

Research Article

Node Selection Algorithms with Data Accuracy Guarantee in Service-Oriented Wireless Sensor Networks

Hongju Cheng, Ronglie Guo, and Yuzhong Chen

College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China

Correspondence should be addressed to Hongju Cheng; cscheng@fzu.edu.cn

Received 20 December 2012; Accepted 26 February 2013

Academic Editor: Neal N. Xiong

Copyright © 2013 Hongju Cheng 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.

The service-oriented architecture is considered as a new emerging trend for the future of wireless sensor networks in which different types of sensors can be deployed in the same area to support various service requirements. The accuracy of the sensed data is one of the key criterions because it is generally a noisy version of the physical phenomenon. In this paper, we study the node selection problem with data accuracy guarantee in service-oriented wireless sensor networks. We exploit the spatial correlation between the service data and aim at selecting minimum number of nodes to provide services with data accuracy guaranteed. Firstly, we have formulated this problem into an integer nonlinear programming problem to illustrate its NP-hard property. Secondly, we have proposed two heuristic algorithms, namely, Separate Selection Algorithm (SSA) and Combined Selection Algorithm (CSA). The SSA is designed to select nodes for each service in a separate way, and the CSA is designed to select nodes according to their contribution increment. Finally, we compare the performance of the proposed algorithms with extended simulations. The results show that CSA has better performance compared with SSA.

1. Introduction

Sensing is considered as one of the most important technologies especially in the emerging big data era. Nodes with sensing ability can be deployed everywhere in the world including airspace, ground, and underwater environment due to their cheapness, simplicity, and small size. Moreover, the wireless radio allows these sensor nodes to be organized into a network, which is generally named as Wireless Sensor Networks (WSN) [1], and local information about the environment is then sensed and reported to the base station in a periodical manner. It is obvious that the wireless sensor networks will create a huge number of data with time ongoing and the number of network increasing. Accordingly, one challenging issue is how to utilize these various wireless sensor networks in the future big data era.

The current wireless sensor networks are generally data-centric or application-centric, which means that each sensor node serves for one special application. However, with the number of different applications increasing rapidly, heterogeneous wireless sensor networks appear and they might be

located in the same physical areas providing different data-collection functions. How to connect these heterogeneous wireless sensor networks efficiently is still a pioneering work in the future ubiquitous computing environment due to several observations. Firstly, two different applications may be interested in the same collected data, and it is unnecessary to place two separate nodes with identical sensing devices but different tasks. Secondary, in case that one application is concerned with several different types of sensed data simultaneously, such a requirement is not guaranteed since current solutions work only in a separate way. Finally, the emergence of powerful sensors, which can provide different types of data sensing, has introduced new issues in the research of wireless sensor networks since they can support different applications simultaneously.

Accordingly, the service-oriented architecture appears as a new emerging trend for the future of wireless sensor networks, in which sensor networks are considered as the services provider and sensors as the data sources for these services [2]. Users and programmers can access the service-oriented wireless sensor networks by using a simple

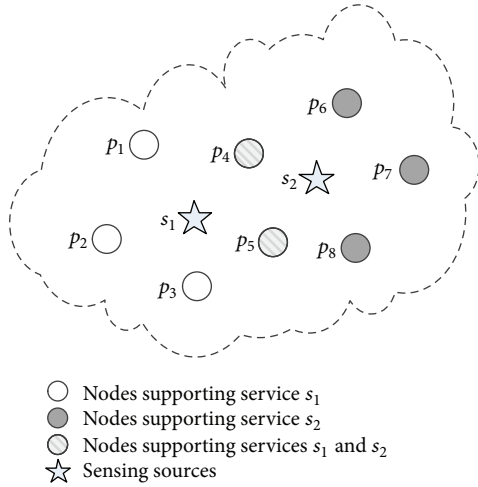


FIGURE 1: An example of a service-oriented wireless sensor network in which service s_1 is supported by nodes p_1 , p_2 , p_3 , p_4 and p_5 , service s_2 by nodes p_4 , p_5 , p_6 , p_7 and p_8 .

service-oriented interface and utilizing encapsulation of the low-level implement details. Services in wireless sensor networks may be the sensing capabilities for example, temperature and humidity or software components provided by nodes, for example, the operations of in-network data aggregation, time synchronization, and data processing [2–5]. The data sensing of nodes can also be defined as the sensing services, and sensed data as the service data similarly. Furthermore, the sensor nodes can be equipped with multiple types of sensing units used to collect environmental information. For example, the MICA2 mote [6] can provide services such as light, sound, and vibration.

In the heterogeneous applications scenarios, different types of sensors can be deployed in the same area to support various service requirements. Figure 1 shows an example of a service-oriented wireless sensor network including eight sensor nodes. The sensed sources are assumed to be located at positions s_1 and s_2 , and each sensed target is assumed one different service. Nodes p_1 , p_2 , and p_3 can only support service s_1 , and nodes p_6 , p_7 , and p_8 support service s_2 . However, nodes p_4 and p_5 can support service s_1 and s_2 simultaneously. In this example, there are several ways to select nodes while providing the services s_1 and s_2 by assuming that at least two nodes are required for each service, that is, nodes p_1 , p_2 for service s_1 and nodes p_6 , p_7 for s_2 , or nodes p_4 , p_5 for both of them simultaneously. Note that sensor networks are generally deployed in dense manner and a huge number of nodes can be provided to collect data from the environment. It leads to the problem of how to select nodes in an efficient way to provide the required services.

The accuracy of the sensed data is one of the key criterions while choosing nodes to provide required service. It is also known that sensor nodes are generally designed under the guideline of cheapness and simplicity and that they are mostly deployed in a dynamic and rough terrain for continuous environmental monitoring. The sensing equipment on sensor

nodes is expected to be unreliable and the collected data to be distorted. Furthermore, some physical attributes exhibit a gradual and continuous variation over the two-dimensional Euclidean space due to the diffusion property and, thus, each observer has different distorted data. To collect all related data around the same sensing sources can help to eliminate or minimize the distortion. However, it might lead to heavy energy consumption in the network. Some applications only are concerned with approximate observations rather than exact results [7, 8], and it is unnecessary to gather data from all nodes in the network in this case. It shall also be mentioned that the diffusion property of physical attributes results in spatial correlation among sensed data observed by nodes closer to the same sensing source. Exploiting the spatial correlation can help to improve the network performance by selecting a subset of nodes to provide the required service with data accuracy guaranteed [9, 10].

Although the service-oriented architecture for the wireless sensor networks has already been introduced recently in related works [4, 5, 11–14], most of them are concerned with the practical architecture framework, such as middleware and platforms. Some works [13, 14] considered the node scheduling to support services query with network lifetime prolonged, and some others are concerned with the spatial correlation among the sensed data with the service unconsidered [15–17]. To provide accurate service to users is an important issue and it is generally one criterion for the applications. In this paper, we focus on the heterogeneous service supporting problem in the future wireless sensor networks and aim at providing efficient node selection algorithms with data accuracy guarantee for the service-oriented sensor networks. Different from previous works, we consider the data accuracy for services by following the observation that sensed data is a noisy version of the physical phenomenon. And we have explored the data accuracy according to their spatial correlation for the same sensing sources.

Due to the inaccuracy and spatial correlation of the sensed data, it is a new and challenging issue to provide the required services with data accuracy guarantee in an *energy-efficient* way for the service-oriented wireless sensor networks. So far, as we know, this is the first paper concerned with both the data inaccuracy and spatial correlation in the sensor networks, and we aim at providing node selection algorithms so as to improve the network performance. The main contributions of this paper are summarized as follows. (1) We have proposed the node selection problem with data accuracy guarantee for service-oriented wireless sensor networks via bipartite graph and formulated it as an integer nonlinear programming problem to illustrate its NP-hard property. (2) We have also presented two efficient heuristic algorithms for this problem; namely, Separate Selection Algorithm (SSA) and Combined Selection Algorithm (CSA) with low-time complexity.

The rest of this paper is organized as follows. In Section 2, we summarize the related works. Section 3 describes the system model and the problem formulation. Sections 4 and 5 have introduced the integer nonlinear programming formulation and two heuristic selection algorithms. Section 6

describes and analyzes the simulation results. We conclude this paper in Section 7.

2. Related Works

This paper focuses on the node selection problem in service-oriented wireless sensor networks. Several works have been done to develop service-oriented architecture specific to the sensor networks, and the architecture has shown many advantages in the heterogeneous applications scenarios. Gračanin et al. [2] proposed a service-centric model that focuses on services provided by a wireless sensor networks and views a wireless sensor networks as a service provider. This model consists of mission, network, region, sensor, and capability layers. Within each layer, there are four planes or functionality sets: communication, management, application, and generational learning. Rezgui and Eltoweissy [4] introduced the service-oriented architecture as an approach for building a new generation of open, efficient, interoperable, scalable, application-aware sensor-actuator networks. In this vision, sensor-actuator networks would not be deployed to provide sensing and actuation capabilities to a specific application but, rather, to provide sensing and actuation services to any application. King et al. [5] developed a service-oriented sensor and actuator platform called Atlas, which enables self-integrative, programmable pervasive spaces. This platform has shown the advantage of improving communication and interoperability between heterogeneous devices in pervasive computing environments. Authors of [11] proposed TinySOA, a service-oriented architecture that allows programmers to access wireless sensor networks from their applications by using a simple service-oriented API via the language of their choice. The main advantage of TinySOA is relieving application developers from dealing with the low-level technical details of the wireless sensor networks to get sensors data. Corchado et al. [12] proposed a service-oriented telemonitoring system for healthcare using heterogeneous wireless sensor networks, which aimed at improving healthcare and assistance for dependent people.

It is an important issue to provide services efficiently with resource-constrained sensor nodes. Node scheduling is considered as an efficient technique to implement the service-supporting schemes, in which sensor nodes should be selected to provide requested services. Recently, Wang et al. [13] investigated the service-availability-aware sleep scheduling design in service-oriented wireless sensor networks. The purpose of this study is to minimize the energy consumption and guarantee that enough sensors are active to ensure service availability at all times. The authors had proven this problem to be NP-hard and presented heuristic linear-programming based-solutions. However, they assumed that each service has a known requirement on the number of active sensors based on the historical service composition requests in the system, which may not be the case in practice. Furthermore, they only consider the sleep scheduling design for the sensors in the service provider overlay network and neglect the routing cost of service data. Authors of [14] try to identify the service composition that is less likely to be invalid in the near future due to nodes going to sleep mode. The goal is to minimize

the recomposition cost. They make use of the dynamic programming to reduce total service composition cost when the minimum number of required service composition solutions is derived. However, the dynamic programming is unsuitable for large-scale problems.

The distributed nature of wireless sensor network results in spatial correlation among the sensed data. And data accuracy is accordingly influenced by the spatial correlation. Under different assumptions, researchers have proposed several mathematical models for spatial correlation in wireless sensor networks. Some [18] assume that the sensed data follow diffusion property, and some [17] use an empirically obtained approximation function for the joint entropy of sensed data. The most commonly used model is the jointly Gaussian [9, 19, 20], which assumes the data to be jointly Gaussian with the correlation being a function of the distance. The jointly Gaussian model is easy to use and analyze. However, the chief limitation is that it forces the joint probability density function of the data values to be jointly Gaussian. Some researchers [21] use variograms to analyze spatial correlation in wireless sensor networks. The proposed model is Markovian in nature and can capture correlation in data irrespective of the node density, the number of source nodes, or the topology. Furthermore, this model derives the data value at a node from other correlated nodes whose data values have already been derived. However, it is not always the case that a given spatial process will be Markovian. Some others proposed correlation model for specific applications, such as soil moisture measurement in wireless underground sensor networks [10]. The presence of spatial correlation among sensor network data has been exploited for solving different problems. The authors in [15] proposed a traffic model for wireless sensor networks, which takes into account the statistical patterns of node mobility and spatial correlation. In [16, 17], spatial correlation was used to design energy-efficient data aggregation algorithms. Ma et al. [16] proposed a distributed clustering algorithm based on the dominating set theory to choose the cluster heads nodes and construct clusters by measuring the spatial correlation between sensors. Pattem et al. [17] studied the correlated data gathering problem and followed the idea of using an empirically obtained approximation function for the joint entropy of sources.

It is important and challenging to provide different services with data accuracy guaranteed through unreliable sensors nodes. Fault tolerance is one of the most important techniques, which has been taken into consideration in many works [22–27]. Han et al. [22] addressed the problem of deploying minimum number of relay nodes to achieve diverse levels of fault tolerance with higher network connectivity in the context of heterogeneous wireless sensor networks. However, they adopted the network model that in which nodes possess different transmission radius, while all of the relay nodes use an identical transmission radius. Banerjee et al. [23] investigated the event detection scheme with fault tolerance for multiple events occurring simultaneously. They proposed the use of polynomial-based scheme that addresses the problems of event region detection by having an aggregation tree of sensor nodes. However, their

work is limited to static sensor and the network topology cannot adapt to the dynamic nature of simultaneous events with varied priorities.

Our work in this paper is concerned with the node selection algorithms, which is similar to the works that aim at dealing with node selection and assignment problems. Cai et al. [28] addressed the multiple directional cover sets problem of organizing the directions of sensors into a group of nondisjoint cover sets to extend the network lifetime. The directional sensors are different from common sensors that have a limited angle of sensing range. The authors proved this problem is NP-complete and presented three heuristic approaches. Lin et al. [29] proposed an adaptive energy-efficient multisensor scheduling scheme for collaborative target tracking in wireless sensor networks. The challenging issue of this problem is how to achieve energy efficiency and track reliability while satisfying the tracking accuracy requirement. In their algorithm, a number of sensors are selected to form a temporary tasking cluster, and the optimal sampling interval is determined to satisfy the given tracking accuracy. Johnson et al. [30] considered sensor-mission assignment problem in wireless sensor networks. In this problem, multiple missions compete for sensor resources. They showed that this problem is NP-hard even to approximate, and presented several heuristic algorithms. Liu et al. [31] studied the topology control problem using a probabilistic network model. They attempted to find a minimal transmission range for each node while the global network reachability satisfies certain threshold. Different from these previous works, we aim at providing efficient node selection algorithms with data accuracy guaranteed for the service-oriented wireless sensor networks by exploring the spatial correlation among the sensed data and the advantages of diverse services provided by different sensor nodes.

3. System Model and Problem Formulation

In this section, we have firstly introduced the network model for the service-oriented wireless sensor networks. Secondly, we have described the spatial correlation model for single as well as multiple services in the network. Finally, we have formulated a definition for the node selection problem with data accuracy guaranteed in the service-oriented wireless sensor networks.

3.1. Network Model. We consider a wireless sensor network in the plane with stationary nodes $P = \{p_1, p_2, \dots, p_n\}$, which are built to provide a series of services $S = \{s_1, s_2, \dots, s_m\}$. Each node p_i can provide one or more service S_i , which is a subset of S ; that is, $S_i \subseteq S$. One service, for example, s_j , can be provided by a group of nodes P_j , and $P_j = \{p_i \mid s_j \in S_i\}$. It is obvious that the set size $|P_j|$ demonstrates the number of nodes in the network which can provide service s_j . The relationship between nodes and services can be further described as a bipartite graph $G = (P, S, E)$, where P denotes the set of nodes, S denotes the set of services, and E denotes the set of edges. There is an edge (p_i, s_j) between p_i and s_j in case that p_i can provide service s_j ; that is, $(p_i, s_j) \in E$. Figure 2

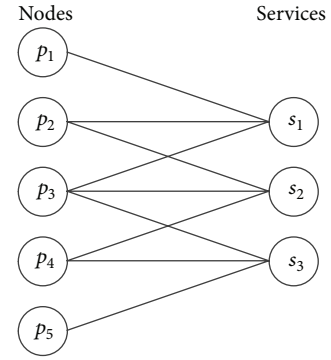


FIGURE 2: An example of the bipartite graph in which each service is supported by three distinct nodes.

has shown an example of the proposed model for service-oriented sensor network with five nodes and each service is supported by three distinct nodes.

To be convenient, the symbols used in this work are summarized in Table 1.

3.2. Spatial Correlation Model. Researchers have proposed several mathematical models for spatial correlation in wireless sensor networks under different assumptions. Pattem et al. [17] proposed to use an empirically obtained approximation function for the joint entropy of sensed data. In [18], the sensed data is assumed to follow the diffusion property and the diffusion is formulated as a function of the distance. The jointly Gaussian is adopted in many related works [9, 19, 20], which assumes the data to be jointly Gaussian with the correlation as a function of the distance. The jointly Gaussian model is easy to use and analyze by forcing the joint probability density function of the data values to be jointly Gaussian. Jindal and Psounis [21] analyzed the spatial correlation among sensed data by using variograms in wireless sensor networks. The proposed model is a special case of Markov random field. In this model, the data value at a node is derived from other correlated nodes whose data values have already been derived. However, it is not always the fact that a given spatial process will be Markovian. In this paper, we are concerned with the data accuracy with the spatial correlation model. The jointly Gaussian model proposed in [9] has considered the measurement noise of nodes and given the distortion function, which is suitable for our problem.

In the sensor networks, the observation result of each node is in fact a noisy version of the physical phenomenon located at the sensing source, and it can be modeled as Gaussian random variable V of zero mean and variable σ_V^2 ; that is, $V \sim \mathcal{N}(0, \sigma_V^2)$. Similarly, the sensed data for the physical phenomenon at node p_i can also be modeled as Gaussian variable V_i , $V_i \sim \mathcal{N}(0, \sigma_{V_i}^2)$. Assume that the sensed data for node p_i/p_j is denoted as V_i/V_j accordingly. The correlations between V and V_i , V_i and V_j are described as

$$\begin{aligned} \text{corr}\{V, V_i\} &= \rho_{v,i} = K_9(d_{v,i}), \\ \text{corr}\{V_i, V_j\} &= \rho_{i,j} = K_9(d_{i,j}), \end{aligned} \quad (1)$$

where $d_{v,i}$ denotes the Euclidean distance between p_i and the sensing source V , $d_{i,j}$ denotes the Euclidean distance between p_i and p_j , $K_\theta(\cdot)$ is the covariance function concerned with the Euclidean distance d and it is formulated as

$$K_\theta(d) = e^{-(d/\theta)}, \quad \theta > 0, \quad (2)$$

where θ controls the correlation between the distances of sensor nodes. In addition, we can see that $K_\theta(\cdot) = 1$ in case that $d = 0$, and $K_\theta(\cdot) = 0$ in case that $d = +\infty$.

The collected data by the sensor node p_i is often subject to noise interference originated from the environment, and it can be represented as

$$X_i = V_i + N_i, \quad (3)$$

where N_i is the additive white Gaussian noise, $N_i \sim \mathcal{N}(0, \sigma_N^2)$. We assume that the noise that each sensor node encounters is independent of each other.

According to [9], the distortion of the estimation for V is formulated as

$$\begin{aligned} \hat{\delta} = \sigma_V^2 - \frac{\sigma_V^4}{L(\sigma_V^2 + \sigma_N^2)} \left(2 \sum_{i=1}^L \rho_{v,i} - 1 \right) \\ + \frac{\sigma_V^6}{L^2(\sigma_V^2 + \sigma_N^2)^2} \sum_{i=1}^L \sum_{j \neq i}^L \rho_{i,j}, \end{aligned} \quad (4)$$

where L ($L > 0$) is the number of sensor nodes.

We use σ_V^2 to normalize $\hat{\delta}$, and the estimated data accuracy is calculated as

$$\begin{aligned} \hat{a} = 1 - \frac{\hat{\delta}}{\sigma_V^2} \\ = \frac{\beta}{L(\beta + 1)} \left(2 \sum_{i=1}^L \rho_{v,i} - 1 \right) \\ - \frac{\beta^2}{L^2(\beta + 1)^2} \sum_{i=1}^L \sum_{j \neq i}^L \rho_{i,j}, \end{aligned} \quad (5)$$

where $\beta = \sigma_V^2 / \sigma_N^2$ denotes the Signal-to-Noise Ratio (SNR).

3.3. Problem Definition. In this paper, we study the problem of node selection in the service-oriented wireless sensor networks with the data accuracy requirement guaranteed for the services. The number of nodes is considered as the optimizing object due to the following considerations. Firstly, there are fewer packets to be transmitted in the network if we select less number of nodes to provide services, which is also helpful to reduce the energy consumption. Secondly, it will increase the collision probability in the contention-based wireless network if too many nodes are kept awake, and significant retransmission cost and additional delay occur accordingly. Finally, it helps to reduce the overhead of data transmission to allow one node to provide multiple services simultaneously. In case that the data of different services is correlated, it can be compressed into a smaller packet; even

in the uncorrelated case, it can still be transmitted in a single packet, and thereby it is helpful to reduce overhead in the network [32].

In this paper, we aim at providing node strategies with the number of selected nodes minimized. Let a_j be the data accuracy requirement of each service s_j , P'_j one subset of nodes selected from P_j to provide service s_j , $P'_j \subseteq P_j$, and $\hat{a}_j(P'_j)$ the estimated data accuracy of service s_j when service s_j is provided by nodes in set P'_j . The node selection problem for the service-oriented wireless sensor networks can be defined as follows: given a bipartite graph $G = (P, S, E)$, in which P denotes the set of nodes, S denotes the set of services, and E denotes the set of edges between P and S , and given the required data accuracy requirement a_j of each service $s_j \in S$ and spatial correlation among these nodes and corresponding sensing sources, the problem is to find a subgraph $G' = (P', S, E')$, $P' \subseteq P$, $E' \subseteq E$, and the objective is to minimize the number of selected nodes in P' under the constraint that $\hat{a}_j(P'_j) \geq a_j$ is satisfied for each service $s_j \in S$.

4. Integer Nonlinear Programming Formulation

In this section, we present an Integer Nonlinear Programming (INLP) formulation for the node selection problem. Integer programming is a mathematical optimization or a feasibility program in which some or all of the variables are restricted to be integers. INLP is a special case of integer programming, where some of the constraints or the objective functions are nonlinear. INLP is considered as an efficient technique to solve the optimization problem with nonlinear constraint, so that it is feasible to express the node selection problem as INLP. This paper is concerned with the node selection problem, and the objective is to minimize the total number of selected nodes with nonlinear data accuracy constraint. We use the following set of binary integer (0 or 1) variables and constraints in the INLP formulation.

(1) Variables $e_{i,j}$ for each node $p_i \in P$ and service $s_j \in S$. The variable $e_{i,j}$ is 1 if and only if $(p_i, s_j) \in E$; that is, node p_i can provide service s_j :

$$e_{i,j} = \begin{cases} 1, & \text{if node } p_i \text{ can provide service } s_j, \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

The variables $e_{i,j}$ can be obtained when the topology graph and the set of services provided by nodes are given. Obviously, the selected nodes for services s_j shall be selected from these nodes with $e_{i,j} = 1$.

(2) Variables $x_{i,j}$ for each node $p_i \in P$ and service $s_j \in S$. The variable $x_{i,j}$ is 1 if and only if node p_i is assigned to provide service s_j :

$$x_{i,j} = \begin{cases} 1, & \text{if node } p_i \text{ is assigned to provide service } s_j, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

As we can see, the set $\{x_{i,j} \mid i = 1, 2, \dots, n, j = 1, 2, \dots, m\}$ had indicated one scheduling scheme for the given wireless

sensor networks, in which each node p_i is assigned to provide service s_j if $x_{i,j} = 1$. Note that $x_{i,j}$ equals 0 in case that $e_{i,j} = 0$, which means that node p_i cannot be assigned to provide service s_j since it is not supported. In this way, we have the following constraint:

$$e_{i,j} - x_{i,j} \geq 0, \quad \forall i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m. \quad (8)$$

Let \hat{a}_j be the estimated data accuracy of service s_j , and \hat{a}_j can be obtained via formula (5), which can be further described as follows:

$$\hat{a}_j = \begin{cases} \frac{\beta}{L_j(\beta+1)} \left(2 \sum_{i=1}^n x_{i,j} \rho_{v,i} - 1 \right) \\ \quad - \frac{\beta^2}{L_j^2(\beta+1)^2} \sum_{i=1}^n \sum_{k \neq i}^n x_{i,j} x_{k,j} \rho_{i,k}, & \text{if } L_j \neq 0 \\ 0, & \text{if } L_j = 0, \end{cases} \quad (9)$$

where $L_j = \sum_{i=1}^n x_{i,j}$ and L_j denotes the number of selected nodes which are assigned to provide service s_j .

In order to satisfy the required data accuracy for all services in the network, we have the following constraint:

$$\hat{a}_j \geq a_j, \quad \forall j = 1, 2, \dots, m. \quad (10)$$

(3) Variables y_i . The variable y_i is 1 if and only if node p_i is selected to provide services required in the network:

$$y_i = \begin{cases} 0, & \text{if node } p_i \text{ is not selected to provide any services,} \\ 1, & \text{otherwise.} \end{cases} \quad (11)$$

By following the above definition, y_i equals 0 if node p_i is not selected to provide any service, and otherwise it equals 1. As mentioned above, the variable $x_{i,j}$ denotes the case that node $p_i \in P$ can provide service $s_j \in S$ or not, and we have the following constraint:

$$m y_i \geq \sum_{j=1}^m x_{i,j} \geq y_i, \quad \forall i = 1, 2, \dots, n. \quad (12)$$

The objective of node selection problem is to minimize the total number of nodes that are selected to provide the required services, and the number of selected nodes can be calculated as $\sum_{i=1}^n y_i$.

Then the node selection problem discussed in this paper can be formulated as

$$\begin{aligned} & \text{Minimized} \quad \sum_{i=1}^n y_i \\ & \text{Subject to} \\ & \quad x_{i,j} \in \{0, 1\}, \quad \forall i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m, \\ & \quad y_i \in \{0, 1\}, \quad \forall i = 1, 2, \dots, n, \\ & \quad \hat{a}_j \geq a_j, \quad \forall j = 1, 2, \dots, m, \\ & \quad e_{i,j} - x_{i,j} \geq 0, \quad \forall i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m, \\ & \quad m y_i \geq \sum_{j=1}^m x_{i,j} \geq y_i, \quad \forall i = 1, 2, \dots, n. \end{aligned} \quad (13)$$

In this section, we have introduced an Integer Nonlinear Programming (INLP) formulation for the node selection problem. The proposed INLP is generally considered as an efficient way to provide an accurate description on the problem formulation. This formulation is useful to find the optimal solution in case that the solution space is small enough with the help of some well-known tools, such as LINGO and MATLAB. However, INLP is a special case of integer programming which is proved to be NP-hard, and accordingly the INLP problem is NP-hard. Many related works have also been proposed to find the suboptimal solution for a given INLP problem [33–35]. Although INLP has shown its good performance in the practical applications, it results in some well-known defects such as computation as well as space complexity, especially in case that the solution space is very large. Unfortunately, the wireless sensor network generally includes hundreds to thousands of nodes; the variables required by the INLP mentioned above might increase exponentially with the node number. Another problem is that for a random network, it is hard to gather all the constraints mentioned above since these nodes are heterogonous and constraints for each node are fully different from each other. Moreover, in the practical applications, the sink is almost impossible to get the *accurate* information in the harsh environment by following the observation that the sensed data is generally a noisy version of the physical phenomenon. It is more practical to adopt a suboptimal solution with the data accuracy guaranteed instead of optimal one that is hard to find for an NP-hard problem. In this way, it is reasonable and necessary to develop heuristic algorithms for the node selection problem in the service-oriented wireless sensor networks.

5. Heuristic Algorithms

Heuristic algorithms are generally considered as an important way to solve the NP-hard problem. In this section, we propose two heuristic algorithms for the node selection problem in the service-oriented wireless sensor networks,

namely, the Separate Selection Algorithm (SSA) and the Combined Selection Algorithm (CSA).

5.1. Separate Selection Algorithm (SSA). The basic idea of the Separate Selection Algorithm (SSA) is that we select a minimum number of nodes for each required service with the data accuracy guaranteed in a separate way, and the union of selected nodes for all services is considered as the problem solution. The key process for the SSA is how to select nodes for each service. Here we follow the idea with which nodes are selected in a sequence way, and in each step, we will choose the node that is potential to improve the data accuracy.

Assume that the current node selection solution is $G' = (P', S, E')$, in which $P' = \{P'_j \mid j = 1, 2, \dots, m\}$, $S = \{s_j \mid j = 1, 2, \dots, m\}$ is the set of services, and $E' = \{(p_i, s_j) \mid p_i \in P', s_j \in S\}$. Let us consider the general case that one node, that is, p_i is considered to provide one special service, that is, s_j . Here we use $I_{i,j}$ to denote the data accuracy increment for service s_j in case that node p_i is selected to provide service s_j , where $(p_i, s_j) \in E$. $I_{i,j}$ can be calculated as follows.

- (1) In case that $\hat{a}_j(P'_j) \geq a_j$, we have $I_{i,j} = 0$, which means that the data accuracy requirement for service s_j has already been satisfied that there is no more improvement once nodes p_i and (p_i, s_j) are added into the final solution.
- (2) In case that $\hat{a}_j(P'_j + \{p_i\}) \leq \hat{a}_j(P'_j)$, we have $I_{i,j} = 0$, which means that the data accuracy of service s_j cannot be increased once nodes p_i and (p_i, s_j) are added into the final solution.
- (3) In case that $\hat{a}_j(P'_j) < \hat{a}_j(P'_j + \{p_i\}) \leq a_j$, we have $I_{i,j} = \hat{a}_j(P'_j + \{p_i\}) - \hat{a}_j(P'_j)$, which denotes the increment of the data accuracy for service s_j in case that nodes p_i and (p_i, s_j) are added into the final solution.
- (4) In case that $\hat{a}_j(P'_j) < a_j < \hat{a}_j(P'_j + \{p_i\})$, we have $I_{i,j} = a_j - \hat{a}_j(P'_j)$, which means that we generally neglect the part of increment that exceeds the requirement since the solution is required only to provide the asked data accuracy.

The pseudocode for SSA is listed in Table 1. In Line 1, the set of selected nodes P'_j for each service s_j is initially set as \emptyset . In Lines 2–9, the algorithm tries to select nodes for each service $s_j \in S$. In case that the current data accuracy cannot satisfy the data accuracy requirement, that is, $\hat{a}_j(P'_j) < a_j$, we firstly check all the candidate nodes that are useful to improve data accuracy. Secondly, we select one node with maximum data accuracy increment as the candidate (in Line 4). Finally, the selected node is added into P'_j to provide service s_j (in Line 6). This process continues until enough nodes are selected for all services.

We illustrate the SSA algorithm by an example given in Figure 3. As we can see from Figure 3(a), the network has four nodes and each service is supported by three distinct nodes. The required data accuracy for each service is listed on the right side; for example, the required data accuracy of s_1 is 0.8,

TABLE 1: Description of the symbols.

Symbol	Description
n	The number of nodes
m	The number of services
G	The node-service bipartite graph
P	The set of nodes
S	The set of services
E	The set of edge between P and S
G'	The node-service assignment bipartite graph
P'	The set of nodes that selected to provide services
E'	The set of edge between P' and S
P_j	The set of nodes that can provide service s_j
P'_j	The subset of nodes that selected from P_j to provide service s_j
S_i	The set of services that node p_i can provide
a_j	The data accuracy requirement of service s_j
$\hat{a}_j(A)$	The estimated data accuracy of service s_j when service s_j is provide by nodes in set A

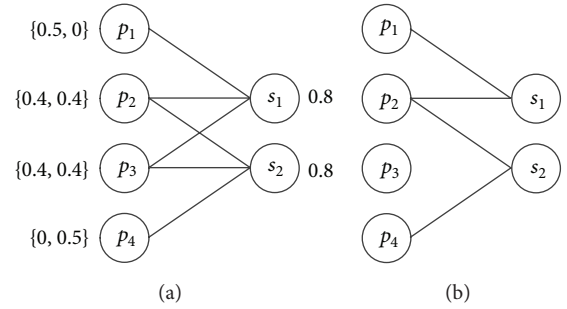


FIGURE 3: An example of the execution of SSA algorithm. (a) The node-service bipartite graph $G = (P, S, E)$. (b) The final solution of node-service selection bipartite graph $G' = (P', S, E')$.

and the data accuracy increment of each node when services is provided only by this node is also listed on the left side; for example, the data accuracy increment of p_1 is 0.5 for s_1 and 0.0 for s_2 . Without loss of generality, we assume that the data accuracy increment of each node can be added directly to simplify description; that is, the data accuracy of s_1 provided by p_1 and p_2 is $0.5 + 0.4$ that equals 0.9. The algorithm will select nodes for s_1 and then s_2 . Node p_1 with the maximum data accuracy increment for s_1 will be selected firstly. In case that the algorithm has selected p_1 , both node p_2 and p_3 can guarantee the required data accuracy of s_1 , and we assume that the randomly selected one is p_2 . Similarly, service s_2 will firstly select p_4 . In case that the algorithm has selected p_4 , both p_2 and p_3 can guarantee the required data accuracy of s_2 . However, p_2 is intended to be selected by s_2 in case that it is already selected by s_1 . After that, the selected nodes have guaranteed the required data accuracy of s_1 and s_2 , and the algorithm will stop. As we can see from the final solution shown in Figure 3(b), the SSA algorithm selects three nodes to provide s_1 and s_2 with one node reduced.

Input: Node-service bipartite graph $G = (P, S, E)$ and requirement for service data accuracy a_1, a_2, \dots, a_m .
Output: Node-service selection bipartite graph $G' = (P', S, E')$.

```

1:  $P' \leftarrow \emptyset, E' \leftarrow \emptyset, P'_j \leftarrow \emptyset, j = 1, \dots, m$ ;
2: for each service  $s_j \in S$ 
3:   while  $\hat{a}_j(P'_j) < a_j$ 
4:     Calculate data accuracy increment  $I_{i,j}$  for each  $p_i, (p_i, s_j) \in E$  and  $(p_i, s_j) \notin E'$ , and
       select the one  $p_i$  with maximum  $I_{i,j}$  and  $I_{i,j} > 0$  as the candidate node; if two nodes have
       identical increment, the node included in  $P'$  is prior to be selected;
5:     If no such a candidate is found, the algorithm stops with no solution found;
6:      $P'_j \leftarrow P'_j + \{p_i\}, E' \leftarrow E' + (p_i, s_j)$ ;
7:   end while
8:    $P' \leftarrow P' \cup P'_j$ ;
9: end for

```

ALGORITHM 1: Pseudocode for Separate Selection Algorithm (SSA).

5.2. Combined Selection Algorithm (CSA). The previous proposed SSA tries to select nodes for each separate service based on the criterion of data accuracy increment. However, some nodes will provide several services simultaneously in the wireless sensor networks, and this multiservice property can help to improve the performance of node selection strategies if we simply select one multiservice node to improve the data accuracy required by different services. As we can see from Line 4 in Algorithm 1, we intend to select the candidate node that is already chosen to provide some other service in SSA algorithm, which means that nodes with multiservice property are more preferred in SSA algorithm during node selection process. However, this separate selection process is not always efficient especially in some cases. For example, the sample network given in Figure 3 can obtain a better solution than the solution found by SSA. If we do not select p_1 and p_4 which have maximum data accuracy increment for one special service, but select p_2 and p_3 which can provide service s_1 and s_2 simultaneously, it is obvious that this solution can guarantee the required data accuracy of s_1 and s_2 . However, this solution has fewer nodes than SSA because it only selects two nodes with two nodes reduced. In this section, we will introduce a new Combined Selection Algorithm (CSA) that has utilized multiservice property for node selection problem in service-oriented wireless sensor networks.

In this paper, we aim at minimizing the number of selected nodes with the data accuracy guaranteed for all services in the network. There are two important factors that will influence the number of selected nodes; that is, the number of services and the service quality of each node. Intuitively, it helps to reduce the number of selected nodes in case that they can provide more kinds of services since more nodes are potential candidates during the selection process. However, nodes might have poor data accuracy when they are far away from the sensing source although they can provide the required services. It means that we shall consider the data accuracy as well as the number of services simultaneously during the node selection process.

The basic idea of the CSA is described as follows. Initially no nodes are selected and the final solution is an empty set.

Then, nodes are chosen and added into the final solution in a sequence way. In each step, we intend to select the node with maximal *contribution increment* to all services in the network (which will be discussed in details below in this section). This process continues until the data accuracy for all services is satisfied, and finally we can obtain the selected nodes as well as the services provided by each node for the problem.

Assume that p_i is considered to provide services s_j in the current selection bipartite graph G' . In case that $I_{i,j} = 0$, it is obvious that there is no benefit for node p_i to provide s_j , and we have $(p_i, s_j) \notin E'$. In case that $I_{i,j} > 0$, we can see that node p_i helps to improve data accuracy of service s_j , and we have $(p_i, s_j) \in E'$. In this way, for a given p_i , we can calculate the $I_{i,j}$ for each s_j . Generally, we intend to choose the node that has more contribution increment to the data accuracy. Here we have introduced $w_{i,j}$ ($0 < w_{i,j} \leq 1$) as a coefficient to demonstrate the impact of current contribution increment on the final data accuracy, and it is formulated as

$$w_{i,j} = \begin{cases} 0, & I_{i,j} = 0, \\ \exp \left\{ - \left(\frac{(a_j - \hat{a}_j(P'_j) - I_{i,j})}{a_j} \right) \right\}, & I_{i,j} > 0. \end{cases} \quad (14)$$

As we can see from the above formulation, $w_{i,j} = 0$ in case that $I_{i,j} = 0$, which shows that node p_i has no contribution to the data accuracy of service s_j . Here we adopt the power exponential function to indicate how much the data accuracy is close to the requirement value.

Let C_i be the contribution increment of node p_i with the current selection bipartite graph G' in case that p_i is selected to provide services, and C_i is formulated as

$$C_i = \sum_{(p_i, s_j) \in E} w_{i,j} I_{i,j}, \quad \forall p_i \in P, \quad (15)$$

where $w_{i,j}$ ($0 < w_{i,j} \leq 1$) is a coefficient and $I_{i,j}$ denotes the data accuracy increment for service s_j in case that node p_i

Input: Node-service bipartite graph $G = (P, S, E)$ and requirement for service data accuracy a_1, a_2, \dots, a_m .
Output: Node-service selection bipartite graph $G' = (P', S, E')$.

```

1:  $P' \leftarrow \emptyset, E' \leftarrow \emptyset, P'_j \leftarrow \emptyset, j = 1, \dots, m;$ 
2: while true
3:   If for each service  $s_j \in S$ , has satisfied  $\hat{a}_j(P'_j) \geq a_j$ , the algorithm stops with solution found;
4:   Calculate data accuracy increment  $I_{i,j}$  and contribution increment  $C_i$  for each  $p_i, p_i \in P$ 
     and  $p_i \notin P'$ , and select the one  $p_i$  with maximum  $C_i$  and  $C_i > 0$  as the candidate node;
5:   If no such a candidate is found, the algorithm stops with no solution found;
6:   for each service  $s_j, (p_i, s_j) \in E$  and  $I_{i,j} > 0$ 
7:      $P'_j \leftarrow P'_j + \{p_i\}, E' \leftarrow E' + (p_i, s_j);$ 
8:     while  $P'_j \neq \emptyset$ 
9:       Calculate  $\hat{a}_j(P'_j - \{p_k\})$  for each  $p_k, p_k \in P'_j$ , and select the one  $p_k$  with maximum
          $\hat{a}_j(P'_j - \{p_k\})$  and  $\hat{a}_j(P'_j - \{p_k\}) \geq \hat{a}_j(P'_j)$ ;
10:      If no such a candidate is found, break;
11:       $P'_j \leftarrow P'_j - \{p_k\}, E' \leftarrow E' - (p_k, s_j);$ 
12:    end while
13:  end for
14:   $P' \leftarrow \bigcup_{j=1}^m P'_j;$ 
15: end while

```

ALGORITHM 2: Pseudocode for Combined Selection Algorithm (CSA).

is selected to provide service s_j , which is as same as that in Section 5.1.

So far we have introduced the basic node selection process for the CSA. However, the algorithm can be further optimized. In case that the algorithm selects one node with maximum contribution increment, this node will provide each service that helps to improve the data accuracy. However, the node with multiservice and maximum contribution increment might have poor data accuracy increment for some services. Although the node that provides poor data accuracy increment can still improve the data accuracy, the data accuracy increment is so small that it needs to select more nodes to guarantee the required data accuracy. Therefore, we can further reduce the number of nodes by removing some already selected nodes that are with poor data accuracy increment for some services. Let us consider an example that two services are supported by two nodes, in which p_1 can support s_1 and s_2 , p_2 can support s_2 , and the required data accuracy for s_1 and s_2 is both 0.8. Suppose that the data accuracy of s_1 provided by p_1 is 0.8, the data accuracy of s_2 provided by p_1 is 0.3, provided by p_2 is 0.8, and provided by p_1 and p_2 is 0.7. In the first selection, the value of C_1 is larger than C_2 according to formula (15), then p_1 will be selected and provide service s_1 and s_2 . In the next selection, p_2 will be selected and provide service s_2 . However, the data accuracy of s_2 provided by p_1 and p_2 is less than that provided by p_2 . It is clear that we can improve the data accuracy of service s_2 if we let p_1 do not provide s_2 . Note that the “bad” assignments (i.e., assigning nodes to provide services that are with poor data accuracy increment) cannot be eliminated during the selection process, due to the fact that they still help to improve the data accuracy. After selecting a new node, we can check all the selected nodes to find the “bad” assignments that were included in the previous selections. The basic idea

of the optimization process is described as follows. In case that there is a special service for example, s_j , has selected a new node, we will firstly calculate $\hat{a}_j(P'_j - \{p_i\})$ for each node $p_i \in P'_j$, and select the one p_i with maximum $\hat{a}_j(P'_j - \{p_i\})$ and $\hat{a}_j(P'_j - \{p_i\}) \geq \hat{a}_j(P'_j)$; after that, we let p_i do not provide s_j . This process continues until no more of this kind of nodes can be found from P'_j .

The pseudocode for CSA is listed in Algorithm 2. In Line 1, the set of selected nodes P'_j for each service s_j is initialized as \emptyset . In Lines 2–15, nodes are selected in a sequence way until data accuracy is guaranteed for all services. In case that there is some service that current data accuracy cannot satisfy its requirement, we will firstly calculate all the candidate nodes' contribution increment, and select one node with maximum contribution increment as the candidate (in Line 4). Secondly, the candidate node is assigned to provide each service s_j that helps to improve the data accuracy (in Line 7). Finally, we check all the nodes in P'_j and remove nodes from P'_j without declining the data accuracy of service s_j , and this subprocess continues until no more of this kind of nodes can be found from P'_j (in Line 8–12). The node selection process continues until enough nodes are selected for all services.

We illustrate the execution of CSA algorithm during one round of the iteration process by an example given in Figure 4. In this example two services are supported by four nodes. The node-service bipartite graph $G = (P, S, E)$ is given in Figure 4(a), and the current node-service selection bipartite graph $G' = (P', S, E')$ is given in Figure 4(b). As we can see from Figure 4(b), the algorithm has selected p_2 in G' , then the available nodes are p_1, p_3 , and p_4 . Suppose that the contribution increment of each available node has been

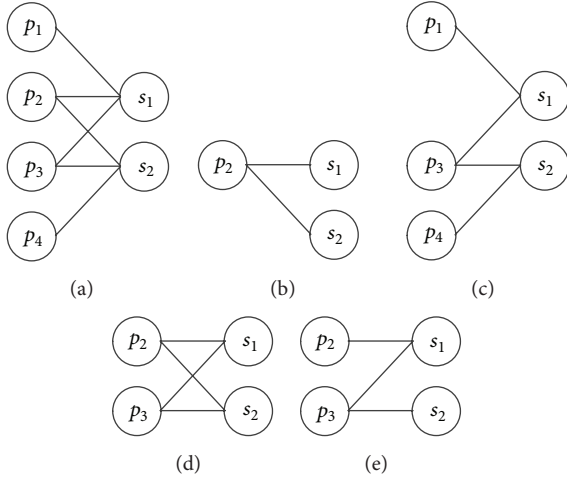


FIGURE 4: An example of the execution of CSA algorithm during one round of the iteration process. (a) The node-service bipartite graph $G = (P, S, E)$. (b) The current node-service selection bipartite graph $G' = (P', S, E')$. (c) The relationship between available nodes and services, and there is an edge between p_i and s_j indicating that p_i helps to improve the data accuracy of s_j . (d) Select p_3 to provide s_1 and s_2 . (e) The final node-service selection bipartite graph G' of this round of the iteration process.

calculated and the value of C_1 , C_3 , and C_4 is 0.1, 0.3, and 0.2, respectively. The relationship between available nodes and services is given in Figure 4(c), and there is an edge between p_i and s_j indicating that the data accuracy increment of p_i is larger than 0, that is, p_i helps to improve the data accuracy of s_j . In the next step, the algorithm will select one node with maximum contribution increment, that is, p_3 in this example. According to Figure 4(c), p_3 helps to improve the data accuracy of s_1 and s_2 . Then p_3 will provide s_1 and s_2 , and the new solution is given in Figure 4(d). After a new node is added into the solution, the algorithm will execute an optimization process. We assume that there is one “bad” assignment (p_2, s_2) ; that is, the data accuracy of s_2 provided by p_3 is not less than that provided by p_2 and p_3 . Then the algorithm will remove the assignment (p_2, s_2) from the solution. The final solution of this round of the iteration process can be observed from Figure 4(e).

5.3. Complexity Analysis

Lemma 1. The time complexity of SSA is $O(mn^2)$.

Proof. During the outside for loop, the algorithm selects nodes for each service $s_j \in S$, and each execution of the outside for loop contains a while loop. In the while loop, the algorithm checks each node p_i , $(p_i, s_j) \in E$ and $(p_i, s_j) \notin E'$ and selects one node with maximum data accuracy increment. The while loop will continue until the data accuracy of s_j is satisfied. Because there are at most n nodes that can provide s_j and each execution selects one

node, the execution of the while loop takes $O(n^2)$ time. Hence, the time complexity of SSA is $O(mn^2)$. \square

Lemma 2. The time complexity of CSA is $O(mn^3)$.

Proof. During the outside while loop, the algorithm will firstly check each node p_i , $p_i \in P$, $p_i \notin P'$, and calculate the data accuracy increment for each services and node's contribution increment. Because each node can provide at most m services and each execution selects one node with maximum contribution increment, this process takes at most $O(mn)$ time. In the next step, the loop is executed to assign the selected node to provide each service that helps to improve the data accuracy, and there are at most m services. In each execution of the inside while loop, the algorithm checks each node in P'_j and tries to find one node that without declining the data accuracy of s_j when this node does not provide s_j . The inside while loop will continue until no more of this kind of nodes can be found. Because each P'_j contains at most n nodes and each execution of the inside while loop selects one node, the inside while loop takes at most $O(n^2)$ time and for loop is $O(mn^2)$. Therefore, each execution of the outside while loop takes at most $O(mn + mn^2) = O(mn^2)$ time. Because there are at most n nodes to be selected, the time complexity of CSA is $O(mn^3)$. \square

6. Simulation Results and Analysis

In order to evaluate the actual behavior of the above algorithms, we have relied on the experimental simulation to show its performance. In this section, we have firstly introduced the building process of our simulation and then analyze the impact of spatial correlation parameters θ and SNR β on the results. Finally, we compare the performance of SSA and CSA in different environments.

6.1. Simulation Setup. We use MATLAB as the platform tool that is used popularly in simulation of wireless networks. The scenarios are built in a square area $500 \text{ m} \times 500 \text{ m}$. The sensor nodes are random placed as well as the sensing sources. Here we assume that each sensing source is dedicated to one special service. Given the sensor nodes and the sensing sources, in the next step we need to decide the set of services provided by these nodes. Here we adopt the randomly model to determine whether node p_i can provide service s_j with a given probability ratio q ($0 < q < 1$); that is, p_i only provides s_j in case that the random value (between 0 and 1) is larger than q . Here we also assume that each service is provided by at least one node. Otherwise, the scenario is rebuilt until this constraint is satisfied. And the data accuracy \hat{a} for each service is assumed to be identical. In this work, we build 100 different scenarios and compare the average performance of the proposed algorithms.

6.2. Impact of Spatial Correlation Parameters θ and SNR β . In this part, we analyze the impact of spatial correlation parameters θ and SNR β on the performance of the SSA and CSA. The spatial correlation parameters θ and SNR β are two

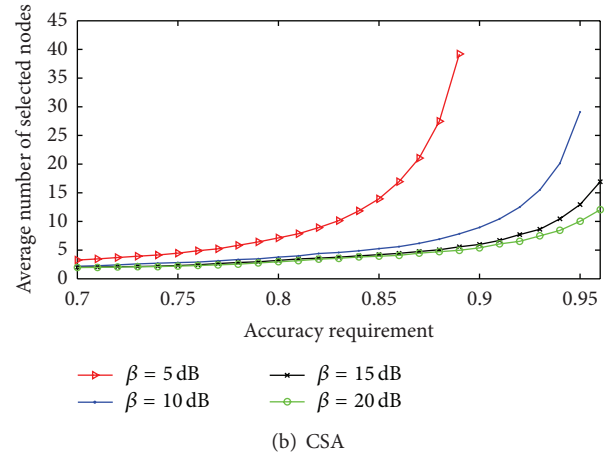
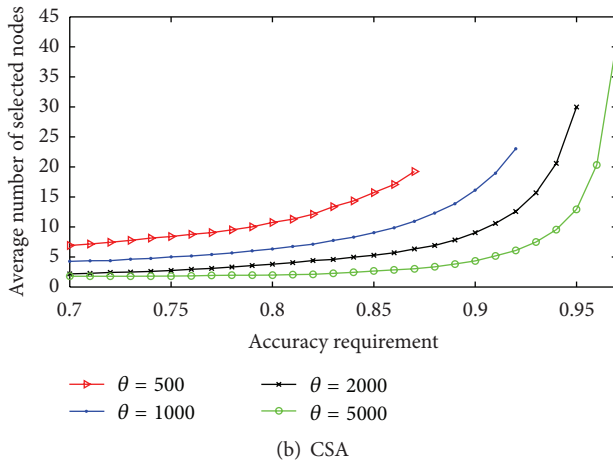
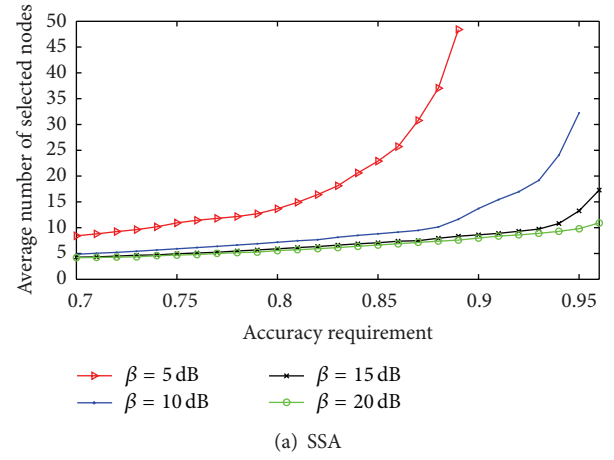
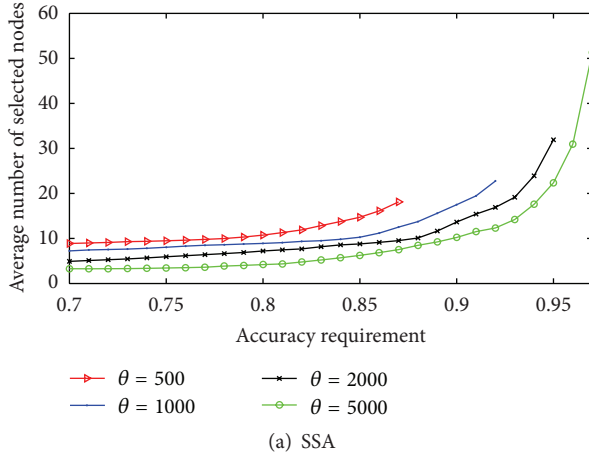


FIGURE 5: Impact of spatial correlation parameter θ , $\theta \in \{500, 1000, 2000, 5000\}$.

FIGURE 6: Impact of SNR β , $\beta \in \{5 \text{ dB}, 10 \text{ dB}, 15 \text{ dB}, 20 \text{ dB}\}$.

parameters in the spatial correlation model, which we have introduced in Section 3.2. The spatial correlation parameter θ denotes the correlation of sensed data between the distances among sensor nodes. As we can see from formula (2), the larger θ indicates a high degree of spatial correlation; that is, the nodes in a network provide strongly correlated service. The SNR β denotes the noise strength that will affect the distortion of service. It is obvious that the larger β will result in low distorted sensed data; that is, the services provided by nodes are more accurate. As we can see, the two parameters θ and β will affect the sensed data, which in turn influences the selection results.

The first set of experiments is concerned with the impact of spatial correlation parameter θ on the number of selected nodes. The simulation is done with 300 nodes and 10 services, and the SNR β is assumed to be 10 dB and q be 0.5. The spatial correlation parameter θ varies from 500, 1000, and 2000 to 5000, and we study the average number of the selected nodes compared with the change of services' accuracy requirement \hat{a} that starts from 0.7 to 0.97. As we can see from Figure 5, the number of selected nodes is minimized in case that $\hat{a} = 0.7$, and it increases together with the increasing of accuracy requirement \hat{a} . However, this process is not so significant

until \hat{a} reaches some special point. For example, the average number of selected nodes is among 1.79 to 4.34 when $0.7 \leq \hat{a} \leq 0.9$ in case that $\theta = 5000$ and using CSA algorithm to find solution; however, it increases rapidly when $\hat{a} > 0.9$. Moreover, there might have not been enough nodes to support the required data accuracy requirement; for example, the maximum data accuracy requirement is about 0.87 in case that $\theta = 500$.

The second set of experiments is concerned with the impact of signal-to-noise ratio β on the number of selected nodes, which is illustrated in Figure 6. The simulation is done with 300 nodes and 10 services, and spatial correlation parameter θ is assumed to be 2000 and q be 0.5. The SNR parameter β varies from 5 dB, 10 dB, and 15 dB to 20 dB, and we study the average number of selected nodes compared with the change of services' accuracy requirement \hat{a} that starts from 0.7 to 0.96. We also can see that the number of selected nodes remains stable or varies linearly when \hat{a} is smaller; however, it increases rapidly when \hat{a} is larger than some special point. This conclusion is similar to that of Figure 5. As we know, the energy budget is an important criterion for the wireless sensor networks, and it will worsen the network performance if too many nodes are involved in the data sensing process. The compromise from a given application

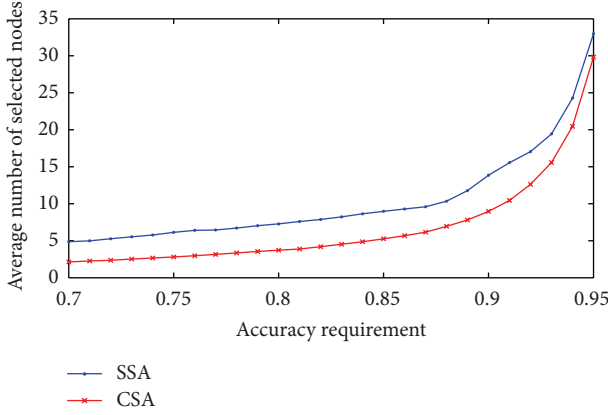


FIGURE 7: Comparison of SSA and CSA with different data accuracy requirements.

scenario will help to reduce the energy consumption by selecting a proper accuracy requirement.

6.3. Performance Comparison between SSA and CSA. So far as we know, this is the first works concerned with the node selection algorithms with data accuracy guaranteed for the service-oriented wireless sensor networks. Most of the related works [28–31] focused on different research issues, such as target tracking, and topology control. Wang et al. [13] had proposed a scheduling algorithm for the service-oriented wireless sensor network, but it did not consider the data accuracy. In this section, we compare the performance of SSA and CSA in different scenarios with varied accuracy requirement, number of nodes n , number of service m , and the value of q , respectively.

Figure 7 has shown the number of selected nodes with SSA and CSA when the accuracy requirement \hat{a} varies from 0.7 to 0.95. The simulation is done with 300 nodes and 10 services, and spatial correlation parameter θ is assumed to be 2000, SNR β to be 10 dB, and q to be 0.5. The experimental results show that CSA has better performance compared with SSA in all situations.

The second set of simulations is done to show the impact of network size on the number of selected nodes. The simulation is done with 300 nodes and 10 services, and spatial correlation parameter θ is assumed to be 2000, data accuracy requirement \hat{a} to be 0.92, SNR β to be 10 dB, and q be 0.5. And the network size varies from 100 to 500. As we can see from Figure 8, the CSA has better performance than SSA in all cases. Furthermore, we have two observations from Figure 8. (1) The average number of the selected nodes is relatively smaller in case that the network size is larger. This is due to the fact that there are more potential candidates for a given service with the network size increasing, and it helps to reduce the number of selected nodes. (2) The number of the selected nodes decreases slightly in case that the network size reaches some special point. It implies that it is helpless to reduce the number of selected nodes by adding more nodes into the network.

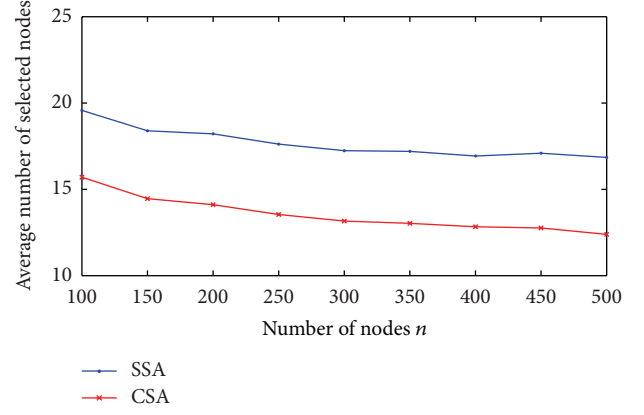


FIGURE 8: Comparison of SSA and CSA with different network size.

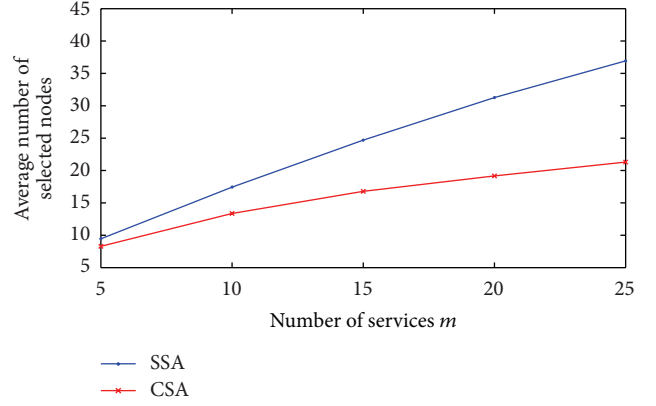


FIGURE 9: Comparison of SSA and CSA with different number of services.

The third set of simulations focuses on the impact of the number of services in the network on the number of selected nodes. We use the similar parameters in the second set of simulations. As we can see from Figure 9, CSA runs better than SSA with different value of m although it is not so significant when m is close to 5.

The fourth set of simulations focuses on the probability q on the number of selected nodes by varying q from 0.1 to 0.9. We also use the similar parameters in the second set of simulations. In fact, the parameter q indirectly represents the number of services provided by nodes in the network. As we can see from Figure 10, the number of the selected nodes is rather close with SSA and CSA when q is small enough. Particularly, the SSA is even slightly better than CSA when $q = 0.1$. However, the CSA shows better performance when the value of q increases. Meanwhile, we can also obtain two conclusions from this set of experiments. (1) The average number of the selected nodes decreases with q increasing. The larger q results in more services that can be provided by selected nodes. Thus, each node can make more contribution to the required services, which in turn reduces the total number of selected nodes. (2) In case that q is larger enough,

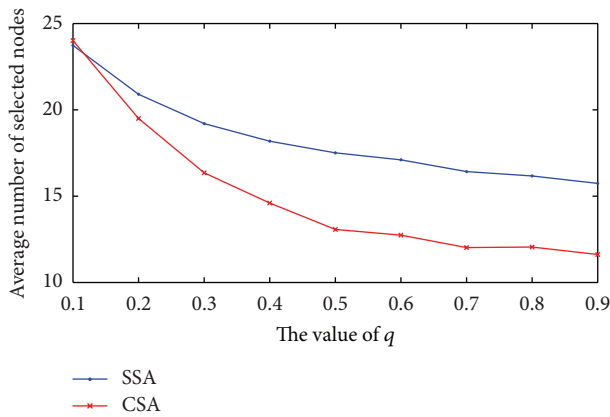


FIGURE 10: Comparison of SSA and CSA in different value of q .

for example, $q > 0.5$, the average number of selected nodes decreases slowly with q increasing.

7. Conclusion

To provide various services is one important trend for the future wireless sensor networks, and the service-oriented architecture allows different services supported simultaneously in the same physical area in which one sensor can provide different kinds of service. Quality of services, such as data accuracy, is one of the key criteria for applications because the sensed data is generally a noisy version of the physical phenomenon. The spatial correlation among the sensed data makes it possible to select a subset of nodes to provide the required services while the data accuracy is guaranteed, which is obviously helpful to improve the performance of the wireless sensor networks. We are concerned with this issue in this paper and have formulated the node selection problem into an Integer Nonlinear Programming (INLP) problem. We also have developed two heuristic algorithms, namely, Separate Selection Algorithm (SSA) and Combined Selection Algorithm (CSA) for the problem. In the future work we are to develop efficient scheduling schemes for the node selection process and aim at providing a solution for the service-oriented wireless sensor networks with the network lifetime maximized. The temporal correlation is also important to optimize the network performance. We also plan to explore energy-efficient scheduling schemes for service-oriented wireless sensor networks with both spatial and temporal correlation considered.

Acknowledgments

This work is supported by Fujian Provincial Natural Science Foundation of China under Grant no. 2011J01345, the Development Foundation of Educational Committee of Fujian Province under Grant no. 2012JA12027, the National Science Foundation of China under Grant no. 61103275, and the Technology Innovation Platform Project of Fujian Province under Grant no. 2009J1007.

References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.
- [2] D. Gračanin, M. Eltoweissy, A. Wadaa, and L. A. DaSilva, "A service-centric model for wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 6, pp. 1159–1166, 2005.
- [3] X. Chu and R. Buyya, "Service oriented sensor web," in *Sensor Network and Configuration: Fundamentals, Techniques, Platforms, and Experiments*, N. P. Mahalik, Ed., pp. 51–74, 2007.
- [4] A. Rezgui and M. Eltoweissy, "Service-oriented sensor-actuator networks: promises, challenges, and the road ahead," *Computer Communications*, vol. 30, no. 13, pp. 2627–2648, 2007.
- [5] J. King, R. Bose, I. Y. Hen, S. Pickles, and A. Helal, "Atlas: a service-oriented sensor platform hardware and middleware to enable programmable pervasive spaces," in *Proceedings of the 31st Annual IEEE Conference on Local Computer Networks (LCN '06)*, pp. 630–638, Tampa, Fla, USA, November 2006.
- [6] Crossbow Technology, <http://www.xbow.com/>.
- [7] J. Li and S. Cheng, " (ϵ, σ) -approximate aggregation algorithms in dynamic sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, pp. 385–396, 2012.
- [8] C. Wang, H. Ma, Y. He, and S. Xiong, "Adaptive approximate data collection for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, pp. 1004–1016, 2012.
- [9] M. C. Vuran, Ö. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: theory and applications for wireless sensor networks," *Computer Networks*, vol. 45, no. 3, pp. 245–259, 2004.
- [10] X. Dong and M. C. Vuran, "Spatio-temporal soil moisture measurement with wireless underground sensor networks," in *Proceedings of the IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net '10)*, pp. 1–8, Juan Les Pins, France, June 2010.
- [11] E. Avilés-López and J. García-Macias, "TinySOA: a service-oriented architecture for wireless sensor networks," *Service Oriented Computing and Applications*, vol. 3, pp. 99–108, 2009.
- [12] J. M. Corchado, J. Bajo, D. I. Tapia, and A. Abraham, "Using heterogeneous wireless sensor networks in a telemonitoring system for healthcare," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 234–240, 2010.
- [13] J. Wang, D. Li, G. Xing, and H. Du, "Cross-layer sleep scheduling design in service-oriented wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 11, pp. 1622–1633, 2010.
- [14] X. Wang, J. Wang, Z. Zheng, Y. Xu, and M. Yang, "Service composition in service-oriented wireless sensor networks with persistent queries," in *Proceedings of the 6th IEEE Consumer Communications and Networking Conference (CCNC '09)*, pp. 1–5, Las Vegas, Nev, USA, January 2009.
- [15] P. Wang and I. F. Akyildiz, "Spatial correlation and mobility-aware traffic modeling for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 19, pp. 1860–1873, 2011.
- [16] Y. Ma, Y. Guo, X. Tian, and M. Ghanem, "Distributed clustering-based aggregation algorithm for spatial correlated sensor networks," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 641–648, 2011.
- [17] S. Patten, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 4, article 24, 2008.

- [18] J. Faruque and A. Helmy, "RUGGED: routing on fingerprint gradients in sensor networks," in *Proceedings of IEEE International Conference on Pervasive Services (ICPS '04)*, pp. 179–188, Beirut, Lebanon, July 2004.
- [19] M. C. Vuran and Ö. B. Akan, "Spatio-temporal characteristics of point and field sources in wireless sensor networks," in *Proceedings of IEEE International Conference on Communications (ICC '06)*, pp. 234–239, Istanbul, Turkey, July 2006.
- [20] D. Zordan, G. Quer, M. Zorzi, and M. Rossi, "Modeling and generation of space-time correlated signals for sensor network fields," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '11)*, pp. 1–6, Houston, Tex, USA, December 2011.
- [21] A. Jindal and K. Psounis, "Modeling spatially correlated data in sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 4, pp. 466–499, 2006.
- [22] X. Han, X. Cao, E. L. Lloyd, and C. C. Shen, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 5, pp. 643–656, 2010.
- [23] T. Banerjee, B. Xie, and D. P. Agrawal, "Fault tolerant multiple event detection in a wireless sensor network," *Journal of Parallel and Distributed Computing*, vol. 68, no. 9, pp. 1222–1234, 2008.
- [24] N. Xiong, A. Vasilakos, L. Yang et al., "Comparative analysis of quality of service and memory usage for adaptive failure detectors in healthcare systems," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 495–509, 2009.
- [25] N. Xiong, A. V. Vasilakos, J. Wu et al., "A self-tuning failure detection scheme for cloud computing service," in *Proceedings of IEEE 26th International Parallel & Distributed Processing Symposium (IPDPS '12)*, pp. 668–679, Shanghai, China, May 2012.
- [26] G. Wu, C. Lin, F. Xia, L. Yao, H. Zhang, and B. Liu, "Dynamical jumping real-time fault-tolerant routing protocol for wireless sensor networks," *Sensors*, vol. 10, no. 3, pp. 2416–2437, 2010.
- [27] J. Bu, M. Yin, D. He, F. Xia, and C. Chen, "SEF: a secure, efficient, and flexible range query scheme in two-tiered sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 126407, 12 pages, 2011.
- [28] 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.
- [29] J. Lin, W. Xiao, F. L. Lewis, and L. Xie, "Energy-efficient distributed adaptive multisensor scheduling for target tracking in wireless sensor networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 6, pp. 1886–1896, 2009.
- [30] M. P. Johnson, H. Rowaihy, D. Pizzocaro et al., "Sensor-mission assignment in constrained environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 11, pp. 1692–1705, 2010.
- [31] Y. Liu, L. Ni, and C. Hu, "A generalized probabilistic topology control for wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 9, pp. 1780–1788, 2012.
- [32] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 14, no. 2, pp. 70–87, 2007.
- [33] W. Zhu and H. Fan, "A discrete dynamic convexized method for nonlinear integer programming," *Journal of Computational and Applied Mathematics*, vol. 223, no. 1, pp. 356–373, 2009.
- [34] W. Zhu and G. Lin, "A dynamic convexized method for nonconvex mixed integer nonlinear programming," *Computers and Operations Research*, vol. 38, no. 12, pp. 1792–1804, 2011.
- [35] M. Schlüter, J. A. Egea, and J. R. Banga, "Extended ant colony optimization for non-convex mixed integer nonlinear programming," *Computers and Operations Research*, vol. 36, no. 7, pp. 2217–2229, 2009.

