Research Article

Game-Theoretic Based Distributed Scheduling Algorithms for Minimum Coverage Breach in Directional Sensor Networks

Jin Li, 1, 2 Kun Yue, 2, 3 Weiyi Liu, 3 and Qing Liu 1, 2

1 School of Software, Yunnan University, Kunming 650091, China
2 Key Laboratory of Software Engineering of Yunnan Province, Kunming 650091, China
3 School of Information Science and Engineering, Yunnan University, Kunming 650091, China

CorrespondenceshouldbeaddressedtoJinLi;ljatynu@gmail.com

Received27September2013;Revised23January2014;Accepted17April2014;Published25May2014

Copyright © 2014 Jin Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A directional sensor network, where a lot of sensors are intensively and randomly deployed, is able to enhance coverage performances, since working directions can be partitioned into different $K$ covers which are activated in a round-robin fashion. In this paper, we consider the problem of direction set $K$-Cover for minimum coverage breach in directional sensor networks. First, we formulate the problem as a game called direction scheduling game (DSG), which we prove as a potential game. Thus, the existence of pure Nash equilibria can be guaranteed, and the optimal coverage is a pure Nash equilibrium, since the potential function of DSGs is consistent with the coverage objective function of the underlying network. Second, we propose the synchronous and asynchronous game-theoretic based distributed scheduling algorithms, which we prove to converge to pure Nash equilibria. Third, we present the explicit bounds on the coverage performance of the proposed algorithms by theoretical analysis of the algorithms' coverage performance. Finally, we show experimental results and conclude that the Nash equilibria can provide a near-optimal and well-balanced solution.

1. Introduction

In recent years, wireless sensor networks (WSNs) have attracted much attention as the promising platform for many applications, such as environmental monitoring and battlefield surveillance [1]. As a fundamental problem for WSNs, coverage optimization has been explored thoroughly in networks based on an omnidirectional sensing model [2]. Recently, with the advent and introduction of video sensors, ultrasonic sensors, and infrared sensors, coverage control algorithms for directional wireless sensors networks (dWSNs) have become an active subject and the state of the art is well surveyed in [3].

Power conversation is still a critical issue in dWSNs since the directional sensors in the network are usually battery-powered and nonrechargeable devices. Therefore, designing coverage optimization algorithms with energy efficiency is quite challenging for successful applications of dWSNs. One approach to meeting these challenges is to partition the working directions of sensors into $K$ covers. By activating a different cover at each time slot and cyclically shifting through these covers, thereby, the network’s lifetime can be extended effectively by a factor of $K$ [4].

Some efforts have recently been devoted to the research of coverage optimization with energy efficiency for directional sensor networks. For example, Ai and Abouzeid [5] proposed a directional sensing model, where a sensor is allowed to work in several directions, and the objective is to find a minimal set of directions that can cover all targets. Cai et al. [6] defined the multiple directional cover set problem of organizing the directions of sensors into a group of nondisjoint cover sets to extend the network lifetime, in order to maximize the network lifetime of a directional sensor network. The network lifetime is defined as the time duration when each target is covered by the working direction of at least one active sensor. Following the work in [6], Wen et al. [7] gave the method for prolonging the lifetime of networks based on the combination of equitable direction optimization algorithm and the neighbors sensing scheduling protocol.
Generally, the aforementioned work is mainly to maximize the network lifetime of directional sensor networks while covering all targets, which is quite strict for general coverage problems. Sometimes, due to energy constraint, coverage breach [8] (i.e., targets are not covered) may occur if available working directions in a cover are not enough to cover all the targets. Instead of finding the maximum number of directional covers for complete target coverage, the problem of direction set \( K - \text{Cover} \) for minimum coverage breach (dKC-MCB) is reduced to schedule the working directions into \( K \) covers (\( K \) is predefined) to minimize the coverage breach. Although considerable research work has been devoted to the problem of set coverage breach, few research efforts have been devoted to the problem of dKC-MCB. Even Yang et al. [11] dealt with the problem of minimum coverage breach under lifetime constraints in directional sensor network by formulating the problem as an integer programming and solving the problem by centralized greedy algorithms.

Although the existing research works have achieved some success on coverage optimization with energy efficiency in directional sensor network, some challenges still remain unanswered, especially for the problem of dKC-MCB. As mentioned in the paper [11], since the directional sensors are energy constrained to the problem of set \( K - \text{Cover} \) for minimum coverage breach in omnidirectional sensing model [4, 9, 10], to our knowledge, few research efforts have been devoted to the problem of dKC-MCB. Inspired by the previous discussion, in this paper, the problem of dKC-MCB is formulated as a game. Two game-theoretic based distributed algorithms are proposed to solve the problem. Specifically, the principal contributions of this paper are as follows.

We first formulate dKC-MCB as a game: directions scheduling game (DSG). Sensors, as players of the game, interact with each other. Each sensor makes decisions independently to maximize its individual coverage utility. The utility for a sensor is defined as the sum of the marginal coverage contribution of working directions to networks coverage.

We then prove that DSG is a potential game, whose potential function is the same as the optimization objective function of dKC-MCB. This correspondingly enables the design of a coverage optimization scheme that induces the equilibrium of a DSG consistency with the optimal coverage of the underlying network objective. Moreover, this consistency allows us to establish near-optimal performance for sensor dynamics applied to the original network coverage optimization, since the natural sensors’ utility-update dynamics converge to a Nash equilibrium.

We propose the synchronous and asynchronous distributed scheduling algorithms, which are proved to converge to pure Nash equilibria. Further, we analyze the coverage performance of the distributed algorithms from the theoretic perspective. We then present the explicit bounds on the coverage performance of both synchronous and asynchronous distributed algorithms.

2. Preliminaries

Game theory [12] is a mathematical tool that analyzes the strategic interactions among rational decision makers. Three major components in a strategic-form game model \( G = \langle N, (A_i)_{i \in N}, (u_i)_{i \in N} \rangle \) are as follows.

1. \( N \) is a finite set of \( n \) players.
2. \( A = A_1 \times A_2 \times \cdots \times A_n \), where \( A_i \) is a finite set of actions (or pure strategies) available to player \( i \). For any set \( X \), let \( \Pi(X) \) be the set of all probability distributions over \( X \). Then the set of mixed strategies for player \( i \) is \( S_i = \Pi(A_i) \). A mixed-strategy profile \( s = (s_1, \ldots, s_n) \) is Cartesian product of the individual mixed strategy. A vector \( a = (a_1, \ldots, a_n) \in A \) is called a pure-strategy profile, often denoted by \( a = (a_i, a_{-i}) \), where \( a_i \) is a strategy of player \( i \) and \( a_{-i} \) is the strategy vector of other \( n-1 \) players.
3. \( u_i(a) : a \mapsto \mathbb{R} \) is a real value utility function of player \( i \). The utility function \( u_i(a) \) measures the outcome of a player at a profile \( a \). In a game, a player chooses the proper actions against other players to maximize its individual utility.

A Nash equilibrium (NE) is a stable strategic profile where no player gets any incentive to unilaterally deviate from this profile. Thus, a Nash equilibrium in some sense is a reasonable outcome of a game. Following, we present the definition of a Nash equilibrium based on the definitions of the players’ best response.

Definition 1 (see [12]). A best response for a player \( i \) to a mixed-strategy profile \( s_i \) is a mixed strategy \( s^* \in S_i \) such that \( u_i(s^*_i, s_{-i}) \geq u_i(s_i, s_{-i}) \) for all strategies \( s_i \in S_i \).

Definition 2 (see [12]). A best response for a player \( i \) to a pure-strategy profile \( a_{-i} \) is a pure strategy \( a^*_i \in A_i \) such that \( u_i(a^*_i, a_{-i}) \geq u_i(a_i, a_{-i}) \) for all strategies \( a_i \in A_i \).
Definition 3 (see [12]). A mixed-strategy profile \( s = (s_1, \ldots, s_n) \) is a Nash equilibrium, if, for all \( i \in N \), \( s_i \) is a best response to the profile \( s \).

Definition 4 (see [12]). A pure-strategy profile \( a = (a_1, \ldots, a_n) \) is a Nash Equilibrium, if, for all \( i \in N \), \( a_i \) is a best response to the profile \( a \).

With respect to the existence of Nash equilibria of a game, Nash [12] proved that every game with a countable number of players and strategy set at least has a mixed strategy Nash equilibrium. However, in general, mixed Nash equilibrium only implies stable probability distributions over profiles, not the fixed play of a particular joint action profile. This type of uncertainty is unacceptable in many applications, such as our direction scheduling scenario. Instead, in this paper, we focus on the game with pure Nash equilibria. However, pure Nash equilibria are not available in every game. Recently, potential games, which were introduced by Monderer and Shapley [14], received increasing attention since they possess some desirable properties for engineering application scenario [15, 16], such as admitting a pure-strategy NE and best response dynamics can achieve the convergence to a pure-strategy NE.

Definition 5 (see [14]). A game \( G = (N, (A_i)_{i \in N}, (u_i)_{i \in N}) \) is a potential game if there exists a potential function \( \Phi : A \rightarrow \mathbb{R} \) such that, for all \( i \in N \), all \( a_i \in A_i \) and \( a_i', a_j' \in A_j \):

\[
\Phi(a_i', a_j) - \Phi(a_i, a_j) = \sum_{j \neq i} \frac{\partial u_i}{\partial a_i} (a_i', a_j) - \frac{\partial u_i}{\partial a_i} (a_i, a_j).
\] (1)

In a potential game, the change in a player’s payoff that results from a unilateral change in strategy equals the change in the potential function. When formula (1) is satisfied, the game is called a potential game with the potential function \( \Phi(\cdot) \). In a potential game, the set of pure Nash equilibria can be found by locating the local optima of the potential function, since the incentives of all players are mapped into one function. Thus, we have the following theorem.

Theorem 6 (see [12]). Every potential game with finite strategies space at least has a pure-strategy Nash equilibrium [14].

In addition to the existence of Nash equilibria, the quality of NE also needs to be considered. The concept of price of anarchy (PoA) [17] was introduced to measure the quality of Nash equilibria. Formally, let \( y(a) : a \rightarrow \mathbb{R} \) be the social objective function and let \( a^\text{OPT} \) be a social optimum if and only if \( a^\text{OPT} = \arg \max_{a \in A} y(a) \).

Definition 7 (see [17]). The price of anarchy of a game \( G \) is defined as \( \text{PoA}(G) = \max_{a \in A} (y(a^\text{OPT}) / y(a)) \).

Intuitively, the PoA of a game is the ratio of the objective social value of the worst possible Nash equilibrium to the value of the social optimum.

3. Problem Statement

In this section, we formally give the definitions of the problem of direction set \( K \)-Cover for minimum coverage breach in DSNs. We consider a directional sensor network of \( n \) directional sensors and \( m \) targets. Let \( S = \{s_1, \ldots, s_n\} \) and \( O = \{o_1, \ldots, o_m\} \) denote the directional sensors and the target set, respectively. Let \( D = \{D_1, \ldots, D_n\} \) be the set of directions of all sensors. Each sensor \( s_i \) has a set of directions \( D_i = \{d_{ij} \mid j = 1, 2, \ldots, w\} \). Without loss of generality, \( l_i \) is the initial lifetime for each sensor \( s_i \), which is the time duration when the sensor is in the active state all the time. \( t_k \) is the \( k \)th time slot of sensor networks and \( TL = \sum_{k=1}^{K} t_k \) is the total lifetime of sensor networks. Being similar to the problem of set \( K \)-Cover for minimum coverage breach defined in [2], the problem of dK-CMC is formally identified as follows.

Definition 8. A direction schedule of directional sensor networks is a set of ordered pairs, \( (A_k, t_k) \), \( k = 1, 2, \ldots, K \), in which \( A_k \subseteq D \) is the set of working directions in time slot \( t_k \). At a time slot \( t_k \), any sensor \( s_i \) has at most one working direction; that is, for all \( i, k, |D_i \cap A_k| \leq 1 \). Since the lifetime for each sensor is limited, for any sensor \( s_i \), one always has

\[
\sum_{k=1}^{K} z_{i,k} \cdot t_k \leq l_i,
\]

\[
\begin{cases}
 z_{i,k} = 1, & \text{if } |D_i \cap A_k| = 1 \\
 z_{i,k} = 0, & \text{if } |D_i \cap A_k| = 0.
\end{cases}
\] (2)

Definition 9. Assume that \( (A_k, t_k), k = 1, 2, \ldots, K \) is a direction schedule. The total network lifetime is given by \( TL = \sum_{k=1}^{K} t_k \). Let \( F(A_k) \) be the set of coverage targets covered by the set of active directions \( A_k \) in time slot \( t_k \). \( O \) is a target set. Total coverage breach is defined as \( \sum_{k=1}^{K} t_k |O| - |F(A_k)| \).

An average coverage breach rate is defined as \( \frac{\sum_{k=1}^{K} t_k |O| - |F(A_k)|}{\sum_{k=1}^{K} t_k} \).

Based on the definitions of direction schedule and coverage breach, we give the definition of dK-CMC.

Definition 10. Given a positive integer \( K \) and network lifetime threshold \( T_{th} > 0 \), and construct a direction schedule \( (A_k, t_k), k = 1, 2, \ldots, K \), such that the network lifetime is not less than \( T_{th} \); that is \( \sum_{k=1}^{K} t_k \geq T_{th} \) and total coverage breach \( \sum_{k=1}^{K} t_k |O| - |F(A_k)| \) is minimized. Equivalently, the problem of direction set \( K \)-Cover for minimum coverage breach is also defined to maximize \( \sum_{k=1}^{K} t_k |F(A_k)| \).

Based on the results in [11], we know that the problem of dK-CMC is NP-complete.

4. Direction Scheduling Game

A direction scheduling game is a triple \( DSG = \langle S, (A_i)_{i \in S}, (u_i)_{i \in S} \rangle \). \( S \) is a set of players, in which a player \( s_i \in S \) is a directional sensor. \( A_i \) is the set of direction scheduling strategies of \( s_i \). A strategy \( a_i \in A_i \) is an allocation of working directions of \( s_i \in S \) among the time slots and can be described as a set of ordered pairs \( a_i = \{d_{ij}, t_k \mid d_{ij} \in D_i, k = 1, 2, \ldots, K\} \). Due to the limitation of energy, a feasible strategy \( a_i \) should satisfy the next properties. (1) For all \( \{d_{ij} \neq \emptyset, t_k \} \in a_i \), \( \sum_{k=1}^{K} t_k \leq l_i \); that is, whole working lifetime of the sensor \( s_i \) is less than \( l_i \).

(2) At a time slot \( t_k \), there is at most a working direction of \( s_i \). At a time, direction scheduling strategies of all sensors are composed of a direction scheduling profile, which is denoted

International Journal of Distributed Sensor Networks 3
by \( a = (a_1, \ldots, a_i, \ldots, a_n) \). At the profile \( a \), the coverage utility of a sensor \( s_i \) is defined as follows.

Definition 11. At a profile \( a \) and a time slot \( t_k \), the working direction of sensor \( s_i \) is denoted by \( d_{i,k}^a \in D_i \). If there is no working direction for \( s_i \) at time slot \( t_k \), \( d_{i,k} = 0 \). Given a profile \( a \), the coverage utility of a sensor \( s_i \) is denoted by \( u_i(a) \) and then defined as follows:

\[
 u_i(a) = \sum_{k=1}^{K} \left( F(A_k^a) - F(A_k^a \setminus \{d_{i,k}^a\}) \right). \tag{3}
\]

Intuitively, the utility function is defined as the sum of marginal coverage contribution of the sensor’s working directions to networks coverage. In a direction scheduling game, a sensor tries to activate its directions in time slots where the sensor obtains most marginal coverage contribution. Obviously, sensors interacted with each other by maximizing each individual utility. Thus, the game is actually a dynamical interaction process. Would the interaction dynamics finally terminate and converge to a pure Nash equilibrium? In order to answer this question, we discuss the mathematical properties of direction scheduling games.

In a direction scheduling game, the set of working directions \( A_k \) in a time slot \( t_k \) is determined by the profile \( a \). Thus, at a profile \( a \), the set of working directions in a time slot \( t_k \) is denoted by \( A_k^a \) and the objective function of the problem of dKC-MCB is denoted by \( Z(a) = \sum_{k=1}^{K} t_k |F(A_k^a)| \). In what follows, we prove that direction scheduling game is a class of potential games, the potential function of which is \( Z(a) \), that is, the objective function of dKC-MCB.

Theorem 12. A direction scheduling game is a potential game with the potential function \( \Phi(a) = Z(a) \).

Theorem 13. A pure Nash equilibrium of direction scheduling game is a local optimal solution of the objective function \( Z(a) = \sum_{k=1}^{K} t_k |F(A_k^a)| \).

Theorem 14. An optimal solution of the problem of dKC-MCB is a pure Nash equilibrium of a direction scheduling game.

We give the proofs of Theorems 12, 13, and 14 in Appendices A.1, A.2, and A.3, respectively.

Some interesting mathematical properties for DSGs to solve the problem of dKC-MCB are well established by the results of Theorems 12, 13, and 14. Specifically, a DSG is proved to be a class of potential games by Theorem 12. From the previous result of Theorem 6, a DSG at least admits a pure Nash equilibrium. The connection between the solutions of dKC-MCB and pure equilibria of DSGs are described by Theorems 13 and 14. In particular, the consistence between the potential function and the objective function of dKC-MCB allows us to establish a near-optimal performance for local decision dynamics applied to the original network coverage optimization.

5. Distributed Scheduling Algorithms for Minimum Coverage Breach

In this section, based on the aforementioned properties of DSGs, we propose both synchronous and asynchronous distributed scheduling algorithms for the problem of dKC-MCB. From the perspective of game theory, both synchronous and asynchronous algorithms are actually some kind of best response dynamics of DSGs.

Specifically, \( n \) sensors are assumed to be randomly deployed in a target area. A sensor is supposed to know its location and be aware of the location of its neighbors through local communications. A sensor has a sensing range \( R_s \) and a communication range \( R_c \). In this paper, we assume the communication range of a sensor node is at least as twice as the sensing range; that is, \( R_c \geq 2R_s \). Denote \( S^i \) as the neighbors of a sensor \( s_i \). In particular, for a sensor \( s_j \in S^i \), the distance between \( s_j \) and \( s_i \) is less than \( 2R_s \). Actually, the utility of a sensor \( s_i \) depends on the strategies of the sensors within \( S^i \). In other words, \( s_i \) obtains the sum of marginal coverage contribution only through local communications with the sensors within \( S^i \). Thus, both synchronous and asynchronous algorithms are based on local information.

In a synchronous distributed algorithm, at each time step, sensors are considered to be able to synchronize their actions with one another according to a system clock. We also propose an asynchronous distributed algorithm for the case that maintaining tight clock synchronization is sometimes difficult. In asynchronous distributed algorithm, each sensor maintains its individual time clock. At a time step, only one sensor has an opportunity to update its direction scheduling strategy.

5.1. Synchronous Distributed Scheduling Algorithm. At each time step of the synchronous distributed scheduling algorithm (SDA), all the sensors are assumed to be able to synchronize their actions with one another according to a system clock. The algorithm will terminate based on a mark \( END = \neg(Update_1 \lor \cdots \lor Update_n) \).

Definition 15. Let \( Update_i \) be the mark of strategy update for a sensor \( s_i \). If a sensor \( s_i \) is able to increase utility by updating its strategy, \( Update_i \) is set to be true. Otherwise, \( Update_i \) is set to be false. Thus, the termination mark of the algorithm is defined as the following Boolean symbol \( END = \neg(Update_1 \lor \cdots \lor Update_n) \).

Specifically, if all the sensors send a message of update false to the system, the algorithm terminates. The synchronous distributed algorithm is shown in Algorithm 1.

Theorem 16. A synchronous distributed scheduling algorithm converges to a pure Nash equilibrium.

Proof. First, we prove that, at a time step of Algorithm 1, if more than one sensor is able to increase utility by updating their strategies, these sensors should be independent of one another on utility.
Input: An initial strategy of \( s_i \); A system time clock \( t = 0 \); The mark of strategy update \( Update_i \leftarrow true \).
Output: A Nash equilibrium strategy of \( s_i \).

(1) \( \text{WHILE END = false DO} \)
(2) Communicate with \( \forall s_j \in S^t \) to obtain the strategy of \( s_j \);
(3) Based on the definition of utility function, compute:
(4) \( \Delta_j(a(t)) = \max_{a'_j} (u_j(a'_j, a(t)_j) - u_j(a_j, a(t)_j)) \)
(5) and \( br_j(a(t)) = \arg \max (u_j(a'_j, a(t)_j) - u_j(a_j, a(t)_j)) \);
(6) \( \text{IF } \Delta_j(a(t)) > 0 \text{ THEN} \)
(7) Broadcasting \( \Delta_j(a(t)) \) to \( \forall s_j \in S^t \);
(8) \( \text{IF } \Delta_j(a(t)) > \max \{ \Delta_j(a(t)) \mid \forall s_j \in S^t \} \text{ THEN} \)
(9) \( a(t) \leftarrow br_j(a(t)); \) Sending \( Update_i \leftarrow true \) to the system;
(10) \( \text{END IF} \)
(11) \( \text{ELSE} \)
(12) Broadcasting \( \Delta_j(a(t)) = 0 \) to \( \forall s_j \in S^t \); Sending \( Update_i \leftarrow false \) to the system;
(13) \( \text{END IF} \)
(14) \( \text{END WHILE} \)

**Algorithm 1: Synchronous distributed algorithm (SDA).**

Let \( a(t) \) be the profile in a time step \( t \). Denote the maximal increment of utility of sensor \( s_i \) at \( a(t) \) by \( \Delta_i(a(t)) \); \( \Delta_i(a(t)) = \max_{a'_i} (u_i(a'_i, a(t)_i) - u_i(a_i, a(t)_i)) \geq 0 \). When Algorithm 1 is not terminated, \( END \) is false. From the definition of \( END \), there is at least one sensor updating its strategy. Assume that more than two sensors update strategies at the same time step. From line 6 to 10 of Algorithm 1, two utility-dependent sensors are not able to update strategies at the same time step. Actually only the sensor of the highest increments on utility has an opportunity to update a strategy. In other words, when SDA algorithm proceeds, if more than one sensor updates strategies simultaneously, then these sensors should be independent of one another on coverage utility.

Second, we prove that when Algorithm 1 proceeds, the network coverage monotonically increases.

Since direction scheduling game is a potential game, as described in (1), the change in a sensor's utility that results from a unilateral change at a profile equals the change in a global potential function. Moreover, based on Theorem 12, the potential function of direction scheduling game is the same as the optimization objective function; that is, \( \Phi(a) = Z(a) \). Therefore, the increment in a sensor's utility that results from a unilateral change at a profile equals the increment in a network coverage.

If Algorithm 1 is not terminated, \( \exists s_j \in S, \Delta_i(a(t)) > 0 \). Based on (1) and \( \Phi(a) = Z(a), \Delta_i(a(t)) > 0 \) results in the increase of \( \Phi(a) = Z(a) = \sum_{c=1}^{k} \epsilon_c F(A) \). Thus, network coverage monotonically increases. When there is more than one sensor that has opportunity to increase utility by updating strategies, from results of the first part of this proof, these sensors should be independent of one another on coverage utility. Thus, the increment of \( Z(a) \) equals the sum of increment on utility resulting from the sensors updating strategies. Also, in this case, network coverage monotonically increases.

Since \( Z(a) \) is finite, Algorithm 1 finally converges. At the time, no sensors are able to update strategies to increase utility; a pure Nash equilibrium is achieved.

5.2 Asynchronous Distributed Scheduling Algorithm. In the asynchronous distributed scheduling algorithm (ADA), we use the asynchronous time model [18], which is well matched to the distributed nature of sensor networks. In particular, each sensor \( s_i \) has an independent clock \( c_k \) whose "ticks" are distributed as a rate 1 Poisson process. A mark of updating strategy \( UT_i \) is set for each sensor \( s_i \) to permit updating its strategy. \( UT_i \) is set to be true when \( c_k \) ticks and to be false at other time. The asynchronous distributed algorithm is shown in Algorithm 2.

**Theorem 17.** An asynchronous distributed scheduling algorithm converges to a pure Nash equilibrium.

**Proof.** Assume that \( n \) sensors are deployed in a target area. Each sensor \( s_i \) has an independent clock \( c_k \) whose "ticks" are distributed as a rate 1 Poisson process. \( s_i \) has an opportunity to update its strategy when the mark \( UT_i \) is true. Since tick times are exponentially distributed, independent among sensors and independent across time, the tick time model can be equivalently formulated in terms of a single global clock ticking according to a rate \( n \) Poisson process. By letting \( \{T_k\}_{k=0}^{\infty} \) denote the arrival times for this global clock, then the individual clocks can be generated from the global clock by randomly assigning each to the sensors according to a uniform distribution. Based on properties of the Poisson process, at each arrival time of \( \{T_k\}_{k=0}^{\infty} \), there is only a sensor that has an opportunity to update its strategy.

Let \( a(t) \) be the profile in a time step \( t \). Denote \( \Delta_i(a(t)) \) as the maximal increment of utility of sensor \( s_i \) at \( a(t) \); that is, \( \Delta_i(a(t)) = \max_{a'_i} (u_i(a'_i, a(t)_i) - u_i(a_i, a(t)_i)) \geq 0 \).
Input: An initial strategy of $s_i; UT_i \leftarrow \text{false};$ The mark of strategy update $Update_i \leftarrow \text{true}.$
Output: A Nash equilibrium strategy of $s_i; a_i^*.$

(1) WHILE $END = \text{false}$ and $UT_i = \text{true}$ DO

(2) Communicate with $\forall s_j \in S_i^r$ to obtain the strategy of $s_j;$

(3) Based on the definition of utility function, compute:

\[
\Delta_i(a_i) = \max \{u_i(a'_i, a_{-i}) - u_i(a_i, a_{-i})\},
\]

and $br_i(a_i) = \arg\max \{u_i(a'_i, a_{-i}) - u_i(a_i, a_{-i})\};$

(4) IF $\Delta_i(a_i) > 0$ THEN

(5) $a_i \leftarrow br_i(a_i);$ Sending $Update_i \leftarrow \text{true}$ to the system;

(6) ELSE

(7) Sending $Update_i \leftarrow \text{false}$ to the system;

(8) END IF

(9) END WHILE

Algorithm 2: Asynchronous distributed algorithm (ADA).

When Algorithm 2 proceeds, $END$ is false. There is a sensor $s_i$ updating its strategy and $\Delta_i(a_i(t)) > 0.$ Since direction scheduling game is a potential game, based on (1) and $\Phi(a_i) = Z(a_i), \Delta_i(a_i(t)) > 0$ results in the increase of $Z(a_i).$ Thus, network coverage monotonically increases with the execution of Algorithm 2. Since the coverage objective function $Z(a_i)$ is finite, Algorithm 2 finally converges. At the time, no sensors are able to update strategies to increase utility and a pure Nash equilibrium is achieved.

6. The Coverage Performance Analysis of Distributed Algorithms

A direction scheduling game is a potential game with potential function $\Phi(a_i) = Z(a_i).$ Hence, network coverage strictly increases with execution of both synchronous and asynchronous distributed algorithms. Furthermore, both synchronous and asynchronous distributed algorithms finally converge to a stable profile, that is, a pure Nash equilibrium of direction scheduling game. Moreover, from the results of Theorem 16, the optimal solution of dKCMCBB is actually a pure Nash equilibrium of DSG. However, pure Nash equilibria are not unique for a DSG. Both synchronous and asynchronous distributed algorithms are not guaranteed to converge to optimal coverage. Thus, we should obtain explicit bounds on the coverage performance of both synchronous and asynchronous distributed algorithms. In terms of algorithmic game theory, we obtain the price of anarchy (PoA) [17] of direction scheduling game to analyze the coverage performance of proposed distributed algorithms.

Definition 18. The price of anarchy of a direction scheduling game DSG is defined as

\[
\text{PoA(DSG)} = \max_{a^* \in A} \frac{Z(a^\text{OPT})}{Z(a^*)},
\]

where $Z(a) = \sum_{k=1}^{K} f_k |F(A_k^a)|,$ that is, the objective function. $a^\text{OPT}$ is the optimal solution of $Z(\cdot).$ $a^*$ is a pure Nash equilibrium of a DSG.

Intuitively, a PoA is the ratio of optimal coverage and the worst possible Nash equilibrium coverage. Theorem 19 presents the explicit bounds on PoA of a DSG strongly depending on the submodularity [19] of coverage utility function of a DSG.

Theorem 19. The upper bound of price of anarchy of a DSG is 2.

We give the proof of Theorem 19 in Appendix B.

From the result of Theorem 19, we know that at least 1/2 optimal coverage can be obtained when synchronous and asynchronous distributed algorithms terminate.

7. Simulation Results

In this section, we evaluate the coverage performances and convergence of SDA and ADA algorithms through simulations. There are two measures for the coverage performance of algorithms. The first is the average coverage rate (ACR) of coverage optimization algorithms. ACR is an average of the coverage rate (CR) of each time slot which is the ratio between the number of targets covered by sensor networks and the number of targets located in a target area. The second is the coverage stability (CS) of coverage optimization algorithms which is measured by the variance of CRs of all time slots. The convergence is evaluated by the speed of convergence of SDA and ADA algorithms to a pure Nash equilibrium.

7.1. Experimental Demonstration of the Coverage Optimization

As an intuitive demonstration of distributed algorithms, Figure 1 shows the snapshots of coverage results of a random deployment and a SDA algorithm. In order to make the results accessible to readers, the small-scale simulations parameters are used in the demonstration. In this demonstration, $K = 3$ and each sensor has 7 sensing directions and can allocate at most 2 working directions into 3 time slots. The subfigures (a), (b), and (c) of Figure 1 show coverage results of random deployment in 3 different time slots. The subfigures (d), (e), and (f) of Figure 1 show
coverage results in 3 different time slots when SDA algorithm converges to a pure Nash equilibrium. As we can see from the results shown in Figure 1, coverage results of SDA algorithm obviously outperform results of random deployment.

7.2. Coverage Performance of the Distributed Algorithms

7.2.1. Average Coverage Rate of the Distributed Algorithms. In order to evaluate the effectiveness of algorithms, we compare the coverage performance of SDA and ADA algorithms with the RANDOM algorithm and MCBLC-G algorithm. In a RANDOM algorithm, each sensor allocates randomly working directions into time slots under the constraint of its lifetime. MCBLC-G is a centralized greedy algorithm for the problem of minimum coverage breach which is developed by Yang et al. in [11]. The experiments are set up by the following settings. The target area is a $10 \times 10$ size area where 50 targets are randomly located. We evaluate the coverage performance of the SDA, ADA, RANDOM, and MCBLC-G algorithms by randomly deployed $n = 100, 125, 150, 175, 200, 225, 250, 275, 300$ sensors with $R_s = 2$ in target area, respectively. Total lifetime $T_L$ is divided equally into 6 time slots; that is, $K = 6$. Each sensor at most chooses 3 time slots to allocate its working directions; that is, $l_i \leq 3$. In order to guarantee the reliability of simulations, for each value of $n$, we repeat the simulations by 50 times. The ACR is finally computed by $((\sum_{k=1}^{K} t_k |F(A^k)|)/K)/50$.

Figure 2 shows coverage performances of the SDA, ADA, RANDOM, and MCBLC-G algorithms. As we can see from Figure 2, SDA, ADA, and MCBLC-G algorithms provide similar coverage performances which are obviously better than results of the RANDOM algorithm. However, MCBLC-G is a centralized algorithm and the SDA and ADA algorithms are distributed algorithms.

Since the potential function of direction scheduling game coincides with the coverage objective function of the problem of dKC-MCB, the coverage of sensor network increases along with the each sensor’s utility increasing when SDA and ADA algorithms proceed. We can see the results from Figures 3 and 4.

7.2.2. Coverage Stability of the Distributed Algorithms. The coverage stability is another important performance measure of coverage optimization algorithms. A coverage optimization algorithm with good coverage stability can guarantee well-balanced coverage performance for every time slot. Figure 5 shows the coverage rate variance of SDA, ADA, RANDOM, and MCBLC-G algorithms. As we can see from the results of Figure 5, RANDOM algorithm randomly allocates working directions into time slots. Coverage rates among time slots cannot be guaranteed well balanced.
This results in a high coverage rate variance for RANDOM algorithm. MCBLC-G algorithm has a lower coverage rate variance than RANDOM. However, the coverage rate variance of MCBLC-G is influenced by the sequence of greedy selection of sensors. Compared with RANDOM and MCBLC-G algorithms, SDA and ADA provide well-balanced coverage performance.

7.3. Convergence of the Distributed Algorithms. A SDA (ADA) algorithm terminates when the algorithm converges to a pure Nash equilibrium. The speed of convergence to a pure Nash equilibrium of a SDA (ADA) algorithm mainly depends on two factors: the first is the number of sensors deployed in a target area. The second is the number of time slots $K$. As shown in Figure 6, the number of iterations for SDA algorithm increases with the number of sensors deployed in a target area. At the same time, given the number of deployed sensors, the number of iterations for SDA algorithm increases with the number of time slots $K$. As the number of $K$ increases, each sensor has more choices when it decides to allocate working directions. This leads to more complicated interactions among sensors and prolongs the convergence dynamics. Similar convergence results of ADA algorithm are shown in Figure 7.

8. Conclusions

Direction scheduling algorithms with energy efficiency are always important for directional sensor network. Since directional sensors are energy constrained, distributed direction scheduling algorithms need to be exploited where a sensor takes direction scheduling decisions independently, based purely on communications with its neighbors. In this paper, the problem of directional $K$-Cover for minimum coverage breach (dKC-MCB) is formulated as a game: direction scheduling game (DSG). Both synchronous and asynchronous game-theoretic based distributed algorithms are proposed for solving dKC-MCB, respectively. The coverage performance of distributed algorithms is analyzed from a theoretic perspective and the explicit bounds on the coverage performance are presented. Experimental results show that our proposed algorithms can provide a near-optimal and well-balanced solution to the problem of dKC-MCB.

Game theory, particularly potential game, is beginning to emerge as a powerful tool for the design and analysis of distributed optimization algorithm [13]. However, many research challenges still remain unanswered, especially for the development of a systematic methodology for the design of distributed optimization functions that satisfies virtually any degree of locality while ensuring the desirability of the resulting equilibria. As future research work, we plan to investigate a systematic approach to distributed optimization using the framework of potential games and apply this approach to solve various real application problems.

Appendices

A. Proof of Theorems 12, 13, and 14

A.1. Proof of Theorem 12. Direction scheduling game is a potential game with the potential function $Z(a) = \sum_{k=1}^{K} t_k |F(A^*_k)|$. 
Proof. Given a game \( G \), to prove \( G \) is a class of potential games, a potential function \( \Phi(\cdot) \) should be constructed, such that for any tow profiles \( a = (a_i, a_{-i}) \) and \( a' = (a'_i, a_{-i}) \) we always have

\[
u_i(a'_i, a_{-i}) - \nu_i(a_i, a_{-i}) = \Phi(a'_i, a_{-i}) - \Phi(a_i, a_{-i}). \quad (A.1)
\]

We prove that, for a direction scheduling game with utility function defined by (3), the potential function is \( \Phi(a) = Z(a) \).

Note that \( a' = (a'_i, a_{-i}) \) and \( a = (a_i, a_{-i}) \) are only different from the strategy of sensor \( s_i \) since \( s_i \) at most activates a direction in a time slot. Thus, in a time slot \( t_k \), the working directions \( A_k \) differ from \( A_k' \) in the following four possible cases.

(a) A sensor \( s_i \) schedules its working direction from \( d_{i,p} \) at a profile \( a \) to \( d_{i,q} \) at a profile \( a' \). In this case, \( A_k' = \{A_k^a \setminus \{d_{i,p}\} \cup \{d_{i,q}\}\}\).
(b) $s_i$ schedules its working direction from $d_{i,p}$ at a profile $a$ to $0$ at a profile $a'$. Thus, $A_k^a = \{A_k^a \setminus \{d_{i,p}\}\}$.

(c) $s_i$ schedules its working direction from $0$ at a profile $a$ to $d_{i,q}$ at a profile $a'$. Thus, $A_k^a = \{A_k^a \cup \{d_{i,q}\}\}$.

(d) $s_i$ maintains an equal schedule between $a$ and $a'$. Thus, $A_k^a = A_k$. Without loss of generality, at $a$ and $a'$, we suppose that $A_k^a = \{A_k^a \setminus \{d_{i,p}\}\}$, $A_k^a = \{A_k^a \setminus \{d_{i,p}\}\}$, and $A_k^a = A_k^a$, $k = 3, 4, \ldots, K$. Therefore, the potential functions of $a$ and $a'$ are, respectively, shown as follows:

$$
\Phi(a) = \sum_{k=1}^{k} F(A_k^a) = F(A_1^a) + F(A_2^a) + F(A_3^a) + \cdots + F(A_L^a),
$$

$$
\Phi(a') = \sum_{k=1}^{k} F(A_k^a) = F(A_1^a) + F(A_2^a) + F(A_3^a) + \cdots + F(A_L^a),
$$

Thus, based on (A.2) we have

$$
\Phi(a') - \Phi(a) = \sum_{k=1}^{k} F(A_k^a) - \sum_{k=1}^{k} F(A_k^a) = \begin{bmatrix} F(A_1^a) - F(A_1^a) \\ F(A_2^a) - F(A_2^a) \\ \vdots \\ F(A_L^a) - F(A_L^a) \end{bmatrix} + \begin{bmatrix} F(A_1^a) - F(A_1^a) \\ F(A_2^a) - F(A_2^a) \\ \vdots \\ F(A_L^a) - F(A_L^a) \end{bmatrix} + \cdots + \begin{bmatrix} F(A_1^a) - F(A_1^a) \\ F(A_2^a) - F(A_2^a) \\ \vdots \\ F(A_L^a) - F(A_L^a) \end{bmatrix}.
$$

In addition, based on the definition of utility function, we obtain the utility of $s_i$ at $a$,

$$
u_i(a) = \left[ F(A_1^a) - F\left(\{A_1^a \setminus \{d_{i,p}\}\}\right) \right] + \left[ F(A_2^a) - F\left(\{A_2^a \setminus \{d_{i,p}\}\}\right) \right],
$$

and at $a'$, respectively,

$$
u_i(a') = \begin{bmatrix} F\left(\{A_1^a \setminus \{d_{i,p}\}\}\right) \cup \{d_{i,q}\} - F(A_1^a) \\ \vdots \\ F\left(\{A_L^a \setminus \{d_{i,p}\}\}\right) - F(A_L^a) \end{bmatrix}.
$$

Thus, based on the results of (A.4) and (A.5), the difference of utility of $s_i$ between $a$ and $a'$ is as follows:

$$
u_i(a') - \nu_i(a) = \begin{bmatrix} F\left(\{A_1^a \setminus \{d_{i,p}\}\}\right) - F(A_1^a) \\ \vdots \\ F\left(\{A_L^a \setminus \{d_{i,p}\}\}\right) - F(A_L^a) \end{bmatrix} + \begin{bmatrix} F\left(\{A_1^a \setminus \{d_{i,q}\}\}\right) - F(A_1^a) \\ \vdots \\ F\left(\{A_L^a \setminus \{d_{i,q}\}\}\right) - F(A_L^a) \end{bmatrix}.
$$

From the results of (A.3) and (A.6), we obtain $u_i(a_i, a_{-i}) - u_i(a_{-i}) = \Phi(a_i) - \Phi(a_{-i})$. Therefore, we prove that a direction scheduling game is a potential game with potential function $\Phi(a) = Z(a)$.

A.2. Proof of Theorem 13. A pure Nash equilibrium of direction scheduling game is a local optimal solution of the problem of Set $K$-Cover for minimum coverage breach.

Proof. Let $a^*$ be a pure Nash equilibrium profile of DSG. By the definition of Nash equilibrium, at the profile $a^*$, there is no sensor that can unilaterally deviate from the profile $a^*$ by changing its scheduling strategy to increase its individual coverage utility. Since DSG is a potential game of which the potential function $\Phi(\cdot)$ is consistent with $Z(\cdot)$, that is, the optimization objective function of dKC-MCB, in a local area around the profile $a^*$, there is no profile $a'$ such that $Z(a^*) < Z(a')$. Thereby, the Nash equilibrium profile $a^*$ is local optimal of the optimization objective function of the problem of dKC-MCB.

A.3. Proof of Theorem 14. An optimal solution to the problem of dKC-MCB is a pure Nash equilibrium.
Proof. Let \( a^{\text{OPT}} = \arg \max_a Z(a) = \arg \max_a \sum_{k=1}^K t_k |F(A_k^a)| \) be the optimal solution profile of dK-C-MCB. At the profile \( a^{\text{OPT}} \), a sensor \( s_i \) unilaterally deviates from the profile \( a^{\text{OPT}} \) by changing its scheduling strategy to a profile \( a' \). Since \( a^{\text{OPT}} \) is the optimal solution profile, we obtain \( Z(a^{\text{OPT}}) \geq Z(a') \). Moreover, since a DSG is a potential game of which the potential function is consistent with \( Z(\cdot) \), \( Z(a^{\text{OPT}}) \geq Z(a') \) implies that, at the profile \( a^{\text{OPT}} \), there is no sensor that can unilaterally deviate from the profile \( a^{\text{OPT}} \) to increase its utility. By the definition of Nash equilibrium, we know the optimal solution profile \( a^{\text{OPT}} \) is a pure Nash equilibrium of a DSG. \(\Box\)

**B. Proof of Theorem 19**

The upper bound of price of anarchy of a direction scheduling game is 2.

Proof. Let \( a^* = (a_1^*, ..., a_n^*) \) be a pure Nash equilibrium of a DSG and let \( a^{\text{OPT}} = (a_1^{\text{OPT}}, ..., a_n^{\text{OPT}}) \) be the optimal profile of the problem of Set K-Cover for minimum coverage breach. Let \( (a^* \oplus a^{\text{OPT}}) \) denote a compound profile where each sensor \( s_i \) can take both \( a_i^* \) and \( a_i^{\text{OPT}} \) strategies. At a compound profile \( (a^* \oplus a^{\text{OPT}}) \), the network coverage is \( Z(a^* \oplus a^{\text{OPT}}) = \sum_{k=1}^K F(A_k^{a^* \oplus a^{\text{OPT}}}) \). Since the network coverage function \( F(\cdot) \) is a monotone function on the set of directions, we have \( Z(a^*) \leq Z(a^{\text{OPT}}) \leq Z(a^* \oplus a^{\text{OPT}}) \).

Denote \( \Delta Z_i(a^* \oplus a^{\text{OPT}}) \) by

\[
\Delta Z_i(a^* \oplus a^{\text{OPT}}) = Z(a^* \oplus (a_1^{\text{OPT}}, ..., a_{i-1}^{\text{OPT}}, a_i^*, a_{i+1}^{\text{OPT}}, ..., a_n^*)) - Z(a^* \oplus (a_1^{\text{OPT}}, ..., a_{i-1}^{\text{OPT}}, a_i^{\text{OPT}}, a_{i+1}^{\text{OPT}}, ..., a_n^*))
= Z(a_1^{\text{OPT}} \oplus a_1^*, ..., a_{i-1}^{\text{OPT}} \oplus a_{i-1}^*, a_i^{\text{OPT}} \oplus a_i^*, a_{i+1}^{\text{OPT}}, ..., a_n^*) - Z(a_1^{\text{OPT}} \oplus a_1^*, ..., a_{i-1}^{\text{OPT}} \oplus a_{i-1}^*, a_i^*, a_{i+1}^{\text{OPT}}, ..., a_n^*). \tag{B.1}
\]

Without loss of generality, assume that, at the profile \( a^{\text{OPT}} \), \( s_i \) allocates directions in time slots indexed by \( (p_1, p_2, ..., p_l) \). Thus, \( s_i \)'s strategy \( a_i^{\text{OPT}} \) is denoted as follows:

\[
a_i^{\text{OPT}} = \left\{ (t_{p_1}, d_{a_1}^{\text{OPT}}), ..., (t_{p_l}, d_{a_l}^{\text{OPT}}) \mid t_{p_k} \in T, d_{a_l}^{\text{OPT}} \in D_i; \sum_{k=1}^l t_{p_k} \leq l_i \right\}. \tag{B.2}
\]

Let \( a' = (a_1^*, ..., a_{i-1}^*, a_i^{\text{OPT}} \oplus a_i^*, a_{i+1}^*, ..., a_n^*) \); that is, \( s_i \)'s strategy \( a_i^* \) is replaced by \( a_i^{\text{OPT}} \) in a pure Nash equilibrium \( a' \). Let \( a = (a_1^*, ..., a_{i-1}^*, a_i^{\text{OPT}} \oplus a_i^*, a_{i+1}^*, ..., a_n^*) \); that is, \( s_i \)'s strategy \( a_i^* \) is replaced by a compound strategy \( a_i^{\text{OPT}} \oplus a_i^* \) in a pure Nash equilibrium \( a' \). Thus, for all \( t_{p_k}, p_k \in (p_1, p_2, ..., p_l) \) and direction \( d_{a_l}^{\text{OPT}} \in D_i \), we have \( (A_k^a \setminus \{d_{a_l}^{\text{OPT}}\}) \subseteq A_k^{a^*} \subseteq A_k^{a^{\text{OPT}}} \). Since the network coverage function is submodular function \([19]\) on the set of directions,

\[
F(A_k^a) - F(A_k^{a^*} \setminus \{d_{a_l}^{\text{OPT}}\}) \leq F(A_k^{a^{\text{OPT}}} \setminus \{d_{a_l}^{\text{OPT}}\}) - F(A_k^{a^*} \setminus \{d_{a_l}^{\text{OPT}}\}). \tag{B.3}
\]

Based on the results of formula (B.3), we know that

\[
\Delta Z_i(a^* \oplus a^{\text{OPT}}) = Z(a^* \oplus (a_1^{\text{OPT}}, ..., a_{i-1}^{\text{OPT}}, a_i^{\text{OPT}} \oplus a_i^*, a_{i+1}^{\text{OPT}}, ..., a_n^*)) - Z(a^{\text{OPT}}) = Z(a_1^{\text{OPT}} \oplus a_1^*, ..., a_{i-1}^{\text{OPT}} \oplus a_{i-1}^*, a_i^{\text{OPT}} \oplus a_i^*, a_{i+1}^{\text{OPT}}, ..., a_n^*) - Z(a_1^{\text{OPT}} \oplus a_1^*, ..., a_{i-1}^{\text{OPT}} \oplus a_{i-1}^*, a_i^*, a_{i+1}^{\text{OPT}}, ..., a_n^*) \leq u_i(a_i^{\text{OPT}}, a_i^*). \tag{B.4}
\]

Since \( a^* \) is a pure Nash equilibrium, suppose that other sensors' strategies are maintained to be unchanged, and any the coverage utility of sensor \( s_i \) is decreased by replacing \( a_i^* \) with \( a_i^{\text{OPT}} \). Thus, \( u_i(a_i^{\text{OPT}}, a_i^*) \leq u_i(a_i^{\text{OPT}}, a_i^*) \). Therefore, we further obtain

\[
Z(a^* \oplus a^{\text{OPT}}) - Z(a^*) = \sum_{i=1}^n \Delta Z_i(a^* \oplus a^{\text{OPT}}) \leq \sum_{i=1}^n u_i(a_i^*). \tag{B.5}
\]

Since \( Z(a^{\text{OPT}}) \leq Z(a^* \oplus a^{\text{OPT}}) \), we have

\[
\frac{Z(a^{\text{OPT}})}{Z(a^*)} \leq 1 + \frac{\sum_{i=1}^n u_i(a_i^*)}{Z(a^*)}. \tag{B.6}
\]

In a direction scheduling game, the utility function of a sensor is defined by the marginal contribution to network coverage; therefore, we have \( \sum_{i=1}^n u_i(a_i^*) \leq Z(a^*) \). Thereby, we obtain

\[
\text{PoA (DSG)} = \max_{a^*} \frac{Z(a^{\text{OPT}})}{Z(a^*)} \leq 1 + \frac{\sum_{i=1}^n u_i(a_i^*)}{Z(a^*)} \leq 2. \tag{B.7}
\]

\(\Box\)

**Conflict of Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.
Acknowledgments

This work was supported by the National Natural Science Foundation of China (61163003), the Yunnan Provincial Foundation for Leaders of Disciplines in Science and Technology (2012HB004), the Natural Science Foundation of Yunnan Province (2011FB020), the Foundation of Key Program of Department of Education of Yunnan Province (2013Z049), the Foundation of Development Program for Backbone Teachers of Yunnan University (XT412003), and the Foundation of the Key Laboratory of Software Engineering of Yunnan Province (2012SE030, 2012SE205).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” IEEE Communications Magazine, vol. 40, no. 8, pp. 102–114, 2002.

[2] B. Wang, “Coverage problems in sensor networks: a survey,” ACM Computing Surveys, vol. 43, no. 4, article 32, 2011.

[3] D. Tao and H. D. Ma, “Coverage control algorithms for directional sensor networks,” Journal of Software, vol. 22, no. 10, pp. 2317–2334, 2011.

[4] S. Slijepcevic and M. Potkonjak, “Power efficient organization of wireless sensor networks,” in Proceedings of the IEEE International Conference on Communications (ICC ’01), pp. 472–476, IEEE Computer Society Press, Helsinki, Finland, June 2000.

[5] J. Ai and A. A. Abouzeid, “Coverage by directional sensors in randomly deployed wireless sensor networks,” Journal of Combinatorial Optimization, vol. 11, no. 1, pp. 21–41, 2006.

[6] Y. Cai, W. Lou, M. Li, and X.-Y. Li, “Energy efficient target-oriented scheduling in directional sensor networks,” IEEE Transactions on Computers, vol. 58, no. 9, pp. 1259–1274, 2009.

[7] J. Wen, L. Fang, J. Jiang, and W. H. Dou, “Coverage optimizing and node scheduling in directional wireless sensor networks,” in Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM ’08), pp. 1–4, Dalian, China, October 2008.

[8] M. X. Cheng, L. Ruan, and W. Wu, “Coverage breach problems in bandwidth-constrained sensor networks,” ACM Transactions on Sensor Networks, vol. 3, no. 2, article 12, 2007.

[9] Z. Abrams, A. Goel, and S. A. Plotkin, “Set K-cover algorithms for energy efficient monitoring in wireless sensor networks,” in Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN ’04), pp. 424–432, ACM Press, Berkeley, Calif, USA, April 2004.

[10] A. Deshpande, S. Khuller, A. Malekian, and M. Toossi, “Energy efficient monitoring in sensor networks,” Algorithmica, vol. 59, no. 1, pp. 94–114, 2011.

[11] H.-Q. Yang, D.-Y. Li, and Z. Li, “Minimum coverage breach and maximum network lifetime in directional sensor networks,” Acta Electronica Sinica, vol. 38, no. 2, pp. 138–142, 2010.

[12] D. Fudenberg and J. Tirole, Game Theory, MIT Press, Cambridge, Mass, USA, 1991.

[13] B. Yang and M. Johansson, “Distributed optimization and games: a tutorial overview,” in Networked Control Systems, vol. 406 of Lecture Notes in Control and Information Sciences, pp. 109–148, Springer, London, UK, 2010.

[14] D. Monderer and L. Shapley, “Potential games,” Games and Economic Behavior, vol. 14, no. 1, pp. 124–143, 1996.

[15] J. R. Marden, G. Arslan, and J. S. Shamma, “Cooperative control and potential games,” IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics, vol. 39, no. 6, pp. 1393–1407, 2009.

[16] U. O. Candogan, I. Menache, A. Ozdaglar, and P. A. Parrilo, “Near-optimal power control in wireless networks: a potential game approach,” in Proceedings of the 29th Conference on Information Communications (INFOCOM ’10), pp. 1954–1962, IEEE Computer Society Press, San Diego, Calif, USA, March 2010.

[17] E. Koutsoupias and C. H. Papadimitriou, “Worst-case equilibria,” in Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science (STACS ’99), pp. 404–413, 1999.

[18] A. D. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, “Geographic gossip: efficient averaging for sensor networks,” IEEE Transactions on Signal Processing, vol. 56, no. 3, pp. 1205–1216, 2008.

[19] S. Fujishige, Submodular Functions and Optimization, Elsevier, Amsterdam, The Netherlands, 2nd edition, 2005.
