% of number of iterations CG needs (439). As a result, revised CG needs 104 seconds to converge and outputs an optimal lifetime 250.00, meanwhile, CG achieves this lifetime using 204 seconds. Nearly 50% of computational time has been saved.

There are still some other important observations about the CG approach. For instance, number of patterns used in the final iterations tends to grow with network lifetime. This phenomena is easy to understand, since for the same network topology, the more patterns we can use, the longer lifetime we can achieve.

VI. Conclusion

In this paper, we address the problem of achieving an optimal network lifetime in surveillance sensor networks with QoS requirements. We formulate the optimization problem in a non-linear programming form. Directly solving this optimization is desperately complicated due to the combinatorial complexity. We have two important contributions to the problem. The first is obtaining the lifetime upper bound. Based on linear programming formulations which is easy to solve, we build two upper bounds via different relaxation techniques. The connection between those two upper bound has been studied and thus physical meanings has been given to those upper bounds. The upper bounds are meaningful since the same relaxation techniques could be applied to similar target coverage problem like Maximum Set Cover problem in [6] to verify the efficiency of proposed heuristic algorithm.

The second contribution is the proposed novel column generation based approach, which decomposes the original formulation into sub-original formulations and solves them iteratively. Numerical results demonstrated that our CG based approach can efficiently solve the problem, achieving optimal or near optimal solutions.

Acknowledgements

This work has been partially supported by the National Natural Science Foundation of China under Grant No. 60872009, 60602016, Hi-Tech Research and Development Program of China 863 No. 2007AA01Z428, 2009AA01Z148; the Natural Science Foundation of Anhui under Grant No. ZD2008005-2, ZD200904, JK2009A013,JK2009A025, Grand-in-Aid for Scientific Research from Japan Society for Promotion of Science (JSPS), the Okawa Foundation for Information and Telecommunications, and NSFC under Grant no. 60740440032.

References

[1] [Online]. Available: http://www.xbow.com

[2] T. He, S. Krishnamurthy, L. Luo, T. Yan, B. Krogh, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, T. F.Abdelzaher, and J. Hui, "Vigilnet: An integrated sensor network system for energy-efficient surveillance," *ACM Transactions on Sensor Networks,TOSN*, vol. 2, pp. 1–38, 2006.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E.Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, pp. 102–114, Aug. 2002.

[4] M. Cardei and J. Du, "Energy-efficient coverage problems in wireless ad hoc sensor networks," *Computer Communications Journal*, vol. 29, pp. 413–420, Feb. 2006.

[5] M. Cardei and D.-Z. Du, "Improving wireless sensor network lifetime through power aware organization," *ACM Wireless Networks*, vol. 11, pp. 333–340, May 2005.

[6] M. Cardei, Y. L. M. Thai, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *Proc. IEEE INFOCOM 2005*, Miami, USA, Mar. 2005.

[7] M. Cardei, J. Wu, M. Lu, and M. Pervaiz, "Maximum network lifetime in wireless sensor networks with adjustable sensing ranges," in *Proc. IEEE Intl. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob'05)*, Aug. 2005.

[8] M. Lu, J. Wu, M. Cardei, and M. Li, "Energy-efficient connected coverage of discrete targets in wireless sensor networks," in *Proc. IEEE International Conference on Computer Networks and Mobile Computing(ICCNMC'05)*, Aug. 2005.

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

[10] H. Liu, P. Wan, and X. Jia, "Maximal lifetime scheduling for k to 1 sensor-target surveillance networks," *Computer Networks*, vol. 50, pp. 2839–2854, Oct. 2006.

[11] ——, "Maximal lifetime scheduling for sensor surveillance systems with k sensors to 1 target," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, pp. 1526–1536, Dec. 2007.

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

[13] T. He, P. A. Vicaire, T. Yan, L. Luo, L. Gu, G. Zhou, R. Stoleru, Q. Cao, J. A. Stankovic, and T. Abdelzaher, "Achieving real-time target tracking using wireless sensor networks," *to appear in ACM Transaction on Embedded Computing System (TECS)*, 2007.

[14] J. L. Suman Nath and F. Zhao, "Sensormap for wide-area sensor webs," *IEEE Computer Magazine*, vol. 40, pp. 90–93, 2007.

[15] G. Desaulniers, J. Desrosiers, and M. M. Solomon, *Column Generation*. New York, NY:Springer, 2005.

[16] Y. Gu, Y. Ji, J. Li, and B. H. Zhao, "Qos-aware target coverage in wireless sensor networks," *Accepted by Wireless Communications and Mobile Computing, Wiley*, 2009.

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

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

[19] 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.

[20] P. Bjorklund, P. Varbrand, and D. Yuan, "Resource optimization of spatial tdma in ad hoc radio networks: a column generation approach," in *Proc. IEEE INFOCOM 2003*, San Francisco, USA, Mar. 2003.

[21] [Online]. Available: http://msdn.microsoft.com/en-us/vs2005/default.aspx

[22] M. E. Lubbecke and J. Desrosiers, "Selected topics in column generation," *Operations Research*, vol. 53, pp. 1007–1023, Dec. 2002.