# Leveraging Prediction to Improve the Coverage of Wireless Sensor Networks

Shibo He, Jiming Chen, *Senior Member*, *IEEE*, Xu Li,
Xuemin (Sherman) Shen, *Fellow*, *IEEE*, and Youxian Sun

**Abstract**—As sensors are energy constrained devices, one challenge in wireless sensor networks (WSNs) is to guarantee coverage and meanwhile maximize network lifetime. In this paper, we leverage prediction to solve this challenging problem, by exploiting temporal-spatial correlations among sensory data. The basic idea lies in that a sensor node can be turned off safely when its sensory information can be inferred through some prediction methods, like Bayesian inference. We adopt the concept of entropy in information theory to evaluate the information uncertainty about the region of interest (RoI). We formulate the problem as a *minimum weight submodular set cover* problem, which is known to be NP hard. To address this problem, an efficient centralized truncated greedy algorithm (TGA) is proposed. We prove the performance guarantee of TGA in terms of the ratio of aggregate weight obtained by TGA to that by the optimal algorithm. Considering the decentralization nature of WSNs, we further present a distributed version of TGA, denoted as DTGA, which can obtain the same solution as TGA. The implementation issues such as network connectivity and communication cost are extensively discussed. We perform real data experiments as well as simulations to demonstrate the advantage of DTGA over the only existing competing algorithm [1] and the impacts of different parameters associated with data correlations on the network lifetime.

**Index Terms**—Prediction, temporal-spatial correlations, coverage, network lifetime, wireless sensor networks.

## 1 Introduction

Last decade has witnessed the rapid development of wireless sensor networks (WSNs) [2], [3], [4], [5], [6]. Consisting of a large number of small-size limited-capability sensor nodes, WSNs are widely used in a variety of application scenarios such as surveillance and environment monitoring. In all these scenarios, a fundamental concern is the quality of sensing, which is often referred to as *coverage* and quantifies the collected information about the region of interest (RoI). Traditional approaches to *coverage* are based on the idea that we can have complete knowledge about the RoI if every point in the RoI is covered. They use a perfect disc sensing model, and resort to computational geometry to get an optimal deployment of sensor nodes [2], [3], [7]. To ensure fault tolerance, $k$-*coverage* solutions have been proposed to guarantee that every point in the RoI is covered by at least $k$ different sensor nodes. Traditional approaches offer general ways to address *coverage*, however, they do not exploit the unique characteristics of applications to enhance energy efficiency. Hence, they are conservative and, when used, will greatly restrict the network's potential in longer term sensing. For example, in target tracking [8], coverage holes may exist and the performance of a WSN can be acceptable as long as any moving object or phenomena can be detected before it travels farther than a predefined distance threshold (regardless of its trajectory and speed). If we enforce the coverage of every point in the RoI, a large amount of energy will be squandered and the network lifetime will be shortened as a result.

Due to the limitation of traditional approaches on prolonging network lifetime, researchers are more inclined to address coverage for a specific application scenario [8], [9]. In [9], the statistical characteristics of event dynamic is exploited to duty cycle sensor nodes in the applications of event monitoring. The existence of coverage hole at certain time will not impact event detection ratio if events are absent during that time. It is shown that the network lifetime can be prolonged significantly only at a minor increase of event loss. And, it is also confirmed that prior knowledge of event dynamic is important to energy efficiency. Inspired by their work, we focus on how to extend network lifetime by exploiting prior knowledge of data structure in this paper.

Spatial data, e.g., temperature, collected by sensor nodes is well recognized to have strong temporal and spatial correlations [10]. We aim to exploit such correlations to improve energy efficiency and guarantee the coverage of WSNs. The basic idea is that once the sensory information at a sensor node can be predicted by other nodes, the node may be turned off safely without undermining coverage performance. How to select a set of active sensor nodes to perform prediction for inactive nodes is therefore essential to energy saving.

We measure the expected prediction error, also referred to as uncertainty, of the information at inactive sensor nodes by

- *S. He is with the State Key Laboratory of Industrial Control Technology, Department of Control, Zhejiang University, Hangzhou 310027, P.R. China, and also the Department of Electrical and Computer Engineering, University of Waterloo. E-mail: shibohe.cn@gmail.com.*
- *J. Chen and Y. Sun are with the State Key Laboratory of Industrial Control Technology, Department of Control, Zhejiang University, Hangzhou 310027, P.R. China. E-mail: {jmchen, yxsun}@iipc.zju.edu.cn.*
- *X. Li is with the Inria Lille - Nord Europe, Parc scientifique de la haute borne, 50 avenue Halley, 59650 Villeneuve d'Ascq, France. E-mail: xu.li@inria.fr.*
- *X. Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada. E-mail: xshen@bbcr.uwaterloo.ca.*

*Manuscript received 9 Mar. 2011; revised 3 May 2011; accepted 2 June 2011; published online 16 June 2011.*
*Recommended for acceptance by X. Cheng.*
*For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2011-03-0136.*
*Digital Object Identifier no. 10.1109/TPDS.2011.180.*

entropy, which is a well-known submodular set function [11]. We refer to a set of sensor nodes as a *submodular set cover* if only activating sensor nodes in the set can achieve no information loss about the RoI by using the entropy metric. Associating each sensor node with an energy-related weight, the problem boils down to finding a *submodular set cover* at each time slot to extend the network lifetime, i.e., *minimum weight submodular set cover* (MWSSC) problem.

As the MWSSC problem is NP hard [12], we concentrate on approximation algorithms. Greedy algorithm (GA) [12] has been proposed to solve the general MWSSC problem, but it has two limitations: 1) it is a centralized algorithm and does not fit the decentralized nature of WSNs, and 2) it has an implicit, possibly unbounded, performance guarantee, in terms of the ratio of the obtained total weight to the optimal one. The goal is then to design an MWSSC solution without these drawbacks to duty cycle sensors for joint optimization of coverage and network lifetime in wireless sensor networks.

Our contributions are threefold: 1) the formulation of MWSSC problem for sensor activity scheduling for joint optimization of coverage and network lifetime, and the design of algorithms that provides a $\varepsilon$-approximation solution to MWSSC, 2) the investigation of implementation issues, and 3) the performance evaluation of the proposed algorithms. Here, $\varepsilon$ is application-specific. It can be determined by the threshold within which the missed information will not impact the reconstruction of the information field.

To solve MWSSC, we propose a *truncated greedy algorithm* TGA. We obtain an explicit performance guarantee of TGA, proving that the weight of set cover obtained by TGA is less than $\omega(C(OPT))(1+\ln\zeta)+N\overline{\omega}$, where $\omega(C(OPT))$ is the weight of set cover found by optimal algorithm, $\zeta$ and $\overline{\omega}$ two constants only dependent on the problem itself, and $N$ the number of sensor nodes. We then show that TGA can be easily modified to obtain a desired fully distributed version DTGA, where each sensor communicates only with sensors in its data correlation range (CR) instead of the whole network. We prove that DTGA finds the same submodular set cover as TGA.

We investigate the implementation issues of DTGA. One issue is how to model the correlations among data and to perform prediction. We introduce *Gaussian process* (GP) [13], to model the intrinsic temporal-spatial correlations for a specific type of sensory data and, adopt Bayesian inference to make predictions [14]. Gaussian process offers a systematic framework for model learning and selection, and possesses many advantages such as easy interpretation of model predictions. It is fully specified by a mean and a covariance function. Based on the real temperature data collected from US national climatic data center (NCDC) [15], we illustrate how to select and train a practical mean and covariance function. We also discuss other issues, e.g., network connectivity and communication cost, etc.

We perform real data experiments as well as simulations to verify our theoretical findings and demonstrate the advantages of DTGA. Real temperature data from US weather stations are incorporated to illustrate data modeling and demonstrate the effectiveness of DTGA. In addition, extensive simulation is conducted to evaluate the impacts of different parameters of covariance functions on the network lifetime under DTGA. Our simulation results show that DTGA can achieve a significantly longer network lifetime compared with the only existing competing algorithm [1].

The remainder of the paper is organized as follows: We discuss existing work on coverage in Section 2. We formulate the MWSSC problem in Section 3. We propose a centralized truncated greedy algorithm to solve this problem in Section 4 and the distributed TGA, i.e., DTGA, in Section 5. We discuss some implementation issues in Section 6 and validate our algorithms through simulation in Section 7. We conclude the paper in Section 8.

## 2 Related Work

Coverage is a fundamental problem in WSNs. Traditional solutions adopt a perfect disc sensing model [16]. A point in the RoI is said to be covered by the sensor network if it falls into the sensing range of at least one sensor node. Deterministic solutions [17], [18], [19] has been investigated to deploy a minimum number of sensor nodes to ensure $k(k \geq 1)$ *coverage* as well as *connectivity*. As deterministic deployment could incur huge management and implementation cost, random deployment is more popular. Ongoing efforts [2], [20], [21], [22], [23] have been invested to conceive scheduling schemes to guarantee coverage and enhance energy efficiency. Tian and Georganas [20] calculated, by computational geometry, whether the sensing region of a sensor node is covered by the union sensing region of its neighbors, and proposed a *coverage-preserving node scheduling* scheme to duty cycle sensor nodes accordingly. This method was adopted and further developed in [9], [24]. Zhang and Hou [2] considered joint coverage and connectivity problem, and indicated that full coverage of a convex region implies connectivity if the communication range is at least two times of sensing range. They also gave a set of optimality conditions for scheduling sensor nodes, by which a distributed algorithm was proposed.

Recent studies focus on some new performance metrics of coverage in specific applications. These studies neither assume a perfect disc sensing model or impose a conservative requirement of no coverage hole. One of the performance metrics is *information coverage*, which assumes a probabilistic sensing model [25], [26]. A point (where an event may occur) is defined to be covered if the absolute value of the estimation error of received signals is less than a threshold. The concept of *information coverage* is further employed to design an efficient and robust barrier coverage in [26]. Balister et al. [8] introduced *trap coverage* in the applications of tracking, evaluating the coverage by a threshold, within which any moving object can be detected by the networks. They estimated the deployment density required to satisfy trap coverage if sensor nodes are deployed according to Poisson point process. He et al. [9] focused on applications of stochastic event capture and demonstrated for the first time that event dynamic can be exploited for sleep scheduling and significantly prolonging the network lifetime. Xing et al. [27] demonstrated that inclusion of data fusion can improve the coverage of the networks. They used false alarm rate $P_F$ and detection probability $P_D$ to determine whether a point is covered or not.

The most relevant work to ours presented here is [1], which exploited the spatial correlation to select a subset of sensor nodes for efficient data gathering. However, it focused only on the spatial structure of sensory data,

overlooking the spectrum of temporal dimension. Because it used *linear regression* to model the correlations, the results are hard to generalize and apply to other applications. In this paper, we address temporal-spatial correlation of data and adopt *Gaussian process*, a generalization of linear regression, to model the structure of sensory data. Gaussian process provides a systematic framework for model selection, and makes it easy to interpret the model predictions. Based on Gaussian process for prediction, we formulate the problem of *minimum weight submodular set cover* for energy-efficient coverage. Simulation results indicate that our approach can significantly extend the network lifetime compared with [1].

## 3 Problem Formulation

We consider a wireless sensor network with $N$ nodes densely deployed in a geographic region of interest. Without loss of generality, we will also use $N$ as the set of sensor nodes throughout the rest of the paper. Each sensor node can collect sensory information about its surroundings and report the information to the sink node through multihop message relay. Initially, RoI is completely covered by the sensor network, and the full coverage can be verified through the existing work of coverage hole verification [28]. This is reasonable as sensors are densely deployed. Under this circumstance, there is no information loss about the RoI if we know the information gathered at every sensor node. Each sensor node $i$ has a unique identifier $ID_i$ and a transmission range $R_t$. It knows its own location through GPS or other localization techniques [29].

The operation time of the WSN is divided into time slots. Due to the dense sensor deployment, there is wide existence of temporary and spatial correlations among sensory data [10]. Each sensor node $i$ is associated with a random variable $\mathcal{X}_i, i \in N$, which denotes the values of sensory information at node $i$. Let $C$ be a subset of $N$ and $\mathcal{X}_C$ denote a set of random variables indexed by sensor nodes in $C$, i.e., $\mathcal{X}_C = (\mathcal{X}_i)_{i \in C}$. A sensor node $i$ has a correlation set $R_i(t)$ composed of sensors whose data are currently spatially correlated with $i$ at slot $t$. Denote the diameter of a correlation set $R_i(t)$ by $\overline{R}_i(t)$, which is defined as the largest distance between any two arbitrary nodes in $R_i(t)$. Let $\overline{R}(t)$ be the largest diameter of all $R_i(t)$, $i = 1, 2, \ldots, N$. Due to temporal correlation, data gathered by nodes in $R_i$ at slot $t', t' = t-1, t-2, \ldots$, may be correlated with that collected by node $i$ at current slot $t$. Let $\overline{R}_{tc}(t)$ denote the largest number of past time slots, within which sensory data are considered to be correlated with that at current slot $t$. Hence, $\mathcal{R}_i(t) = \{R_i(t), R_i(t-1), \ldots, R_i(t-\overline{R}_{tc}(t))\}$ is the correlation set of node $i$. Let $x_i(t)$ be the value of information gathered by sensor node $i, i \in N$ at time $t$. Obviously, $x_i(t)$ is a sample value of $\mathcal{X}_i$ at time $t$. If data sensed at node $i$ can be inferred by data from its correlation set $\mathcal{R}_i$, $i$ can be turned off without any degradation of coverage performance. Denote the entropy function by $H(\cdot)$. For a set of random variables $\mathcal{X}_C$ ($C = \{1, 2, \ldots, m\}$), $H(\mathcal{X}_C)$ is the joint entropy, that is,

$$H(\mathcal{X}_C) = -\int_{x_1,\ldots,x_m} \Pr(x_1, \ldots, x_m) \log \Pr(x_1, \ldots, x_m) dx_1 \ldots dx_m.$$

From the perspective of information theory, $x_i(t)$ can be inferred if and only if the conditional entropy

$$H(\mathcal{X}_i|\mathcal{X}_{\mathcal{R}_i}) = -\int_{x_i, x_{\mathcal{R}_i}} \Pr(x_i, x_{\mathcal{R}_i}) \log \Pr(x_i|x_{\mathcal{R}_i}) dx_i d_{x_{\mathcal{R}_i}},$$

will approach zero. The smaller the entropy, the more we know about the unknown information.

We adopt the same energy consumption model as used in [30]. We assume each sensor node $i \in N$ has an initial energy of $E_i$. Denote by $l_i(t)$ the energy consumed at sensor node $i$ before time slot $t$. At the beginning of each time slot, a node selection algorithm (NSA) is executed. Any sensor node selected by NSA should be active during the current slot while others may go to sleep. Without loss of generality, we further assume $E_i$ is an integer, and one unit energy will be depleted if the sensor is scheduled to be active at the slot. The network lifetime is defined as the number of time slots before the first sensor node in the network runs out of energy. We concentrate on how to conceive a NSA in order to maximize the network lifetime. Before formally formulating the problem, we give the following important definitions.

**Definition 1 (Submodularity).** *A set function $f$ is submodular if and only if $f(A \cup S) - f(A) \geq f(B \cup S) - f(B)$ for any set $S$, $A$, $B$, satisfying $A \subseteq B$.*

According to [11], entropy $H(\cdot)$ is a submodular set function. For ease of presentation, we use $H(C)$ instead of $H(\mathcal{X}_C)$ to denote the entropy of $\mathcal{X}_C$ when there is no confusion.

**Definition 2 (Weight).** *Assume each sensor node $i$ is associated with a weight $w_i$, and $A$ is a set of sensor nodes, then the weight of $A$ is defined as*

$$w(A) = \sum_{i \in A} w_i.$$

The weight of a sensor node $i$, $w_i$, can be interpreted as the cost incurred when $i$ is activated. In this paper, $w_i$ is a function of the residual energy of node $i$, e.g., $w_i = \kappa^{-\frac{E_i - l_i}{E_i}}$, where $\kappa$ is a constant.

**Definition 3 (Submodular set cover).** *A subset $C$ of $N$ sensor nodes is a* submodular set cover *if and only if 1) any sensor node $i, i \in C$, has a residual energy no less than 1 unit, and 2) $H(C) = H(N)$. $C$ is said to be $\varepsilon$-*approximation submodular set cover *if the condition 2) is changed to $H(C) \geq H(N) - \varepsilon$.*

With the definitions above, we formulate the *minimum weight ($\varepsilon$-approximation) submodular set cover (MWSSC/$\varepsilon$-MWSSC)* problem as follows:

**Definition 4 (MWSSC/$\varepsilon$-MWSSC).** *Given a sensor network consisting of $N$ nodes, each with a weight $w_i$, the MWSSC/$\varepsilon$-MWSSC problem is to find a subset $C$ of $N$ with minimum aggregate weight, such that $C$ is a ($\varepsilon$-*approximation) submodular set cover, *i.e.,*

$$\min_{C \subseteq N} w(C) \quad s.t. \quad H(C) = H(N) \text{ (or } H(C) \geq H(N) - \varepsilon). \tag{1}$$

Notice that, when $\varepsilon$ approaches zero, $\varepsilon$-MWSSC approaches MWSSC. In the sequel, we mainly focus on $\varepsilon$-MWSSC to give a general solution.

## 4 Truncated Greedy Algorithm

As proven in [12], the general submodular set cover problem is NP-hard. Therefore, we focus on designing an efficient approximation algorithm to solve the minimum weight submodular set cover problem.

Due to the monotonicity of submodular set function, greedy algorithm is a competitive candidate. However, existing proposed greedy algorithm has two significant drawbacks, which make it unsuited to apply in this paper. The first drawback is that general greedy algorithm is a centralized algorithm, and difficult to implement distributively. The second one is that it has an implicit, possibly unbounded performance ratio in terms of the obtained aggregate weight to the optimal one [12].

We propose a centralized truncated greedy algorithm to overcome these drawbacks, with a minor performance degradation, i.e., TGA finds an $\varepsilon$-approximation submodular set cover. We derive an explicit performance guarantee of TGA. In the next section, we modify TGA to obtain a distributed algorithm.

Let $\rho_i(A) = H(A \cup i) - H(A)$. Denote by $C^k$ the submodular set that TGA finds at iteration $k$, $S$ the set of sensor nodes that have decided to go to sleep. At each iteration $k$, TGA calculates $\rho_i(C^{k-1})$ for every sensor $i \in N \setminus C^{k-1} \setminus S$, removes the sensors who have $\rho_i(C^{k-1})$ less than $\frac{\varepsilon}{N}$, and adds into set $C^k$ the sensors $i_k \in N \setminus C^{k-1} \setminus S$, satisfying

$$i_k = \min_{i \in N \setminus C^{k-1} \setminus S} \frac{w_i}{\rho_i(C^{k-1})}.$$

The algorithm terminates when set $N \setminus S \setminus C^k$ is empty or $H(C^k) = H(N)$. We sketch TGA in Algorithm 1.

**Algorithm 1.** Truncated Greedy Algorithm (TGA)

1) $k = 0$; $C^k = \Phi$, $S = \Phi$;
   if $H(\Phi) = H(N)$,
   then stop and return $\Phi$.
2) $k = k + 1$;
   if $\rho_i(C^{k-1}) < \frac{\varepsilon}{N}$ for $i \in N \setminus C^{k-1} \setminus S$,
   then add $i$ into $S$, i.e., $S = S \cup i$;
   let
   $$i_k = \min_{i \in N \setminus C^{k-1} \setminus S} \frac{w_i}{\rho_i(C^{k-1})};$$
   if there are multiple nodes selected as $i_k$,
   then let $i_k$ be the one with minimum ID;
   add $i_k$ into set $C^k$, i.e.,
   $$C^k = C^{k-1} \cup i_k;$$
   let
   $$\theta^k = \frac{w_{i_k}}{\rho_{i_k}(C^{k-1})};$$
3) if $N \setminus C^k \setminus S = \Phi$ or $H(C^k) = H(N)$,
   then let $T = k$, stop and return $C^T$;
   otherwise go to step 2).

**Remark.** From the process of TGA, the total information loss $H(N) - H(C^T)$ can be bounded by

$$H(N) - H(C^T) = \sum_{i \in N \setminus C^T} \rho_i(C^T) \le \sum_{i \in N} \rho_i(C^T) \le \sum_{i \in N} \frac{\varepsilon}{N} \le \varepsilon.$$

Therefore, TGA finds a $\varepsilon$-approximation solution to MWSSC.

The rationale of TGA is: if adding a sensor node into the submodular set cover hardly contributes to the increase of information about the RoI, we remove it from the candidate set. The algorithm has an advantage of avoiding (possibly) the extremely low-convergence rate of the algorithm. Below we will give a theoretical bound of the performance of TGA.

**Lemma 1.** *Assume* $0 < a_1 \le a_2 \le \cdots \le a_n$ *and* $b_1 \ge b_2 \ge \cdots \ge b_n > 0$. *Let* $L = \sum_{j=1}^{n-1} a_j(b_j - b_{j+1}) + a_n b_n = a_1 b_1 + \sum_{j=1}^{n-1}(a_{j+1} - a_j) b_{j+1}$. *Then,*

$$L = \left(\max_j a_j b_j\right)\left[1 + \ln \min\left(\frac{a_n}{a_1}, \frac{b_1}{b_n}\right)\right].$$

**Proof.** See detailed proof in [12]. □

Let $\overline{\omega} = \max_{i \in N} \omega_i$, $\underline{\omega} = \min_{i \in N} \omega_i$, $\overline{\rho} = \max_{i \in N} \rho_i(\Phi)$ where $\Phi$ represents empty set, $\zeta = \frac{\overline{\omega}\overline{\rho}N}{\underline{\omega}\varepsilon}$, and $\alpha = \frac{\overline{\omega}}{1+\ln\zeta}$. Denote the submodular set covers with minimum weight obtained by TGA and optimal algorithm, respectively, by $C(TGA)$ and $C(OPT)$. We have the following Theorem:

**Theorem 1.**

$$\omega(C(TGA)) \le \omega(C(OPT))(1 + \ln\zeta) + N\overline{\omega}. \tag{2}$$

**Proof.** Consider the following linear programming $P_1$ and its dual problem $P_2$.

$$\begin{aligned} P_1: \quad & \min \sum_{i \in N} (\omega_i + \alpha) y_i \\ s.t. \quad & \sum_{i \in N} \rho_i(C^{k-1}) y_i \ge H(N) - H(C^{k-1}) \\ & k = 1, 2, \ldots, T, \end{aligned}$$

and

$$\begin{aligned} P_2: \quad & \max \sum_{k=1}^{T} (H(N) - H(C^{k-1})) z_k \\ s.t. \quad & \sum_{k=1}^{T} \rho_i(C^{k-1}) z_k \le \omega_i + \alpha, i \in N, \end{aligned} \tag{3}$$

where $C^k$, $k = 1, 2, \ldots, T-1$, is the set cover obtained by TGA at iteration $k$, and $C^0 = \Phi$.

Obviously, for each $i \in N$, $\rho_i(C^0) \ge \rho_i(C^1) \ge \cdots \ge \rho_i(C^{T-1}) \ge 0$. Due to the termination condition, $\rho_i(C^T)$, $i \in N$, is either 0 or less than $\frac{\varepsilon}{N}$. Let $k_i$ be the iteration that satisfies

$$\rho_i(C^{k_i}) \ge \frac{\varepsilon}{N} \quad \text{and} \quad \rho_i(C^{k_i+1}) < \frac{\varepsilon}{N}.$$

According to the greedy selection of $\theta^k$, $k = 1, 2, \ldots, T$, $0 < \theta^1 \le \theta^2 \le \cdots \theta^T$, and $\theta^k \rho_i(C^{k-1}) \le \omega_i, \forall k = 1, 2, \ldots, k_i$, and we have

$$\begin{aligned}
&\theta^1\rho_i(C^0)+\sum_{k=2}^{T}(\theta^k-\theta^{k-1})\rho_i(C^{k-1})\\
&<\theta^1\rho_i(C^0)+\sum_{k=2}^{k_i}(\theta^k-\theta^{k-1})\rho_i(C^{k-1})\\
&\quad+\sum_{k=k_i+1}^{T}\frac{\varepsilon}{N}(\theta^k-\theta^{k-1})\\
&\le\max_{k=1,2,\ldots,k_i}\{\theta^k\rho_i(C^{k-1})\}\left(1+\ln\frac{\theta^{k_i}}{\theta^1}\right)+\frac{\varepsilon}{N}(\theta^T-\theta^{k_i})\\
&\le\omega_i(1+\ln\zeta)+\overline{\omega}.
\end{aligned}$$

The second inequality holds because of Lemma 1 and the fact of $\sum_{k=k_i+1}^{T}(\theta^k-\theta^{k-1})=\theta^T-\theta^{k_i}$; the third one holds because $\theta^T=\min_{i\in N\setminus C^{T-1}\setminus S}\frac{w_i}{\rho_i(C^{T-1})}\le\frac{N\max_{i\in N}w_i}{\varepsilon}=\frac{N\overline{w}}{\varepsilon}$.

Let $\Theta=(\theta^1,\theta^2-\theta^1,\ldots,\theta^T-\theta^{T-1})$. Then, $\frac{1}{1+\ln\zeta}\Theta$ is a feasible solution of linear programming $P_2$. Let $D(\frac{1}{1+\ln\zeta}\Theta)$ be the objective value of (3) corresponding to the feasible solution $\frac{1}{1+\ln\zeta}\Theta$, $\omega(OPT_1)$ and $\omega(OPT_2)$ be the optimal values of Problem $P_1$ and $P_2$, respectively. We have

$$\begin{aligned}
D\left(\frac{1}{1+\ln\zeta}\Theta\right)&=\frac{1}{1+\ln\zeta}\left\{\theta^1(H(N)-H(C^0))\right.\\
&\quad\left.+\sum_{k=2}^{T}(\theta^k-\theta^{k-1})(H(N)-H(C^{k-1})\right\}\\
&\le\omega(OPT_2)=\omega(OPT_1).
\end{aligned}\tag{4}$$

On the other hand, the $\omega(C^T)$ obtained by TGA is given by

$$\begin{aligned}
\omega(C^T)&=\sum_{i\in C^T}\omega_i=\sum_{k=1}^{T}\rho_{i_k}\theta^k\\
&=\sum_{k=1}^{T}\theta^k(H(C^k)-H(C^{k-1}))\\
&\le\theta^T(H(N)-H(C^{T-1}))\\
&\quad+\sum_{k=1}^{T-1}\theta^k(H(C^k)-H(C^{k-1}))\\
&=\theta^1(H(N)-H(C^0))\\
&\quad+\sum_{k=2}^{T}(\theta^k-\theta^{k-1})(H(N)-H(C^{k-1})\\
&=(1+\ln\zeta)D\left(\frac{1}{1+\ln\zeta}\Theta\right)\\
&\le(1+\ln\zeta)\omega(OPT_2).
\end{aligned}\tag{5}$$

The first inequality holds because $H(C^T)\le H(N)$, and the fourth equation holds according to (4).

Consider the following three linear integer programming problems:

$$\begin{aligned}
P_3:\quad&\min\sum_{i\in N}(\omega_i+\alpha)y_i\\
s.t.\quad&\sum_{i\in N}\rho_i(A)y_i\ge H(N)-H(A),A\subseteq N\\
&y_i\in\{0,1\}
\end{aligned}$$

$$\begin{aligned}
P_4:\quad&\min\sum_{i\in A}(\omega_i+\alpha)\\
&s.t.\ H(A)=H(N),\ A\subseteq N,
\end{aligned}$$

and

$$\begin{aligned}
P_5:\quad&\min\sum_{i\in A}\omega_i\\
&s.t.\ H(A)=H(N),\ A\subseteq N.
\end{aligned}$$

Denote the optimal solutions of $P_3$, $P_4$, and $P_5$ by $OPT_3$, $OPT_4$, and $OPT_5$, respectively. Obviously, $P_2$ is a relax of $P_3$, leading to $\omega(OPT_2)\le\omega(OPT_3)$. For $A\subseteq N$, $\sum_{i\in A}(\omega_i+\alpha)\le\sum_{i\in A}\omega_i+N\alpha$. Therefore, we have

$$\omega(OPT_4)\le\omega(OPT_5)+N\alpha,$$

from which we get

$$\omega(OPT_2)\le\omega(OPT_3)\le\omega(OPT_4)\le\omega(OPT_5)+N\alpha.\tag{6}$$

Combining (5) and (6), the proof completes. □

## 5 Distributed TGA (DTGA)

The proposed TGA in the previous section is a general centralized algorithm, which is impractical due to the decentralized nature of WSNs. In this section, we introduce a distributed truncated greedy algorithm, obtaining the same set of submodular set cover as TGA for the MWSSC problem.

At the beginning of each time slot, DGTA is executed to select a submodular set cover from all functional sensor nodes. The selected nodes are activated in the current slot, while other nodes can be turned off. Denote by $C_i$ the subset of $R_i$ that has already decided to be active, $S_i$ the subset of $R_i$ that has decided to go to sleep. Let $\vartheta_i^k=\frac{\omega_i}{\rho_i(C_i)}$ at iteration $k$. We describe DTGA briefly. Without ambiguity, we use $C_i$ instead of $C_i^k$ at iteration $k$. Initially, each sensor sets $C_i=\Phi$ and $S_i=\Phi$. At iteration $k$, each sensor $i$ calculates its own $\rho_i(C_i)$. It decides to go to sleep if $\rho_i(C_i)<\frac{\varepsilon}{N}$ and transmits a SLEEP message to nodes in $R_i$; otherwise it transmits a message including $\vartheta_i$ and $ID_i$ to nodes in $R_i$. Upon receiving the decision message from nodes in $R_i$, sensor node $i$ first adds node $j$ into $S_i$ if it receives a SLEEP message from $j$, and then compares its $\vartheta_i$ with $\vartheta_j$, $j\in R_i\setminus(C_i\cup S_i)$. If its $\vartheta_i$ is the only smallest, or if its $\vartheta_i$ is the minimum and it has a minimum ID, then it sends an ACTIVE message to the nodes in $R_i\setminus(C_i\cup S_i)$; otherwise, it keeps silent. This process continues until each node has decided to be either ACTIVE or SLEEP. We sketch DTGA in Algorithm 2.

**Algorithm 2.** Distributed TGA (DTGA)

1) $k=0$; $C_i=\Phi$, $S_i=\Phi$, $i=1,2,\ldots,N$;
2) $k=k+1$;
   a) each sensor node $i$ computes $\rho_i(C_i)$; if $\rho_i(C_i)<\frac{\varepsilon}{N}$, the node $i$ selects to go to sleep, and sends a SLEEP message to the nodes in $R_i\setminus(S_i\cup C_i)$;
   b) each sensor node $i$ receives message from nodes in $R_i\setminus(S_i\cup C_i)$; if it receives a SLEEP message from node $j$, then $S_i=S_i\cup j$;

c) each sensor node $i$ compares its $\vartheta_i^k$ with $\vartheta_j^k$, $j \in R_i \backslash (S_i \cup C_i)$; let $\underline{\vartheta}_i = \{j | \min_{j \in R_i \backslash (S_i \cup C_i)} \vartheta_j^k\}$; if node $i \in \underline{\vartheta}_i$ and it has a minimum ID in the set $\underline{\vartheta}_i$, then $i$ marks itself ACTIVE, and sends an ACTIVE message to nodes in $R_i \backslash (S_i \cup C_i)$;
d) each sensor node $i$ receives messages from nodes in $R_i \backslash (S_i \cup C_i)$, and updates its $C_i = C_i \cup j$ if it receives an ACTIVE message from node $j$;

3) if every node has marked itself either ACTIVE or SLEEP, then terminate; otherwise go to step 2).

**Remarks.** 1) At the beginning of each time slot $t$, DTGA will be executed to select a submodular set cover. Each sensor node knows which sensor node in its correlation set is selected to be active at past time slots, and the information can be automatically included for prediction. Therefore, although each sensor node $i$ has a correlation set $\mathcal{R}_i$, it only needs to communicate with nodes in its spatial correlation set $R_i$. The inclusion of temporal correlation does not bring any extra communication cost, and 2) In DTGA, each sensor node makes its decision according to local information, overcoming the drawback of global communications. Hence, DTGA is a distributed algorithm.

**Theorem 2.** *Given the same weights $\omega_i$ for $i \in N$ and data correlation model, TGA and DTGA produce the same submodular cover set.*

**Proof.** Our proof is based on [30, Theorem 3]. Let $C_1 = \{v_1, v_2, \ldots, v_m\}$ and $C_2 = \{u_1, u_2, \ldots, u_n\}$ be the submodular set cover obtained by TGA and DTGA, respectively. Denote by $\vartheta_{1,v_i}^{k_{v_i}}$ the value of $\vartheta_{v_i}$ when $v_i$ is selected by TGA at time $k_{v_i}$, and by $\vartheta_{2,u_i}^{k_{u_i}}$ the value of $\vartheta_{u_i}$ when $u_i$ is selected by DTGA at time $k_{u_i}$. To facilitate the proof, we sort the elements increasingly in $C_1$ and $C_2$ by their $\vartheta_{1,v_i}^{k_{v_i}}$ and $\vartheta_{2,u_j}^{k_{u_j}}$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$. Without loss of generality, we still use $C_1$ and $C_2$ to denote the sorted sets. We have the following two observations:

1. $\vartheta$ value of each node $i$ is an increasing function of time, i.e., $\vartheta_i^{k_1} \leq \vartheta_i^{k_2}, \leq \ldots, \vartheta_i^{k_n}$, when $k_1 \leq k_2 \leq \ldots k_n$.
2. If $u_j \in R_{u_i}$ (or $u_i \in R_{u_j}$) and $\vartheta_{2,u_i}^{k_{u_i}} < \vartheta_{2,u_j}^{k_{u_j}}$, then $u_i$ should be selected by DTGA earlier than $u_j$, i.e., $k_{u_i} < k_{u_j}$.

The first observation is valid as $\rho_{u_i}(C_i)$ is a decreasing function of time. We can get the second observation by contradiction. If $k_{u_j} \leq k_{u_i}$, by observation 1), we have $\vartheta_{2,u_i}^{k_{u_j}} \leq \vartheta_{2,u_i}^{k_{u_i}} < \vartheta_{2,u_j}^{k_{u_j}}$. Hence at time $k_{u_j}$, node $u_i$ should be selected instead of $u_j$ according to DTGA, which violates the assumption. In the following, We show $C_1 = C_2$.

We first show $C_1 = C_2$ when $C_1 \subseteq C_2$ or $C_2 \subseteq C_1$. If $C_2 \subseteq C_1$, TGA will terminate when it obtains $C_2$ as $C_2$ is already a $\varepsilon$-approximation submodular set cover. Hence, $C_1 = C_2$ in this case. When $C_1 \subseteq C_2$, consider the time $k_{u_{m+1}}$ when DTGA selects the node $u_{m+1}$. As $v_1 = u_1, v_2 = u_2, \ldots, v_m = u_m$, node $u_{m+1}$ has the same set of $C_{u_{m+1}}$ at $k_{u_{m+1}}$ by DTGA as that by TGA, thus $\vartheta_{u_{m+1}}$ under two algorithms are the same. According to TGA, the algorithm terminates only when $\rho_j(C^m) < \frac{\varepsilon}{N}$, $j \in N \setminus C^m$. Therefore, node $u_{m+1}$ should not be selected in the set $C_2$, which violates the fact $u_{m+1} \in C_2$. Hence, $C_1 = C_2$.

Let $j$ be the first index that satisfies $v_j \neq u_j$. We now show $C_1 = C_2$ when $C_1 \not\subseteq C_2$ and $C_2 \not\subseteq C_1$, by demonstrating that $\vartheta_{v_j} > \vartheta_{u_j}$ or $\vartheta_{u_j} > \vartheta_{v_j}$ cannot hold. As $v_l = u_l$, $l = 1, 2, \ldots, j-1$, $\vartheta_{1,v_j}^{k_{v_j}} = \vartheta_{2,v_j}^{k_{u_j}}$. TGA should select $u_j$ instead of $v_j$ when $\vartheta_{v_j} > \vartheta_{u_j}$ as TGA is a centralized algorithm. Therefore, $\vartheta_{v_j} > \vartheta_{u_j}$ cannot hold and we only consider $\vartheta_{v_j} < \vartheta_{u_j}$. If $v_j \in R_{u_j}$, in the DTGA, $v_j$ should be selected earlier than $u_j$, violating the fact that $u_j$ is selected earlier than $v_j$. So $v_j \notin R_{u_j}$. In TGA, at time $k_{v_j}$, $\rho_{v_j}(C^{k_{v_j}})$ is larger than $\frac{\omega}{N}$ as it is not selected by the algorithm before $k_{v_j}$. In DTGA, at least one sensor node should be selected to make $\rho_{v_j} < \frac{\omega}{N}$. We note that $v_j$ cannot be selected. This is because $\vartheta_{2,v_j} < \vartheta_{2,u_j}$ and $C_2$ is increasingly sorted by $\vartheta$. Assume node $u_l$ is the first node in $R_{v_j}$ selected by DTGA at time $k_{u_l}$, and thus $\vartheta_{2,u_l}^{k_{u_l}} < \vartheta_{2,v_j}^{k_{u_l}}$. Thus, we have

$$\vartheta_{1,v_j}^{k_{v_j}} = \vartheta_{2,v_j}^{k_{u_j}} = \vartheta_{2,v_j}^{k_{u_l}} > \vartheta_{2,u_l}^{k_{u_l}} \geq \vartheta_{2,u_j}^{k_{u_j}},$$

which violating the assumption $\vartheta_{1,v_j} < \vartheta_{2,u_j}$.

Therefore, we get $\vartheta_{v_j} = \vartheta_{u_j}$, which correspondingly leads to the conclusion of $v_j = u_j$. The proof completes. □

## 6 Implementation Issues

In this section, we discuss the implementation issues of DTGA, which follows the procedure: 1. model data correlation and perform prediction, 2. detect the termination of network lifetime, 3. guarantee network connectivity, and 4. discuss about the communication cost of DTGA.

### 6.1 Correlation Modeling and Prediction

We discuss how to model data correlation and perform prediction for inactive sensor nodes, based on which the conditional entropy function is derived.

*Gaussian process* is widely applied for model selection and prediction [31]. It is fully decided by a mean function $M(x)$ and a covariance function $K(x, x')$ (See [13] for large collection of existing literatures on covariance functions).

When a Gaussian process with some hyperparameters are selected for an application, the next step is to train the model, i.e., to specify the parameters in the selected mean and covariance functions in the light of available data. The basic idea is to choose hyperparameters that best fit the available data. Similar to maximum likelihood estimation, the log marginal likelihood function, denoted by $L$, can be used to evaluate the probability of fitness [14]

$$\begin{aligned} L &= \log \Pr(\boldsymbol{f}_x | \boldsymbol{x}, \mathcal{P}) \\ &= -0.5 \log |\mathcal{K}| - 0.5 (\boldsymbol{f}_x - \boldsymbol{\mu})^T \mathcal{K}^{-1} (\boldsymbol{f}_x - \boldsymbol{\mu}) \\ &\quad - \frac{n}{2} log(2\pi), \end{aligned}$$

where $\boldsymbol{x}$ and $\boldsymbol{f}_x$ are the input variables and output values of available data, respectively, $\mathcal{P}$ are the hyperparameters, $\boldsymbol{\mu}$ is mean values of $\boldsymbol{f}_x$, and $\mathcal{K}$ is the covariance matrix of $\boldsymbol{x}$.

Since $L$ is a function of hyperparameters $\mathcal{P}$, by maximizing $L$ based on its partial derivatives, we can get the values of $\mathcal{P}$

$$\frac{\partial L}{\partial \mathcal{P}_M} = -(\boldsymbol{f}_x - \boldsymbol{\mu})^T \mathcal{K}^{-1} \frac{\partial M}{\partial \mathcal{P}_M}, \tag{7}$$

$$\frac{\partial L}{\partial \mathcal{P}_K} = \frac{1}{2} trace\left(\mathcal{K}^{-1} \frac{\partial K}{\partial \mathcal{P}_K}\right) + \frac{1}{2}(\boldsymbol{f}_x - \boldsymbol{\mu}) \frac{\partial K}{\partial \mathcal{P}_K} \mathcal{K}^{-1} \frac{\partial K}{\partial \mathcal{P}_K} (\boldsymbol{f}_x - \boldsymbol{\mu}), \tag{8}$$

where $\mathcal{P}_M$ and $\mathcal{P}_K$ denote the hyperparameters of the mean and covariance functions, respectively. Proper hyperparameters $\mathcal{P}$ can be obtained from (7) and (8), possibly with the help of a numerical optimization method, e.g., conjugate gradients.

We now show how to perform prediction by Gaussian process. Denote $f_D$ the function values of the known variable set $\boldsymbol{x}_D$, and $f_*$ the values of unknown variables $\boldsymbol{x}_*$. A good property of Gaussian process is that the posterior distribution for a specific set of unknown variables conditioned on a given variable set is still a Gaussian process [14]

$$f_* | x_D = \mathcal{GP}(M_*, K_*), \tag{9}$$

where

$$M_*(x) = M(x) + K(x_D, x)^T \mathcal{K}^{-1}(f_D - M_D), \tag{10}$$

$$K_*(x, x') = K(x, x') - K(x_D, x)^T \mathcal{K}^{-1} K(x_D, x), \tag{11}$$

where $K(x_D, x)$ is a vector of covariances between every variable in $x_D$ and $x$, $\mathcal{K}$ is the covariance matrix of $x_D$, and $M_D$ is the mean vector of $x_D$.

**Remarks.** 1) Equations (10) and (11) are central for Gaussian process predictions. Note that posterior variance function $K_*(x, x')$ is smaller than prior variance function $K(x, x')$ because $K_*(x, x')$ equals to $K(x, x')$ minus a positive term. This means we can reduce the uncertainty of unknown data by using available data. 2) The conditional entropy $H(x|x_D) = \rho_{x_D}(x) = \frac{1}{2}\log(2\pi e K_*(x, x))$ [11]. Therefore, by using Gaussian process, the conditional entropy $\rho_j(A)$ in Algorithms 1 and 2 is very easy to calculate.

### 6.2 Network Lifetime

At initiation, all the sensor nodes are turned on. They sense information from the geographic RoI and route it through a multihop path to the sink node, where all the collected data are used to train a Gaussian process. The sink node broadcasts the Gaussian process model to the network, which is used by each sensor node to build its own $R_i$. Then, each sensor node sends its location information and $ID$ to the nodes in $R_i$. A synchronization technique, e.g., [32], is adopted to synchronize the entire network, and wake up the sleep node in the current time slot at the beginning of next time slot. At the beginning of each time slot, DTGA is executed to activate a subset of sensor nodes until the network lifetime is declared to be terminated.

According to the definition of network lifetime, the network terminates when one sensor node runs out of energy. With the operation of the network, some nodes fail due to factors such as hardware failure or energy depletion. The network can be considered to be working as long as the remaining sensor nodes can guarantee no information loss about RoI. We adopt the most frequently used definition of network lifetime to show a theoretical lower bound of network operation time. Under this definition, each sensor can detect the termination of network when at least one of its neighbors is dead.

### 6.3 Network Connectivity

By using a process similar to [2], the submodular set cover selected by DTGA is also a connected cover when $R_t \geq 2\overline{R}$. However, this cannot always hold in practice. When $\overline{R}$ is much larger than $R_t$, connectivity cannot be guaranteed.

There are two approaches to deal with the connectivity issue. First, each sensor node stores its collected data in its memory and transmits the data to the sink node at the beginning of next time slot. Since all sensor nodes will be activated to execute DTGA at this moment, the network is connected. The second approach is to activate more sensor nodes as relay nodes to ensure connectivity. Approximation algorithms for *Steiner tree problem* [33] can be introduced to select relay nodes, after DTGA is executed to get a submodular set cover. The first approach can avoid selecting more sensor nodes, while the second one can transmit data to the sink node with less delay. With respect to application requirement, one approach can be selected to address the connectivity problem.

### 6.4 Communication Cost

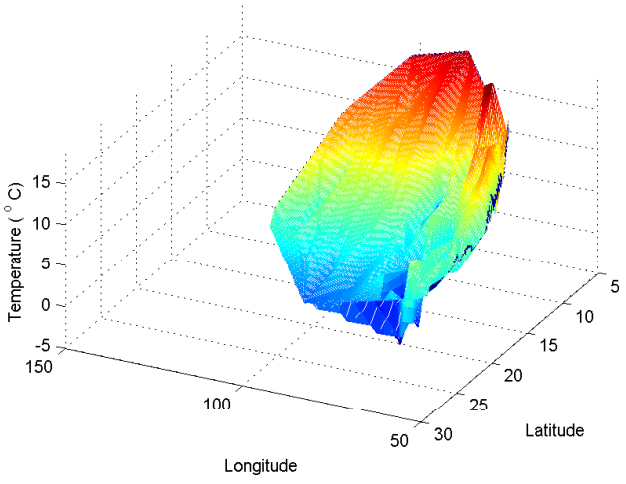
The communication cost for each sensor node $i$ during the execution of DTGA at the beginning of each time slot depends on the cardinality of correlation set $R_i$. When the correlation diameter $\overline{R}_i$ of sensor node $i$ is large, $i$ has more chance to go to sleep, thus longer lifetime. However, this gives arise to huge communication cost needed to execute DTGA, since nodes in $R_i$ can be node $i$'s $k(k \geq 2)$-hops neighbors. At the beginning of each time slot, each sensor node $i$ has to exchange messages with nodes in $R_i$ through multiple hops. Therefore, the correlation set $R_i$ for each node $i$ should be carefully determined in order to trade-off between longer lifetime chance and larger communication cost.

There are two ways to determine the correlation set for each sensor node $i$. First, it is to use correlation coefficient. In GP, the correlation coefficient between sensor nodes $i$ and $j$, denoted by $c_{ij}$, can be determined from the covariance function $K(\cdot, \cdot)$. Using $c_{ij}$, node $i$ can decide whether $j$ is in its correlation set $R_i$: $j$ is in $R_i$ if $c_{ij} \geq th$, where $th$ is a constant threshold. The other way is to use spatial distance to decide if a sensor node $j$ is in the correlation set of sensor node $i$. For example, we can use $\overline{R} = 2R_t$ as a threshold, within which two sensors are considered highly correlated.

## 7 Performance Evaluation

In this section, we use real data collected from US National Climatic Data Center [15] to perform data modeling and algorithm evaluation. We also conduct extensive simulations to evaluate the algorithm performance under various parameter settings.


(a) Location of the weather stations in USA.




(b) The average temperature at each station.

Fig. 1. Real weather data of US in 2008.

### 7.1 Evaluation Using Real Data

We use the average temperature of each day in year 2008 collected from 224 weather stations in US to test data modeling and prediction in DTGA. The location of these stations are marked in Fig. 1a (both blue and red squares). Without loss of generality, we transform latitudes and longitudes of the stations into intervals $[0, 90°]$ and $[0,\ 180°]$ by subtracting two constants, respectively. We get the monthly average temperature of each station to train the correlation model, and the annually average temperatures are depicted in Fig. 1b.

To facilitate data modeling, we need to find proper mean and covariance functions to specify a Gaussian process. From Fig. 1b, we can see that the annually average temperature at each weather station is roughly a decreasing function of the latitude. We adopt the mean function as follows:

$$M(\boldsymbol{x}) = (a_1 t^2 + a_2 t + a_3)(a_4 x_1 + a_5 x_2 + a_6), \qquad (12)$$

where $\boldsymbol{x} = (t, x_1, x_2)$, $x_1$ and $x_2$ correspond to the longitude and latitude of a weather station. For covariance function, we use rational quadratic covariance function to model the temporal as well as spatial correlations [13], i.e.,

$$K(\boldsymbol{x}', \boldsymbol{x}'') = \lambda^2 \left(1 + \frac{d_1^2(t', t'')}{2\phi\eta}\right)^{-\phi} \left(1 + \frac{d_2^2}{2\gamma\beta}\right)^{-\gamma}, \qquad (13)$$

where $d_1(t', t'') = |t' - t''|$, and $d_2$ is the shorter surface distance between two points on the earth.

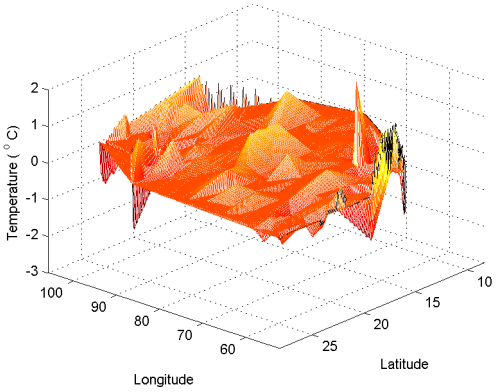



Fig. 2. Prediction error at each location by using DTGA.

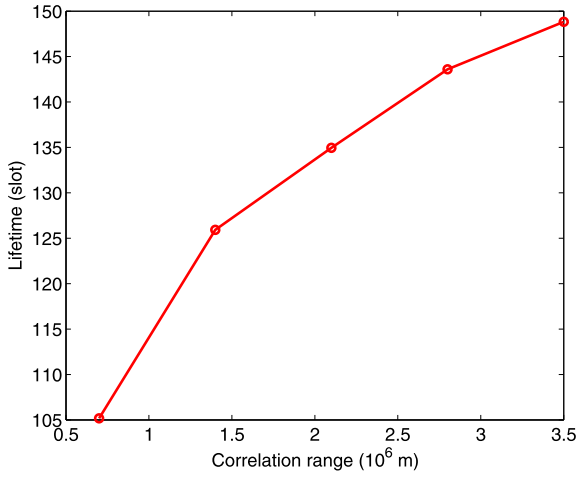
We use the training process described in Section 6.1 to get the prior Gaussian process. The obtained parameters of mean and covariance functions are $a_1 = 0.0398$, $a_2 = -0.2393$, $a_3 = 2.9282$, $a_4 = -9.6994$, $a_5 = -0.8006$, $a_6 = 5.7541$, $\lambda = 7.2581$, $\eta = 2.1238$, $\phi = 0.1602$, $\beta = 0.0980$, and $\gamma = 0.0266$. Codes about the training process in different programming languages can be obtained in [34].

We proceed to validate the efficiency of our proposed DTGA algorithm. To do this, we regard each weather station as a sensor node, from which we have a wireless sensor network with a temporal and spatial correlation model. We set the transmission range between two sensor nodes to be $R_t = 0.7 \times 10^6$ m. We find that temporal and spatial correlations between sensor nodes are very high. Hence, we adopt the second approach discussed in Section 6, i.e., use spatial distance, to decide the correlation set for each sensor node. Hereafter, unless otherwise stated, we set $\overline{R}_{tc} = 3, \overline{R}_i = 2R_t, \forall i \in N$, to trade-off the benefits of correlations and communication cost.

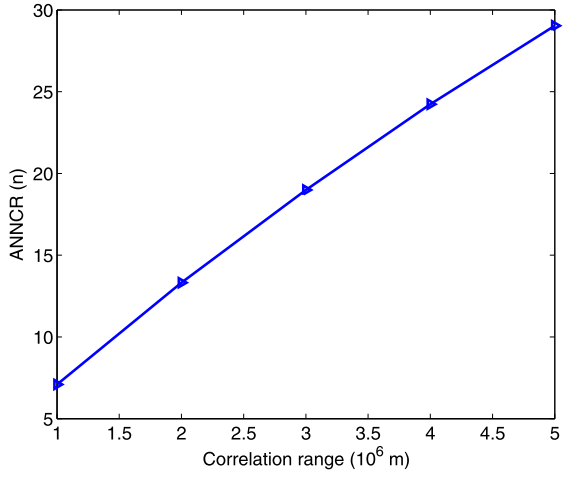
We run DTGA within MATLAB, using real data to validate the efficiency of DTGA algorithm. Since we have obtained monthly average temperature at each weather location, we can regard each month as a time slot. At the beginning of each time slot, DTGA is executed, and the selected sensor nodes are activated in the current slot. We calculate prediction values of inactive sensor nodes by using the data of active sensor nodes in current and last three slots (when $t = 1$, only data of current active nodes are used). More than one third of sensor nodes in the experiments can be turned off to save energy without degrading the sensing performance at each time slot. Due to space limitation, we only take the algorithm results in the forth slot (the first slot that can exploit data correlations of previous three slots and current slot). In Fig. 1a, the active sensor nodes (weather stations) are marked in red squares and inactive sensor nodes in blue squares. The corresponding prediction errors are plotted in Fig. 2, from which we can see that the prediction error at each inactive sensor node is very small. Although a large number of sensor nodes are turned off, the sensing quality of the network is almost not degraded. The average prediction errors at each slot are summarized in Table 1. Obviously, the average prediction error at any slot (1-12) of each inactive sensor node is less than 0.75. All these validate the efficiency of DTGA.

TABLE 1
Average Prediction Error of Each Month by DTGA

| Month | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Prediction error | 0.4469 | 0.5931 | 0.5606 | 0.5131 | 0.6930 | 0.6160 |
| Month | 7 | 8 | 9 | 10 | 11 | 12 |
| Prediction error | 0.6358 | 0.5334 | 0.5214 | 0.7479 | 0.7109 | 0.5572 |

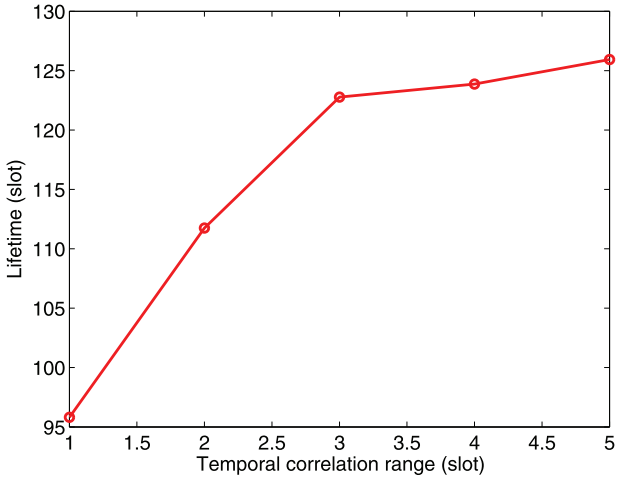



(a) Network lifetime Vs. spatial correlation range.

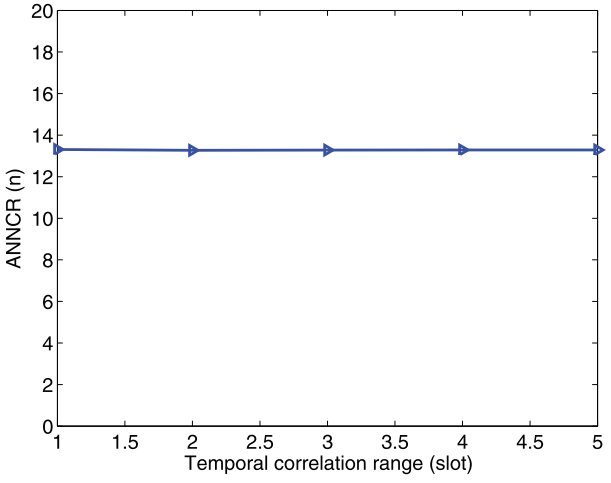



(b) ANNCR Vs. spatial correlation range.

Fig. 3. Network lifetime and ANNCR under different spatial correlation ranges.




(a) Network lifetime Vs. temporal correlation range.




(b) ANNCR Vs. temporal correlation range.

Fig. 4. Network lifetime and ANNCR under different temporal correlation ranges.

## 7.2 Evaluation through Simulation

In the previous section, we evaluated the algorithm performance using real data with fixed parameters. In this section, we shall demonstrate the algorithm performance over different network settings and correlation parameters.

Let $n = 100$ sensor nodes be deployed to a $[0, 50] \times [0, 50]$ (longitude $\times$ latitude) area according to uniform distribution. We set initial energy of each sensor node to be 50 slots. One slot energy will be consumed if a sensor node is active at one time slot. We use the first strategy for connectivity mentioned in Section 6. The constant $\kappa$ in the weight $\omega_i$ is set to be $e$, i.e., the natural base. We use the same mean and covariance functions as specified in (12) and (13), but with different values of parameters. First, we vary the spatial correlation range $\overline{R}_{sc}$ to be $R_t$, $2R_t$, $3R_t$, $4R_t$, and $5R_t$, meanwhile other parameters are fixed. Other simulation settings are the same as those in Section 7.1. We run the simulations for each setting 100 times to eliminate the random deviation, and the corresponding network lifetimes are plotted in Fig. 3a. From this figure, we see that the network lifetime increases with the spatial correlation range. This is because as the correlation range becomes larger, each sensor node in the network will have more correlation neighbors (see Fig. 3b), thus more chance to be inactive. However, as aforementioned in Section 6, the communication cost will also increase due to increased Average Number of Nodes in the spatial Correlation set under different Ranges (ANNCR). Therefore, in practice, the correlation range should be carefully chosen to trade-off the benefit of correlation and communication cost.

We run simulations for different $\overline{R}_{tc}$. Similarly, we vary $\overline{R}_{tc}$ from 1 to 5 and obtain the corresponding network lifetime and ANNCR, which are plotted in Fig. 4. With the increase of temporal correlation range $\overline{R}_{tc}$, the network lifetime increases. This is reasonable, as the larger the temporal correlation range, the more data that can be exploited for prediction. Note that at the same time

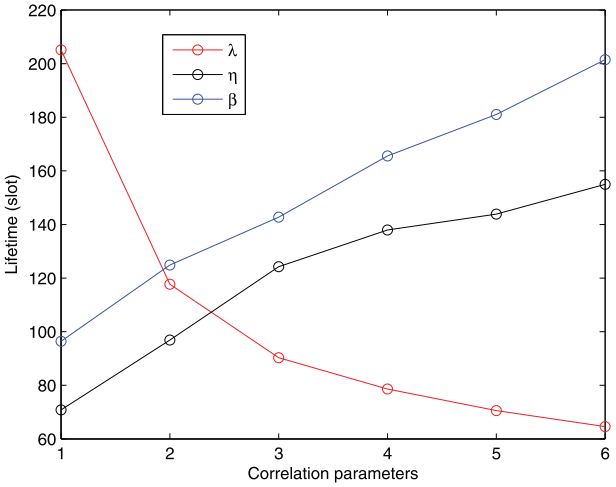



Fig. 5. Network lifetime under different values of $\lambda$, $\eta$, and $\beta$.

the average number of neighbors does not increase. Hence, it is more effective to exploit temporal correlation to prolong the network lifetime than spatial correlation, as there is less communication cost.

We vary $\lambda$, $\eta$, and $\beta$ from 1 to 5, in the covariance function (13), respectively, and perform simulations 100 times for each case to get the average network lifetime. The results are plotted in Fig. 5. We can see that the network lifetime decreases exponentially when parameter $\lambda$ increases linearly. As $\lambda$ is the standard deviation of each variance in Gaussian process, the larger the $\lambda$, the more sensor nodes are needed to predict the temperatures of an inactive node, thus less network lifetime. In addition, the network lifetime is an increasing function of parameters $\eta$ and $\beta$. This is due to the fact that parameters $\eta$ and $\beta$ indicate to what extent the correlation is between two sensors in the rational quadratic covariance function.

To demonstrate the advantages of our proposed algorithm, we compare the network lifetime obtained by DTGA with that of existing approaches. To the best of our knowledge, [1] is the only one related to our work. In [1], a linear predictive model is proposed to model the data correlation. Due to the limitation of linear predictive model, it is very difficult to model the temporal correlation. As mentioned in [14], using Gaussian process for prediction can be thought as a "generalized linear regression" (GLR), thus can obtain better results than linear predictive model does. We denote algorithm of using Gaussian process for prediction without exploiting temporal correlation by GLR, and compare DTGA with GLR. If the performance obtained by DTGA is better than GRL, it is surely better than the linear predictive model in [1]. To give a fair comparison, we run two algorithms under the same network setting. The results are plotted in Fig. 6. We can see that the network lifetime under DTGA is about 160 time slots, almost two times of the network lifetime of GLR. Therefore, by exploiting temporal correlation, network lifetime can be significantly lengthened. We plot the total number of active sensor nodes at each slot in Fig. 6. Obviously, DTGA obtains a much less number of active sensors at each slot, which is the main reason that DTGA can yield a much longer network lifetime than GRL.

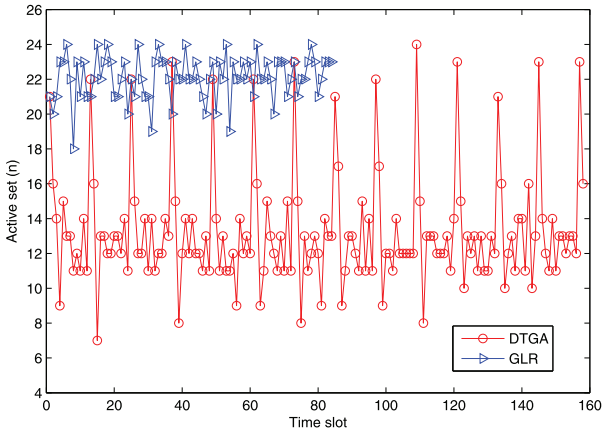



Fig. 6. The number of active sets selected by DTGA and GLR at each time slot.

## 8 Conclusion

We have studied the coverage problem in wireless sensor networks. As there are typically temporal and spatial correlations among the data sensed by different sensor nodes, we exploit such data correlations and leverage prediction to prolong the network lifetime. The issue has been formulated as a *minimum weight submodular set cover* problem. We proposed a truncated greedy algorithm with a theoretical performance guarantee to solve it. We modified TGA into a distributed algorithm, DTGA, and proved that these two algorithms obtain the same set cover. The implementation issues such as network connectivity and communication cost are extensively discussed. Real data experiments as well as simulations were conducted to show the advantage of DTGA over existing generalized linear regression algorithms and evaluate the impacts of different parameters of covariance function on the network lifetime.

## Acknowledgments

This work was supported in part by the NSFC under Grants 61028007, 61004060, and 60974122, the Specialized Research Fund for the Doctoral Program of China under Grant 20100101110066, and the Natural Science Foundation of Zhejiang Province under Grant R1100324.

## References

[1] H. Gupta, V. Navda, S.D. Das, and V. Chowdhary, "Efficient Gathering of Correlated Data in Sensor Networks," *Proc. ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2005.

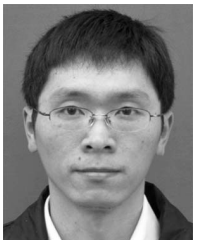
[2] H. Zhang and J. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," *Ad hoc and Sensor Wireless Networks*, vol. 1, nos. 1/2, pp. 89-124, 2005.

[3] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks," *ACM Trans. Sensor Networks*, vol. 1, no. 1, pp. 36-72, 2005.

[4] J. Chen, W. Xu, S. He, Y. Sun, P. Thulasiramanz, and X. Shen, "Utility-Based Asynchronous Flow Control Algorithm for Wireless Sensor Networks," *IEEE J. Selected Areas in Comm.*, vol. 28, no. 7, pp. 1116-1126, Sept. 2010.

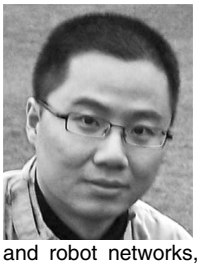
[5] R. Lu, X. Lin, H. Zhu, X. Liang, and X. Shen, "Becan: A Bandwidth-Efficient Cooperative Authentication Scheme for Filtering Injected False Data in Wireless Sensor Networks," *IEEE Trans. Parallel and Distributed Systems*, To Appear.

[6] H. Tan, I. Korpeoglu, and I. Stojmenovic, "Computing Localized Power Efficient Data Aggregation Trees for Sensor Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 22, no. 3, pp. 489-500, Mar. 2011.
[7] X. Li, H. Frey, N. Santoro, and I. Stojmenovic, "Strictly Localized Sensor Self-Deployment for Optimal Focused Coverage," *IEEE Trans. Mobile Computing,* To Appear, 2011.
[8] P. Balister, Z. Zheng, S. Kumar, and P. Sinha, "Trap Coverage: Allowing Coverage Holes of Bounded Diameter in Wireless SensorNetworks ," *Proc. IEEE INFOCOM,* 2009.
[9] S. He, J. Chen, D.K.Y. Yau, H. Shao, and Y. Sun, "Energy-Efficient Capture of Stochastic Events by Global- and Local-Periodic Network Coverage," *Proc. the Sixth ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc),* 2009.
[10] N.A. Cressie, *Statistics for Spatial Data.* Wiley, 1991.
[11] A. Krause and C. Guestrin, *A Note on the Budgeted Maximization on Submodular Functions,* Technical Report. Carnegie Mellon Univ. 2005.
[12] L. Wolsey, "An Analysis of the Greedy Algorithm for the Submodular Set Covering Problem," *Combinatorica,* vol. 2, no. 4, pp. 385-393, 1981.
[13] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning.* Massachusetts Inst. of Technology Press, 2006.
[14] C. Williams, "Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond," Technical Report NCRG/97/012, 1997.
[15] *Nat'l Climatic Data Center,* www.ncdc.noaa.gov/cgi-bin/res40.pl?page=gsod.html, 2011.
[16] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic, "Localized Sensor Area Coverage with Low Communication Overhead," *IEEE Trans. Mobile Computing,* vol. 7, no. 5, pp. 661-672, May 2008.
[17] R. Kershner, "The Number of Circles Covering a Set," *Am. J. Math.,* vol. 61, pp. 665-671, 1939.
[18] X. Bai, C. Zhang, and D. Xuan, "Deploying Wireless Sensors to Achieve both Coverage and Connectivity," *Proc. ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc),* 2006.
[19] X. Bai, Z. Yun, D. Xuan, T. Lai, and W. Jia, "Deploying Four-Connectivity and Full-Coverage Wireless SensorNetworks ," *Proc. IEEE INFOCOM,* 2008.
[20] D. Tian and N. Georganas, "A Coverage Preserving Node Scheduling Scheme for Large Wireless SensorNetworks," *Proc. Int'l Workshop Wireless Sensor Networks and Applications,* 2002.
[21] T. Dam and K. Langendoen, "An Adaptive Energy-Efficient Mac Protocol for Wireless Sensor Networks," *Proc. Int'l Conf. Embedded Networked Sensor Systems (SenSys),* 2003.
[22] S. Yang, F. Dai, M. Cardei, J. Wu, and F. Patterson, "On Connected Multiple Point Coverage in Wireless Sensor Networks," *Int'l J. Wireless Information Networks,* vol. 13, no. 4, pp. 289-301, 2006.
[23] A. Keshavarzian, H. Lee, and L. Venkatraman, "Wakeup Scheduling in Wireless SensorNetworks," *Proc. Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc),* 2006.
[24] C. Hsin and M. Liu, "Network Coverage Using Low Duty-Cycled Sensors: Random and Coordinated Sleep Algorithms," *Proc. the Int'l Symp. Information Processing in Sensor Networks (IPSN),* 2004.
[25] B. Wang, K. Chua, V. Srinivasan, and W. Wang, "Information Coverage in Randomly Deployed Wireless Sensor Networks," *IEEE Trans. Wireless Comm.,* vol. 6, no. 8, pp. 2994-3004, Aug. 2007.
[26] G. Yang and D. Qiao, "Barrier Information Coverage with Wireless Sensors," *Proc. IEEE INFOCOM,* 2009.
[27] G. Xing, R. Tan, B. Liu, J. Wang, and C. Yi, "Data Fusion Improves the Coverage of Wireless Sensor Networks," *Proc. MobiCom,* 2009.
[28] R. Ghrist and A. Muhammad, "Coverage and Hole-Detection in Sensor Networks via Homology," *Proc. the Int'l Symp. Information Processing in Sensor Networks (IPSN),* 2005.
[29] K. Yedavalli and B. Krishnamachari, "Sequence-Based Localization in Wireless Sensor Networks," *IEEE Trans. Mobile Computing,* vol. 7, no. 1, pp. 1-14, Jan. 2008.
[30] G. Kasbekar, Y. Bejerano, and S. Sarkar, "Lifetime and Coverage Guarantees through Distributed Coordinate-Free Sensor Activation," *Proc. MobiCom,* 2009.
[31] R. Garnett, M. Osborne, and S. Roberts, "Bayesian Optimization for Sensor Set Selection," *Proc. the Int'l Symp. Information Processing in Sensor Networks (IPSN),* 2010.
[32] S. Ganeriwal, R. Kumar, and M.B. Srivastava, "Timing-Sync Protocol for SensorNetworks," *Proc. Int'l Conf. Embedded Networked Sensor Systems (SenSys),* 2003.
[33] H. Takahashi and A. Matsuyama, "An Approximate Solution for the Steiner Problem in Graphs," *Math. Japonica,* vol. 24, no. 6, pp. 573-577, 1980.
[34] http://www.gaussianprocess.org, 2011.



**Shibo He** is currently working toward the PhD degree in control science and engineering at Zhejiang University, Hangzhou, China. He is a member of the Group of Networked Sensing and Control (IIPC-nesC) in the State Key Laboratory of Industrial Control Technology at Zhejiang University. His research interests include coverage, cross-layer optimization, and distributed algorithm design problems in wireless sensor networks.

**Jiming Chen** (M'08-SM'11) received the BSc and PhD degrees both in control science and engineering from Zhejiang University in 2000 and 2005, respectively. He was a visiting researcher at Inria in 2006, National University of Singapore in 2007, University of Waterloo from 2008 to 2010. Currently, he is a full professor with Department of control science and engineering, and the coordinator of group of Networked Sensing and Control in the State Key laboratory of Industrial Control Technology at Zhejiang University, China. His research interests are estimation and control over sensor network, sensor and actuator network, target tracking in sensor networks, optimization in mobile sensor network. He has published more than 50 peer-reviewed papers. He currently servers as associate editors for several international Journals. He is a guest editor of *IEEE Transactions on Automatic Control*, *Computer Communication* (Elsevier), *Wireless Communication and Mobile Computer*(Wiley), and *Journal of Network and Computer Applications* (Elsevier). He also serves as a cochair for Ad hoc and Sensor Network Symposium, IEEE Globecom 2011, general symposia CoChair of ACM IWCMC 2009 and ACM IWCMC 2010, WiCON 2010 MAC track CoChair, IEEE MASS 2011 Publicity CoChair, IEEE DCOSS 2011 Publicity CoChair, and TPC member for IEEE ICDCS 2010, IEEE MASS 2010, IEEE SECON 2011, IEEE INFOCOM 2011, IEEE INFOCOM 2012, etc. He is a senior member of the IEEE.



**Xu Li** received the bachelor's degree from Jilin University, China in 1998, the master's degree from the University of Ottawa, Canada in 2005, and the PhD degree from Carleton University, Canada in 2008, all in computer science. He held postdoc positions at the University of Waterloo, Canada, the University of Ottawa, Canada, CNRS, France and Inria, France, where he is currently a research officer. His research interests are in wireless ad hoc, sensor and robot networks, topology control, data communications, mobility management, network security, and QoS provisioning. He is on the editorial board of the *European Transactions on Telecommunications*, *Ad Hoc & Sensor Wireless Networks: an International Journal, Parallel and Distributed Computing and Networks.* He is a guest editor of *Computer Communications* (2011) and *Journal of Communications* (2012). He is/was in different chairing positions or among the technical program committees for many conferences and workshops, e.g., AdHoc-NOW'08-11, IEEE DCOSS'11, IEEE MASS'07&11, IEEE LCN'10&11, IEEE PIMRC'09&11, etc. He was a recipient of NSERC postdoctoral fellowship awards and a number of other awards.

**Xuemin (Sherman) Shen** (M'97-SM'02-F'09) received the BSc (1982) degree from Dalian Maritime University (China) and the MSc (1987) and PhD (1990) degrees from Rutgers University, New Jersey, all in electrical engineering. He is a professor and University research chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on resource management in interconnected wireless/wired networks, UWB wireless communications networks, wireless network security, wireless body area networks, and vehicular ad hoc and sensor networks. He is a coauthor of three books, and has published more than 400 papers and book chapters in wireless communications and networks, control and filtering. He served as the Technical Program Committee chair for IEEE VTC10, the Symposia Chair for IEEE ICC10, the tutorial chair for IEEE ICC08, the Technical Program Committee chair for IEEE Globecom07, the General CoChair for Chinacom07 and QShine06, the founding chair for IEEE Communications Society Technical Committee on P2P Communications and Networking. He also served as a founding area editor for *IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS*, editor-in-chief for *Peer-to-Peer Networking and Application*, associate editor for *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, *Computer Networks; and ACM/Wireless Networks*, etc., and the guest editor for *IEEE JSAC*, *IEEE Wireless Communications,* I*EEE Communications Magazine*, and *ACM Mobile Networks and Applications*, etc. He received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004 and 2008 from the University of Waterloo, the Premiers Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. He is a registered professional engineer of Ontario, Canada, an IEEE fellow, an Engineering Institute of Canada Fellow, and a Distinguished lecturer of IEEE Communications Society.

**Youxian Sun** received the diploma from the Department of Chemical Engineering, Zhejiang University, China, in 1964. He joined the Department of Chemical Engineering, Zhejiang University, in 1964. From 1984 to1987, he was an Alexander Von Humboldt research fellow and visiting associate professor at University of Stuttgart, Germany. He has been a full professor at Zhejiang University since 1988. In 1995, he was elevated to an Academician of the Chinese Academy of Engineering. His current research interests include modeling, control, and optimization of complex systems, and robust control design and its application. He is author/coauthor of 450 journal and conference papers. He is currently the director of the Institute of Industrial Process Control and the National Engineering Research Center of Industrial Automation, Zhejiang University. He is the president of the Chinese Association of Automation, also has served as a vice chairman of IFAC Pulp and Paper Committee and a vice president of China Instrument and Control Society.