Journal of Information Processing Vol. 19 473–490 (Oct. 2011)

Recommended Paper

Probabilistic Coverage Methods in People-Centric Sensing

Asaad Ahmed,[†1,†2] Keiichi Yasumoto,[†1,†4]
Yukiko Yamauchi[†3,†4] and Minoru Ito[†1]

Aiming to achieve sensing coverage for given *Areas of Interest* (AoI) over time at low cost in a *People-Centric Sensing* manner, we propose a concept of (α, T)-coverage of a target field where each point in the field is sensed by at least one mobile node with the probability of at least α during time period T. Our goal is to achieve (α, T)-coverage of a given AoI by a minimal set of mobile nodes. In this paper, we propose two algorithms: *inter-location* algorithm that selects a minimal number of mobile nodes from nodes inside the AoI considering the distance between them and *inter-meeting-time* algorithm that selects nodes regarding the expected meeting time between the nodes. To cope with the case that there is an insufficient number of nodes inside the AoI, we propose an extended algorithm which regards nodes inside and outside the AoI. To improve the accuracy of the proposed algorithms, we also propose an updating mechanism which adapts the number of selected nodes based on their latest locations during the time period T. In our simulation-based performance evaluation, our algorithms achieved (α, T)-coverage with good accuracy for various values of α, T, AoI size, and moving probability.

1. Introduction

Recently, the demand for realtime environmental information about specific regions in urban areas has been increasing for various purposes such as surveillance, navigation, and event detection. People moving inside an urban area offer the possibility of covering a given *area of interest* (AoI) at low cost. Exploiting people as a part of the sensing infrastructure, introduces a new sensing paradigm called *People-Centric Sensing* (PCS) [1]. PCS realizes that people with mobile devices can act as mobile sensors to sense and gather information from the environment to serve sensing applications and their users. In PCS, the coverage depends on the uncontrollable mobility of people, therefore it is difficult to achieve full coverage of the target AoI. Consequently, it is preferable to measure the expected coverage degree as a ratio.

Here, we describe our motivation scenario and our problem settings. The realtime urban sensing scenarios drive an interesting motivating application. In a city sensing application, for instance, users want to know the information in a specific AoI such as interesting spots, crowded places, events on specific locations, and so on. In such an application, a user sends a query about a geographic area as the AoI, a required coverage ratio α (i.e., the required percent of coverage of the AoI), the required information (e.g., noise level), and a query interval (maximum allowable response time) T. Then, the query responding process will be carried out by some people with mobile devices in the AoI, which satisfy the query requirements. Here, to minimize cost, it is desirable to select a minimal number of people with mobile devices that can provide the desired information. We refer to this problem as the (α, T)-*coverage problem*.

In this paper, we formally describe the (α, T)-coverage problem. Given a target field that is composed of a set of points, an AoI as a subset of it, a set of mobile nodes, and a query with a required coverage ratio α and a specified time interval T, we define the problem that finds a minimal set of mobile nodes such that each point in the AoI is visited and sensed by at least one node within T with a probability of at least α. To solve this problem, we need to predict the future locations visited by each mobile node depending on its current location when a query is initiated and its mobility. Thus, we model the mobility of the mobile nodes with a discrete Markov chain. The solution for this problem depends critically on the number and the initial locations of mobile nodes inside and near the target AoI when a query is initiated. One possible solution for this problem is the random selection of nodes. The main drawback of random selection is inefficiency by selecting a set of nodes that are likely to visit the same locations in AoI in the future and this set may not be minimal to achieve the coverage.

†1 Graduate School of Information Science, Nara Institute of Science and Technology
†2 Mathematics Department, Faculty of Science, Al-Azhar University
†3 Graduate School of Information Science and Electrical Engineering, Kyushu University
†4 Japan Science and Technology Agency, CREST.

The initial version of this paper was presented at the SIGMBL conference held on Sep. 2010, which was sponsored by SIGMBL. This paper was recommended to be submitted to IPSJ Journal by the chairman of SIGMBL.

To avoid this drawback, we should carefully select a minimal set of nodes that are not likely to visit the same locations. Based on this insight, we propose two algorithms: *inter-location* and *inter-meeting-time* algorithms, to meet a coverage ratio α in time period T. The inter-location algorithm estimates the probability of locations in the AoI being visited by each mobile sensor node in T, and selects a minimal number of mobile nodes inside the AoI considering the distance between the nodes. The inter-meeting-time algorithm selects a minimal number of nodes regarding the expected time until any two of the nodes will meet at a location. Sometimes, the required coverage may not be achieved due to an insufficient number of nodes existing inside the AoI. To meet the required coverage in this case, we also propose an extended algorithm which takes into account not only nodes existing inside the AoI, but also nodes outside the AoI.

The future estimated location of each node could be inaccurate when T is large, resulting in inaccurate coverage. For more accurate coverage, we propose an updating mechanism for the inter-location and the inter-meeting-time algorithms which aims to remove useless nodes and add some extra nodes that contribute more to AoI coverage. This updating mechanism is periodically executed every specified time interval during T.

We conducted simulation experiments to evaluate the performance of the proposed algorithms for various parameter settings. As a result, we confirmed that the proposed algorithms achieve (α, T)-coverage with good accuracy for a variety of values of α, T, and AoI size, and the inter-meeting time algorithm selected the smallest number of nodes without deterioration of the coverage accuracy.

The rest of this paper is organized as follows. Section 2 reviews the related studies. Section 3 defines the (α, T)-coverage problem. Section 4 describes the proposed algorithms. Section 5 shows the performance evaluation of the proposed algorithms through simulation-based experiments, and finally Section 6 concludes the paper.

2. Related Work

Many studies have proposed data gathering protocols to realize efficient communication between sensor nodes in wireless sensor networks (WSNs) [2)–5)]. Some studies also have proposed the use of mobile sensor nodes in WSNs to improve coverage, lifetime, and/or fault-tolerance [6),7)].

Recently, information collection by pedestrians in PCS has received increasing attentions. PCS is different from existing WSNs because we cannot control the mobility of mobile nodes. In addition, the two important criteria in PCS are coverage of the AoI and time. There are several studies and research projects based on PCS [8)–15)].

Cartel [8)] is a mobile communications infrastructure based on car-mounted communication platforms exploiting open WiFi access points in a city, and provides urban sensing information such as traffic conditions. CitySense [9)] provides a static sensor mesh offering similar types of urban sensing data feeds. SensorPlanet [10)] is a platform that enables the collection of sensor data on a large and heterogeneous scale, and establishes a central repository for sharing the collected sensor data. Bubble-sensing [11)] is a sensor network that allows mobile phone users to create a connection between tasks and places of interest in the physical world. Mobile users are able to affix task bubbles at places of interest and then receive sensed data as it becomes available in a delay-tolerant fashion. PriSense [12)] relies on data slicing and mixing and binary search to enable privacy-preserving queries, where each node slices its data into $(n+1)$ data slices, randomly chooses n other nodes, and sends a unique data slice to each of them. Finally, each node sends the sum of its own slice and the slices received from others to the aggregation server. Anonysense [13)] is a privacy-aware architecture for realizing pervasive applications based on collaborative, opportunistic sensing by personal mobile devices. AnonySense allows applications to query and receive context through an expressive task language and by leveraging a broad range of sensor types on mobile devices, and at the same time respects the privacy of users. GreenGPS [15)] is a navigation service that uses participatory sensing data to map fuel consumption on city streets and find the most fuel-efficient route for vehicles between arbitrary endpoints.

Most of these approaches focus on information collection, but do not consider the probabilistic coverage in PCS when the information collection period is restricted to a short time duration such as an on-demand query. They consider neither the difficulties of achieving sensing coverage of a relatively wide area nor the time requirements of on-demand sensing by mobile users. However, these two criteria are very important in PCS. To meet these criteria, it is also very important to estimate the area covered by each mobile node in a specified time interval. However, existing studies do not consider such a spatiotemporal coverage by mobile nodes.

In Refs. 16) and 17), we formulated the (α, T)-coverage problem and proposed two probabilistic algorithms: an inter-location based algorithm, called ILB, and an inter-meeting-time based algorithm, called IMTB, that consider on-demand sensing by mobile users, and probabilistic coverage in PCS based on the mobility of people. Also, we evaluated the performance of ILB and IMTB for various parameter settings including a realistic scenario on a specific city map. ILB and IMTB algorithms were based only on the initial locations of mobile nodes inside AoI and did not consider the latest locations during the time period. To improve the accuracy of the proposed algorithms, our contribution in this paper is the proposal of two extensions: (i) an updating mechanism for ILB and IMTB algorithms which aims to remove useless nodes and add some extra nodes that contribute more to AoI coverage during the query interval, and (ii) an extended algorithm which regards not only nodes existing inside the AoI, but also nodes near the AoI. In addition, we compare the proposed algorithms with a random selection method to evaluate their performance.

3. The (α, T)-Coverage Problem

In this section, we first describe the models and assumptions for our target PCS application, then formulate the target problem to realize the application.

3.1 Assumptions and Models

3.1.1 System Model

We assume an application such that when requested, some of the mobile users take part in a task to obtain the latest environmental information such as noise level, sunshine intensity, temperature, exhaust gas concentration, and so on, over a specified geographical area of the urban district in a PCS fashion. We assume that those participating users are willing to serve as mobile sensors based on some incentive such as electronic currency or coupons given by a service provider.

We denote the whole service area by A. A road (street) network on which mobile users can move spans the area A. A service user wants to know the approximate condition of a specific area called the *Area of Interest* (*AoI*) produced by obtaining the environmental information about some locations in the AoI. Thus, we assume that there are multiple *sensing locations* with a uniform spacing Δ [⋆1] (e.g., $\Delta = 50$ m) on each road and that sensing coverage is achieved by obtaining the environmental information about all of the sensing locations in the specified AoI. We show an example road network with sensing locations in a service area in **Fig. 1**.

We represent the road network with sensing locations by a connected graph $G = (V, E)$, where V is the set of vertices corresponding to sensing locations (some of them are intersections) and E is the set of edges corresponding to segments between neighboring sensing locations on roads.

Multiple service users of this application exist on service area A and are moving on graph G. Each mobile user is equipped with a portable computing device such as smartphones capable of accessing the Internet via a cellular network (CDMA, GSM) from any place in A, measuring the current location, and sensing the nearby environmental information with its built-in sensors (camera, microphone, light-intensity sensor, etc). Hereafter, we refer to a service user with a mobile device simply as a *node*.

We assume that time progresses discretely $(0, 1, 2, \ldots)$. Let U denote a set of nodes on G at time 0 (i.e., the time when a query is initiated). Each node moves from one vertex to one of its neighboring vertices on G in a unit of time. The mobility of nodes is based on a probabilistic model. Let $v_0^u \in V$ denote the initial (at time 0) location of node u. Let $Prob(u, t, v_0^u, v_t)$ denote the probability that each node u with its location v_0^u at time 0 visits a vertex $v_t \in V$ at time t.

3.1.2 Service Model

We assume that our target application provides users with an on-demand query service for sensing a specific AoI and we treat a single query at a time. We assume that there is a fixed server s in the Internet that can communicate with nodes of U and executes required tasks.

We say that the AoI is *α-covered* if every sensing location in the AoI is visited (and thus the environmental information is sensed) by at least one node with a

⋆1 We assume that each road can be divided into an integer number of segments with length Δ.

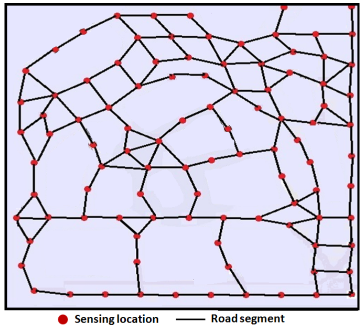


Fig. 1 Example of road network.

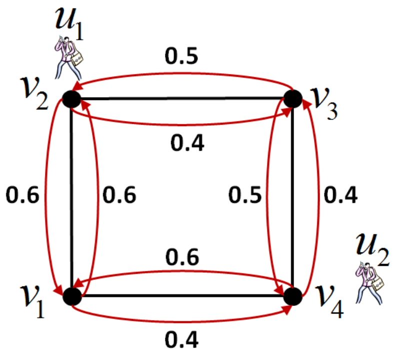


Fig. 2 Moving probability example for four sensing locations $\{v_1, v_2, v_3, v_4\}$ and initial locations of two nodes $\{u_1, u_2\}$.

probability of at least α. Here, we call α the *coverage ratio*. In our application, a node sends s a query which asks for sensing a specified AoI with a specified coverage ratio α in a specified time interval T. We denote each query q by a quadruple $\langle AoI, S_{type}, \alpha, T\rangle$. Here, AoI is the area of interest in the service area specified by a set of sensing locations of V, and S_{type} specifies the type of environmental information to be sensed, such as temperature.

3.2 Problem Formulation

We call the probability of a set of nodes $U'(\subseteq U)$ visiting a sensing location v $(\in V)$ in a time interval T, *the set coverage probability* denoted by $SetProb(v, U', T)$ and define it by the following equation.

$$SetProb(v, U', T) = 1 - \prod_{u\in U'} \prod_{t=0}^{T} (1 - Prob(u, t, v_0^u, v)) \quad (1)$$

Figure 2 shows an example for four sensing locations v_1, v_2, v_3, and v_4 and the moving probabilities between them. As shown in Fig. 2, there are initially two nodes u_1 and u_2 at sensing locations v_2 and v_4, respectively. **Table 1** shows the set coverage probabilities of v_1, v_2, v_3, and v_4 by $U' = \{u_1, u_2\}$ when $T = 2$.

Definition 1. $(\boldsymbol{\alpha}, \boldsymbol{T})$-coverage: *Given a graph $G = (V, E)$, an area specified by a set of sensing locations $AoI \subseteq V$, a set of nodes $U' \subseteq U$, a required coverage ratio α, and a time interval T, the area AoI is called **(α, T)-covered** if the following condition holds.*

$$\forall v \in AoI, SetProb(v, U', T) \geq \alpha \quad (2)$$

Table 1 Visiting time and set coverage probabilities for the example in Fig. 2 with $T = 2$.

Sensing location	**node u_1** $Prob(u_1, t, v_2, v)$			**node u_2** $Prob(u_2, t, v_4, v)$			$SetProb(v, U', 2)$
v	$t=0$	$t=1$	$t=2$	$t=0$	$t=1$	$t=2$	$U' = \{u_1, u_2\}$
v_1	0	0.6	0	0	0.6	0	$1-(1-0.6)(1-0.6) = 0.84$
v_2	1	0	0.56	0	0	0.56	$1-(1-1)(1-0.56)(1-0.56) = 1$
v_3	0	0.4	0	0	0.4	0	$1-(1-0.4)(1-0.4) = 0.64$
v_4	0	0	0.44	1	0	0.44	$1-(1-0.44)(1-1)(1-0.44) = 1$

We formally define the (α, T)-coverage problem as follows:

Definition 2. *Given a service area as a connected graph $G = (V, E)$, a set of nodes U on G at time 0, and a query $q = \langle AoI, S_{type}, \alpha, T\rangle$, the **$(\alpha, T)$-coverage problem** is the problem of selecting a minimal set of nodes $U' \subseteq U$ which achieves (α, T)-coverage of AoI.*

We define the objective function of this problem by the following equation.

minimize $|U'|$ (3)

subject to AoI is (α, T)*-covered* (4)

This problem is NP-hard since it implies, as a special case, the *Minimum Set Covering Problem* (MSCP)[18] which is known to be NP-hard.

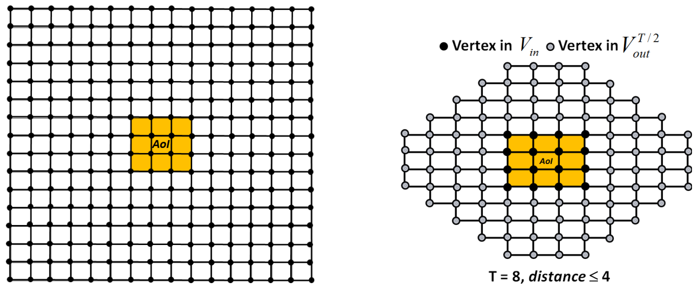
4. Algorithms

In this section, we propose two heuristic algorithms for the problem defined in Section 3, named *Inter-Location Based* (*ILB*) and *Inter-Meeting Time Based* (*IMTB*) algorithms. We assume that all algorithms are executed by the server s in a centralized fashion.

4.1 Preliminaries

Our basic idea is to select nodes that have higher probabilities of visiting distinct sensing locations in the specified AoI within a time interval T, prior to selecting other nodes.

The proposed algorithms depend on the probability $Prob(u, t, v_0^u, v_t)$ of each node u with initial location v_0^u visiting a location v_t at time t $(0 \leq t \leq T)$. To simplify our explanation, we represent the graph $G = (V, E)$ for the service



(a) Original service area graph (b) Reduced service area graph

Fig. 3 An example of a service area graph with AoI and its reduction for $T = 8$.

area by a grid of sensing locations (vertices) with a uniform spacing Δ between neighboring vertices and only vertical and horizontal edges (here, each edge is bi-directional), as shown in **Fig. 3** (a). Let N denote the number of vertices (i.e., $|V|$) and x_i denote the i-th vertex of $V (1 \leq i \leq N)$. We model the node movement on the grid as a discrete Markov chain. For each node u, we define a vector with N states where the i-th state represents the probability that u is in vertex x_i.

Assuming that there are a sufficient number of nodes in target area A, we select nodes only within the specified AoI. Here, at time 0, we are given a query and the current distribution of nodes. Let $U_0(\subseteq U)$ denote the set of nodes which are located in the target AoI at time 0.

4.1.1 Computation of Coverage Probability of a Vertex

Let $\boldsymbol{P}$ denote the probability matrix with size $N \times N$, where its i-th row and j-th column element represents the probability of a node at vertex x_i to move to vertex x_j by a unit of time. We define an initial state vector $\boldsymbol{v_0^u}$ representing that a node u is initially located at $x_i \in V$ by the following equation.

$$\boldsymbol{v_0^u} = (p_1, p_2, \ldots, p_N) \tag{5}$$

where

$$p_j = \begin{cases} 0 & (j \neq i) \\ 1 & (j = i) \end{cases} \tag{6}$$

Then, we can calculate the coverage probability of vertex $x_k \in V$ by node u at time t by the following equation.

$$Prob(u, t, v_0^u, x_k) = [\boldsymbol{v_0^u} \times \boldsymbol{P}^t]_k \tag{7}$$

Here, $[\,]_k$ denotes the k-th element in the resulted vector.

4.1.2 Reduction of Probability Matrix Size

If the target service area contains many sensing locations, the probability matrix $\boldsymbol{P}$ will be large, resulting in a serious computational overhead in the server s. However, we only select nodes in the specified AoI and thus we do not need to consider the nodes which move more than $T/2$ away from the border of the AoI since such nodes never come inside the AoI again. This fact allows us to reduce the size of the probability matrix from $N \times N$ to $(M + L) \times (M + L)$, where M is the number of sensing locations included in the AoI and L is the number of sensing locations outside the AoI such that their shortest distance to the AoI border is at most $T/2$. Here, note that $N \gg M + L$ holds for typical scenarios where *AoI* and T are reasonably small and N is large.

Let $V_{in}(\subseteq V)$ denote a set of vertices included in the AoI. Let $V_{out}(= V - V_{in})$ denote the set of vertices outside the AoI, but in the service area. Let $distance(x, y)$ denote the shortest distance from vertex x to vertex y on G. Let $V_{out}^{T/2}$ denote a set of vertices in V_{out} such that the shortest distance from any vertex of $V_{out}^{T/2}$ to at least one vertex of V_{in} is at most $T/2$. $V_{out}^{T/2}$ is defined by the following equation.

$$V_{out}^{T/2} = \{x \mid x \in V_{out} \wedge \exists y, distance(x, y) \leq T/2 \wedge y \in V_{in}\} \tag{8}$$

The vertices that belong to $V_{out}^{T/2}$ are illustrated in Fig. 3 (b).

We can calculate the coverage probability of all vertices in V_{in} taking into account only the node moving probability at each vertex of $V_{in} \cup V_{out}^{T/2}$. Consequently, we define the new probability matrix $\boldsymbol{P'}$ for vertices of $V_{in} \cup V_{out}^{T/2}$.

We define the i-th row and j-th column element $p'_{i,j}$ of $\boldsymbol{P'}$ by the following equation.

$$p'_{i,j} = \begin{cases} p_{i,j} & (x_i, x_j \in V_{in} \cup V_{out}^{T/2} \wedge i \neq j) \\ \displaystyle\sum_{x \in Ngh(i)} p(i, x) & (x_i \in V_{out}^{T/2} - V_{out}^{T/2-1} \wedge i = j) \end{cases} \tag{9}$$

Here, $Ngh(i)$ is the set of neighboring vertices outside $V_{out}^{T/2}$ and $p_{i,j}$ is the probability of the corresponding edge in the original matrix $\boldsymbol{P}$. Equation (9) represents that the moving probability from x_i to x_j is the same as the original matrix $\boldsymbol{P}$ if i is not equal to j. In addition, knowing that nodes once going outside $V_{out}^{T/2}$ cannot go inside the AoI in T, we for convenience set the probability of a node staying at the same location x_i at border of $V_{out}^{T/2}$ to the sum of probabilities of outgoing edges to outside $V_{out}^{T/2}$.

4.2 The Inter-Location Based Algorithm (ILB)

The ILB uses the distance between nodes as a metric to select a set of mobile nodes. Intuitively, the more distant these nodes are, the more likely it is for these nodes to visit distinct sensing locations of AoI. We denote the distance between the initial locations of nodes u and u' in U_0 by $d_{u,u'}$ which is determined as the length of the shortest path between v_0^u and $v_0^{u'}$ on G. The ILB selects a minimal set of mobile nodes $U'(\subseteq U_0)$ such that the distance between any pair of nodes u and u' in U' is equal to or larger than a threshold d_{th}, and the specified AoI is (α, T)-covered. The above statement is defined as follows.

$$\textbf{minimize } |U'| \textbf{ subject to } (11) - (12) \tag{10}$$

$$d_{u,u'} \geq d_{th}, \forall u, u' \in U' \tag{11}$$

$$\text{AoI is } (\alpha, T)\text{-}\textit{covered} \tag{12}$$

The value of d_{th} should be dependent on three parameters: the total number of time steps T, the required coverage ratio α, and the maximum distance $d_{\max}$ that is the largest distance between the initial locations of two nodes in U_0. Intuitively, as T increases and/or α decreases, the number of selected nodes should decrease. On the contrary, as T decreases and/or α increases, the number of selected nodes must be increased to meet the (α, T)-coverage constraint. To reflect the above relationship among parameters, we define the distance threshold d_{th} by the following equation.

$$d_{th} = \min\left(\frac{T}{\alpha \cdot d_{\max}}, d_{\max}\right) \tag{13}$$

Algorithm 1 *The Inter-location based algorithm (ILB)*

Input: U, AoI, α, T, $G = (V, E)$

Output: U'

1: $U' \leftarrow \emptyset$
2: Compose V_{in}, $V_{out}^{T/2}$, U_0 from AoI and U
3: $\boldsymbol{P} \leftarrow$ ComputeProbMatrix(AoI,$V_{in} \cup V_{out}^{T/2}$)
4: **for** $\forall u \in U_0$ **do**
5: Compose u's initial state vector $\boldsymbol{v_0^u}$
6: **end for**
7: $d_{\max} \leftarrow \max_{u,u' \in U_0} \{d_{u,u'}\}$
8: $d_{th} \leftarrow \min(\frac{T}{\alpha \cdot d_{\max}}, d_{\max})$
9: **while** $SetProb(v, U', T) < \alpha, \forall v \in V_{in}$ **do**
10: **if** $U_0 = \emptyset$ **then**
11: return $\emptyset$
12: **end if**
13: Select $u \in U_0$ at random
14: **if** $U' = \emptyset$ or $\min_{u' \in U'} \{d_{u,u'}\} \geq d_{th}$ **then**
15: $U' \leftarrow U' \cup \{u\}, U_0 \leftarrow U_0 - \{u\}$
16: **end if**
17: **end while**
18: **return** U'

Algorithm 1 shows the node selection process of ILB. The input parameters are the set of mobile nodes U, the area of interest AoI, the required coverage ratio α, the query interval time T, and the service area graph $G = (V, E)$. In line 1, the algorithm initializes U' to be empty. In line 2, it composes the sets of vertices V_{in} and $V_{out}^{T/2}$, and the set of nodes in the AoI, U_0. In line 3, it composes the probability matrix $\boldsymbol{P}$. In lines 4 to 6, it composes the initial state vector for each node $u \in U_0$. In lines 7 and 8, the algorithm determines the maximum distance $d_{\max}$ between nodes in U_0 and the distance threshold d_{th}, as defined in Eq. (13). In lines 9 to 18, the algorithm selects a set of nodes U' as follows: (i) while the AoI is not (α, T)-covered, the algorithm checks the state of U_0 and if U_0 is empty, the algorithm returns $\emptyset$ (i.e., the current U_0 is not sufficient to satisfy the required coverage α), as shown in lines 9 to 12, (ii) the algorithm selects a node $u \in U_0$ at random, as shown in line 13; and (iii) it adds the node

u to the selected set of nodes U' if U' is empty or the distance between u and each node $u' \in U'$ is no less than the threshold d_{th}, as shown in lines 14 to 16. Finally, in line 18, the algorithm returns the selected set of nodes U'.

4.3 The Inter-Meeting Time Based Algorithm (IMTB)

The ILB algorithm is based on the distance between nodes. Hence, the selection process is location-dependent and does not take the query interval time T into consideration. To make the node selection more efficient taking into account the value of T, we propose an inter-meeting time based (IMTB) algorithm which uses the expected first meeting time between nodes as a metric. This meeting time metric reflects the probability of nodes visiting distinct sensing locations of the AoI and describes the expected first meeting time of any pair of nodes $u, u' \in U_0$. Intuitively, as the meeting time between nodes increases, the probability of visiting distinct sensing locations also increases ⋆1 because those nodes explore different locations until they meet for the first time. We denote the expected first meeting time between nodes u and u' in U_0 by $mt_{u,u'}$. The IMTB algorithm selects a minimal set of nodes $U'(\subseteq U_0)$ such that the meeting time $mt_{u,u'}$ between any pair of nodes u and u' in U' is no less than a meeting time threshold mt_{th}, and the specified AoI is (α, T)-covered. The above statement is defined as follows.

$$\textbf{minimize } |U'| \textbf{ subject to } (15) - (16) \tag{14}$$

$$mt_{u,u'} \geq mt_{th}, \forall u, u' \in U' \tag{15}$$

$$\text{AoI is } (\alpha, T)\text{-}\mathit{covered} \tag{16}$$

The values of $mt_{u,u'}$ and mt_{th} are calculated as follows.

The expected first meeting time $mt_{u,u'}$ represents the earliest time when two nodes u and u' in U_0 may meet at some location $v_t \in V_{in}$ and is defined by the following equation.

$$mt_{u,u'} = \begin{cases} \min\limits_{t \in MT_{u,u'}} \{t\} & (MT_{u,u'} \neq \emptyset) \\ T & (MT_{u,u'} = \emptyset) \end{cases} \tag{17}$$

where $MT_{u,u'}$ is a set of possible meeting time between u and u' during the time period T and is defined by the following equation.

$$MT_{u,u'} = \{t \mid Prob(u, t, v_0^u, v_t) > 0 \wedge \ Prob(u', t, v_0^{u'}, v_t) > 0,\ 0 \leq t \leq T, \ \exists\, v_t \in AoI\} \tag{18}$$

The meeting time threshold mt_{th} should be dependent on three parameters: the total number of time steps T, the required coverage ratio α, and the maximum expected first meeting time $mt_{\max}$ between pairs of nodes in U_0. Intuitively, as T increases and/or α decreases, the number of selected nodes will decrease. To reflect the above relationship among parameters, we define the meeting time threshold mt_{th} as follows.

$$mt_{th} = \min\left(\frac{T}{\alpha \cdot mt_{\max}}, mt_{\max}\right) \tag{19}$$

Algorithm 2 shows the node selection process of IMTB. The input parameters are the same as in Algorithm 1. In lines 1 to 6, the algorithm does the same steps as lines 1 to 6 in Algorithm 1. In lines 7 and 8, the algorithm determines the maximum expected first meeting time $mt_{\max}$ between nodes in U_0 and the threshold mt_{th}, as defined in Eq. (19). In lines 9 to 18, the algorithm selects a set of nodes U' as in Algorithm 1, except in line 14, where it adds the node u to the selected set of nodes U' if U' is empty or the expected first meeting time between u and each node $u' \in U'$ is no less than the threshold mt_{th}.

4.4 The Extended Algorithm without Thresholds (EWOT)

As we described in the previous two subsections, the ILB and IMTB algorithms apply the selection process only on a set of nodes located inside AoI at time 0, U_0, and do not consider the nodes outside the AoI. The number of nodes inside the AoI at time 0 may not be sufficient to guarantee the α-coverage of the AoI in time period T, if it is too small.

To cope with this situation, we extend the algorithms to add more nodes located outside the AoI in the selection process based on their contributions to the coverage of the AoI. Here, the contribution of a node means the expected number of locations in the AoI visited by the node during the time period T. The contribution of a node located outside the AoI should be dependent on its initial location and the time period T. In other words, it should be dependent on the shortest distance from the added node to the AoI. Intuitively, if this distance

⋆1 This is not the case if the probability of a node staying at the same location is high, but we suppose the environment where most of the nodes near the AoI are likely to move directly to their destinations.

Algorithm 2 *The Iner-meeting time based algorithm (IMTB)*

Input: U, AoI, α, T, $G = (V, E)$
Output: U'

$U' \leftarrow \emptyset$
Compose V_{in}, $V_{out}^{T/2}$, U_0 from AoI and U
$\boldsymbol{P} \leftarrow$ ComputeProbMatrix(AoI,$V_{in} \cup V_{out}^{T/2}$)
for $\forall u \in U_0$ **do**
  compose u's initial state vector $\boldsymbol{v_0^u}$
end for
$mt_{\max} \leftarrow \max_{u,u' \in U_0}\{mt_{u,u'} : mt_{u,u'} \neq \infty\}$
$mt_{th} \leftarrow \min(\frac{T}{\alpha \cdot mt_{\max}}, mt_{\max})$
while $SetProb(v, U', T) < \alpha, \forall v \in V_{in}$ **do**
  if $U_0 = \emptyset$ **then**
    return $\emptyset$
  end if
  Select $u \in U_0$ at random
  if $U' = \emptyset$ or $\min_{u' \in U'}\{mt_{u,u'}\} \geq mt_{th}$ **then**
    $U' \leftarrow U' \cup \{u\}, U_0 \leftarrow U_0 - \{u\}$
  end if
end while
return U'

of a new added node is more than T, then the node will not visit any locations in the AoI within the time period T. So, the distance must be less than or equal to T. In order to avoid a very large number of added nodes, we only add nodes if the shortest distance to the AoI is less than or equal to $\lfloor \frac{T}{2} \rfloor$. We denote the extended algorithm without thresholds by $EWOT$.

Algorithm 3 shows the node selection process of EWOT. The input parameters are the same as in Algorithms 1 and 2. In line 1, the algorithm initializes U' to empty. In line 2, it composes the sets of vertices V_{in} and $V_{out}^{T/2}$, and the sets of nodes U_0 and $U_0^{T/2}$ (this contains all nodes that initially exist in $V_{out}^{T/2}$). In line 3, it composes the probability matrix $\boldsymbol{P}$. In lines 4 to 6, it composes the

Algorithm 3 *The Extended Algorithm without Thresholds (EWOT)*

Input: U, AoI, α, T, $G = (V, E)$
Output: U'

1: $U' \leftarrow \emptyset$
2: Compose V_{in}, $V_{out}^{T/2}$, U_0, $U_0^{T/2}$ from AoI and U
3: $\boldsymbol{P} \leftarrow$ ComputeProbMatrix(AoI,$V_{in} \cup V_{out}^{T/2}$)
4: **for** $\forall u \in U_0 \cup U_0^{T/2}$ **do**
5:   compose u's initial state vector $\boldsymbol{v_0^u}$
6: **end for**
7: **while** $SetProb(v, U', T) < \alpha, \forall v \in V_{in}$ **do**
8:   **if** $U_0 \neq \emptyset$ **then**
9:     Select u with the highest coverage contribution of U_0
10:     $U' \leftarrow U' \cup \{u\}, U_0 \leftarrow U_0 - \{u\}$
11:   **else if** $U_0 = \emptyset$ **then**
12:     **if** $U_0^{T/2} = \emptyset$ **then**
13:       return $\emptyset$
14:     **end if**
15:     Select u with the highest coverage contribution of $U_0^{T/2}$
16:     $U' \leftarrow U' \cup \{u\}, U_0^{T/2} \leftarrow U_0^{T/2} - \{u\}$
17:   **end if**
18: **end while**
19: **return** U'

initial state vector for each node $u \in U_0 \cup U_0^{T/2}$. In lines 7 to 19, the algorithm selects a set of nodes U' as follows: (i) while the AoI is not (α, T)-covered, if U_0 is not empty, the algorithm selects a node u with the highest coverage contribution of U_0 and adds it to the selected set of nodes U', as shown in lines 7 to 10. (ii) if U_0 is empty (i.e., the current U_0 is not sufficient to satisfy the required coverage α), the algorithm checks the state of $U_0^{T/2}$, and if it is empty, the algorithm returns $\emptyset$, as shown in lines 11 to 14, (iii) the algorithm selects a node u with the highest coverage contribution of $U_0^{T/2}$, as shown in line 15; (iv) it adds the node u to the selected set of nodes U', as shown in line 16. Finally, in line 19,

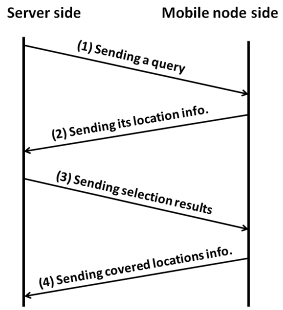


Fig. 4 Query execution sequence between the server and mobile nodes.

the algorithm returns the selected set of nodes U'.

Figure 4 shows the execution sequence process of a query for the proposed algorithms between the server and the mobile nodes. The steps of this execution sequence process as follows: (1) the server sends the query to all mobile nodes that exist in the service area, (2) each mobile node sends its location information to the server, (3) the server performs the selection algorithm steps (ILB, IMTB, or EWOT) and notifies the selected nodes to start their sensing job, and (4) finally, at the end of the query time interval, all selected mobile nodes send the sensed data of their covered locations to the server.

4.5 The ILB and IMTB Algorithms with Updating Mechanism

As we described in the previous two subsections, the ILB and IMTB algorithms are based only on the initial locations of nodes inside the AoI and do not consider the latest locations of nodes during the query period T. There can be a scenario that some nodes initially exist in the AoI and may go out of the AoI after some time during the query period T. Also, some nodes may initially exist outside of the AoI and may go into the AoI after some time during the query period T. If we track the location of nodes in and near the AoI during period T, we can achieve more accurate coverage with lower cost by removing useless nodes and adding some extra nodes that contribute more coverage. For more accurate coverage, we propose an updating mechanism for the ILB and IMTB algorithms that aims to adapt the number of selected nodes based on the latest location of nodes. This updating mechanism is executed every specified time interval during the time period T.

Let $t_{current}$ denote the current time step. The updating mechanism consists of the following steps

(1) Calculate the remaining required coverage ratio, β ⋆1 ($\beta = \alpha - \gamma$, where γ is the already achieved coverage ratio).

(2) Estimate the coverage probability for all uncovered locations in the AoI by the nodes in U' if their current locations exist in the AoI by using the ILB or IMTB algorithms.

(3) If the estimated coverage probability is less than β, then one-by-one add a new node to the selected set while all locations in the AoI are (β, $T - t_{current}$)-covered.

(4) If the estimated coverage probability is larger than β, then one-by-one remove a node from U' as long as all uncovered locations in AoI are (β, $T - t_{current}$)-covered.

By using this updating mechanism, the ILB and IMTB algorithms can adapt the number of selected nodes by adding or removing some nodes to improve the accuracy of the coverage probability of the AoI as much as possible. This updating mechanism is executed periodically every specified time interval which is called the *updating interval* UI.

It is preferable to determine the value of UI internally. In other words, it should be dependent on T and α. So, we use the distance threshold d_{th} and the meeting time threshold mt_{th} of ILB, and IMTB, respectively to determine the value of UI as follows.

$$UI = \begin{cases} d_{th} & \text{for} \quad ILB \\ mt_{th} & \text{for} \quad IMTB \end{cases} \tag{20}$$

We refer to the ILB and IMTB with the updating mechanism by *ILB-up* and *IMTB-up*, respectively.

4.6 Complexity

Here, we evaluate the computing time of the proposed algorithms according to

⋆1 Here, to minimize the total overhead, β represents the maximum deficit coverage ratio among all locations in AoI.

the size of matrix $\boldsymbol{P}$, $(M+L)^2$, the number of nodes in U_0 (i.e., inside AoI), n, and the total number of steps, T. According to the coverage probability of a vertex which is defined in Eq. (7), the computing time of ILB and IMTB is $\boldsymbol{O}\ (n \cdot (M+L)^2 \cdot T)$, while the computing time for EWOT is $\boldsymbol{O}((n+n') \cdot (M+L)^2 \cdot T)$, where n' is the number of nodes in $U_0^{T/2}$ (i.e., outside AoI).

5. Performance Evaluation

In this section, we show the results of simulation experiments that examine the coverage performance of the proposed algorithms in terms of the number of nodes selected and the accuracy of the achieved coverage ratio. We compared the proposed algorithms with the random selection method which repeats selecting a node randomly among all nodes inside the AoI until satisfying (α, T)-coverage and does not use any distance or time thresholds.

5.1 Simulation Environment

The QualNet [19] simulator was used with the input parameters listed in **Table 2**, such as service area size, number of nodes, node speed, etc. In addition, the node mobility was based on a discrete Markov model as described in Section 4. The service area was represented as a grid of sensing locations arranged with uniform spacing, 50 *meters*. We selected the AoI as a rectangular region where its position was selected at random within the service area. The ratio of its size to the service area size, called *AoI-Size*, was selected from {0.01, 0.25, 0.45, 0.5, 0.65, 0.85} and the corresponding number of sensing locations in each AoI was {4, 36, 56, 66, 77, 99}. The initial node location was selected at random among all sensing locations in the service area. We repeated every simulation experiment 5 times with different initial node distributions, then averaged the results.

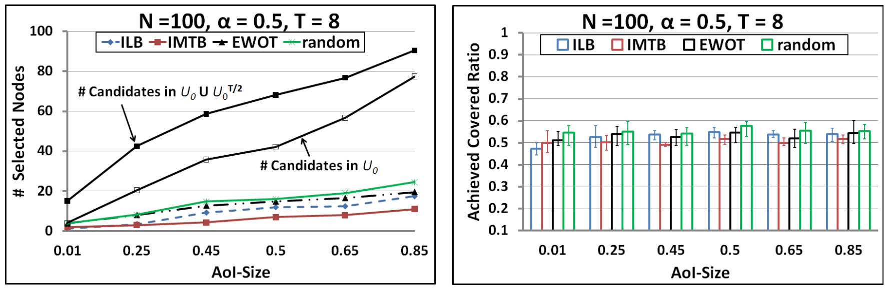
Table 2 Configuration parameters.

Configuration parameter	Value in simulation
# nodes	25 to 200
Node speed	1 m/s
Field size	500 m × 500 m
Required coverage, α	0.2, 0.4, 0.5, 0.6, 0.8, 0.9
Total # sensing locations in A	121
AoI-Size (# sensing locations)	0.01 (4), 0.25 (36), 0.45 (56), 0.5 (66), 0.65 (77), 0.85 (99)
Δ	50 m
Total # steps (time period), T	2, 4, 6, ..., 20

We measured the performance of the proposed algorithms in terms of the number of selected nodes and the achieved coverage ratio by changing the number of nodes, the AoI-Size, the total number of time steps (query interval time), and the required coverage ratio. Here, we define the achieved coverage ratio as the ratio of the number of sensing locations visited in the AoI by at least one node to the total number of sensing locations in the AoI. We say that the algorithms satisfy the required coverage ratio if the average achieved coverage ratio of several simulation runs is no less than the required ratio.

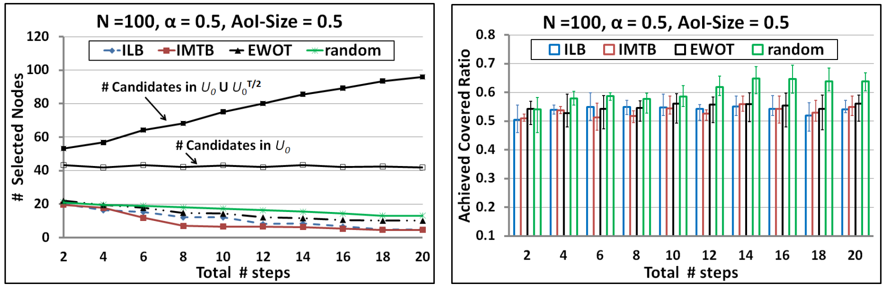
5.2 Simulation Results without Updating Mechanism

In this section we show the simulation results for the proposed algorithms without the updating mechanism in two cases. In the first case, the moving probabilities of a node at a location to its neighboring locations were equal probabilities (i.e., uniform and equal to 0.25). To show the performance of the proposed algorithms under non-uniform moving probabilities, in the second case, the moving probabilities of a node at a location to its neighboring locations were unequal probabilities. We show simulation results in **Figs. 5**, **6**, **7**, **8**, **9**, **10** and **11** (The black lines with empty and solid rectangles in Figs. 5–11 (a) represent the number of candidate nodes in U_0 and $U_0 \cup U_0^{T/2}$, respectively).



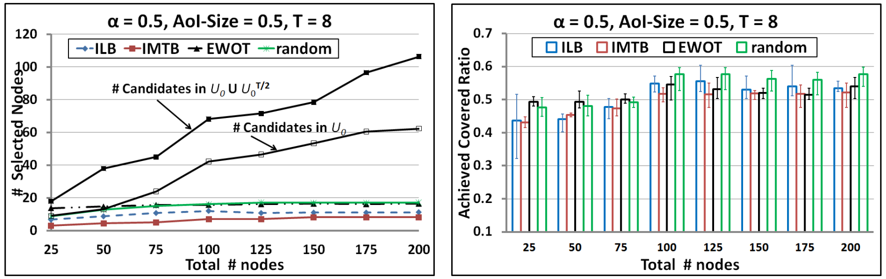
(a) Number of selected nodes (b) Achieved coverage ratio

Fig. 5 Performance for different AoI sizes.



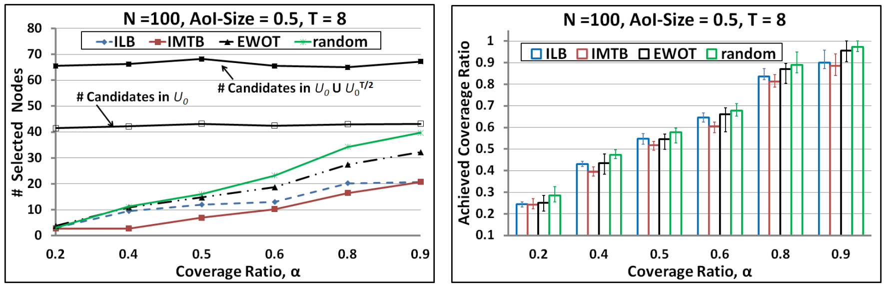
(a) Number of selected nodes (b) Achieved coverage ratio

Fig. 6 Performance for different time steps.



(a) Number of selected nodes (b) Achieved coverage ratio

Fig. 7 Performance for different number of nodes.



(a) Number of selected nodes (b) Achieved coverage ratio

Fig. 8 Performance for different required coverage ratios.

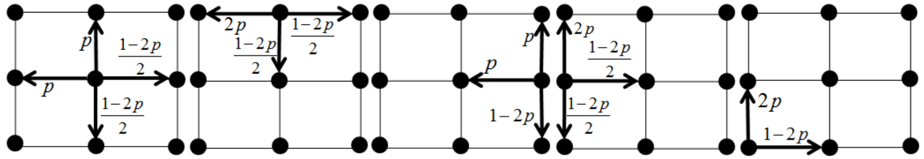
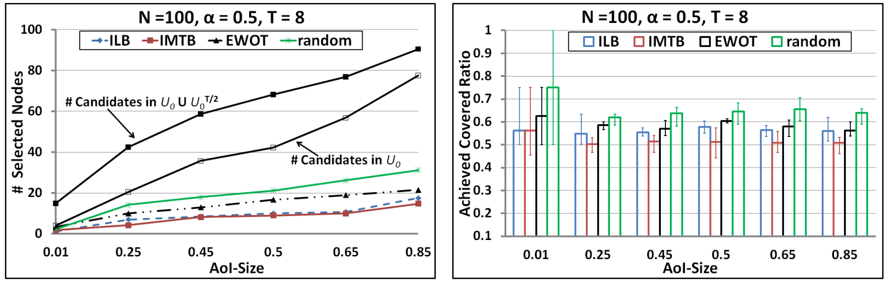
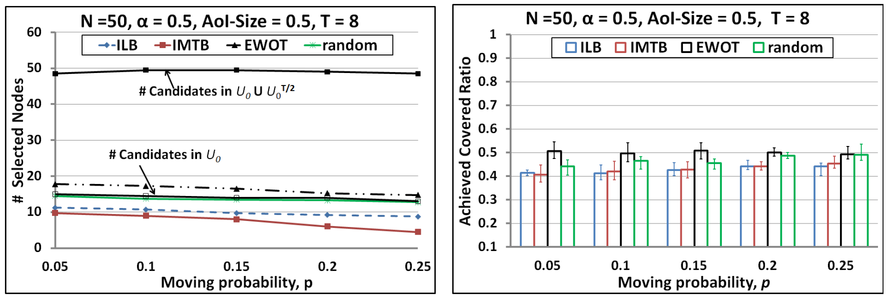


Fig. 9 Moving probabilities cases for probability p.



(a) Number of selected nodes (b) Achieved coverage ratio

Fig. 10 Performance for different AoI sizes with random moving probability.



(a) Number of selected nodes (b) Achieved coverage ratio

Fig. 11 Performance for different moving probabilities, p.

5.2.1 Results for Equal Moving Probabilities

Figure 5 shows the performance for different AoI-Size in the case of medium required coverage ratio and medium number of time steps. The number of nodes was 100, the required coverage α was 0.5, and the total number of steps was 8. In

Fig. 5 (a), the number of selected nodes increased as the AoI-Size increased. This is because, when the AoI-Size increased, we needed more nodes to satisfy the required coverage ratio. As shown in Fig. 5 (a), the number of selected nodes for the proposed algorithms was much smaller than the number of candidates nodes in the AoI and it was reduced by 75.46%, 79.77%, and 57.06% for ILB, IMTB, and EWOT, respectively, while for the random algorithm, its reduction was 53.62%. In Fig. 5 (b), the required coverage was almost satisfied by all algorithms. When the AoI-Size was 0.01, the number of selected nodes and the variance of ILB was smaller than other algorithms. For a larger AoI-Size, the number of selected nodes and the variance of IMTB were smaller than other algorithms. As a result, for a smaller values of the AoI-Size, the ILB is the best, while the IMTB is the best in the case of a larger AoI-Size.

Figure 6 shows the performance for different numbers of time steps with a medium size AoI and a medium required coverage ratio. The AoI-Size was 0.5. In Fig. 6 (a), the number of selected nodes decreased as the total number of steps increased. This is because the distance and meeting time threshold increases in proportion to the total number of steps. The number of selected nodes for IMTB was lower than other algorithms since the required coverage is medium and the meeting time increased when the AoI-Size is medium. The number of selected nodes was reduced by 74.55%, 79%, and 66.6% of the number of candidates nodes in the AoI for ILB, IMTB, and EWOT, respectively, while for random algorithm, its reduction was 60.83%. In Fig. 6 (b), all algorithms satisfied the required coverage and the variance of IMTB was smaller than other algorithms.

Figure 7 shows the performance for different numbers of nodes with a medium size AoI, a medium required coverage ratio, and a medium number of time steps. The AoI-Size was 0.5. In Fig. 7 (a), when the number of nodes was 25 to 125, the number of selected nodes increased as the number of nodes increased. This is because, when the number of nodes increases, the algorithms add more nodes to satisfy the required coverage. However, when the number of nodes was larger than 125, the number of selected nodes was fixed since the number of selected nodes is bound by the number of nodes needed to satisfy the required coverage. Also, when the number of nodes was 25 to 75, the number of selected nodes was reduced by 49.93% of the number of candidate nodes inside and outside the AoI for EWOT. For a larger number of nodes, it was reduced by 78.35%, 85.23%, 68.74%, and 67.38% of the number of candidate nodes in the AoI for ILB, IMTB, EWOT, and random algorithms, respectively. In Fig. 7 (b), the required coverage was not satisfied by ILB, IMTB, and random algorithms when the total number of nodes was 25 to 75, while the EWOT algorithm satisfied the required coverage with small variance. This is because the EWOT algorithm takes into account nodes that also exist outside the AoI and it can add more nodes to meet the required coverage. For a larger number of nodes, all algorithms satisfied the required coverage and the variance of IMTB was smaller than other algorithms. As a result, for a small number of nodes inside AoI, the EWOT is the best, while the IMTB is the best in the case of a larger number of nodes.

Figure 8 shows the performance for different required coverage ratio with a medium size AoI and a medium number of time steps. The AoI-Size was 0.5. In Fig. 8 (a), the number of selected nodes increased as the required coverage ratio increased. This is because, as the required coverage ratio increases, we need more nodes to satisfy it. The number of selected nodes for IMTB was lower than other algorithms and it was reduced by 76.68% of the number candidate nodes in the AoI. On the other hand, it was reduced by 69.46%, 58%, and 50.45% for ILB, EWOT, and random algorithms, respectively. In Fig. 8 (b), the required coverage was satisfied by all algorithms and the variance of IMTB is smaller than other algorithms.

Here, we summarize the simulation results as follows.

- ILB, IMTB, and EWOT algorithms reduce the number of selected nodes to a great extent for (α, T)-coverage compared to the number of candidate nodes in the AoI.
- For a small AoI, ILB can select a smaller number of nodes to meet the required coverage with a smaller variance than IMTB, EWOT, and random algorithms.
- For medium and large AoI, IMTB can select a smaller number of nodes to meet the required coverage with a smaller variance than ILB, EWOT, and random algorithms.
- When only a small number of nodes are initially located in the AoI, only the EWOT algorithm can meet the required coverage.

5.2.2 Results for Unequal Moving Probabilities

In the real environment, the moving probabilities of a node at any location to its neighboring one are almost unequal. In order to investigate to what extent the unequalness of the moving probability affects the performance of the proposed methods, we conducted simulations according to the following two scenarios.

- **a) random moving probabilities:** in this scenario, the moving probability of a node at a location i to one of its neighboring locations is determined randomly between 0.01 and 0.09 such that the sum of all moving probabilities to its neighboring locations is equal to 1.
- **b) baised moving probability p:** in this scenario, we constructed a model by defining a moving probability parameter p as shown in Fig. 9. In the simulations, the value of p was selected from $\{0.05, 0.1, 0.15, 0.2, 0.25\}$. Based on this model, when p is small, most of the nodes are likely to move towards a specific direction with higher probabilities (e.g., towards bottom right corner). Here, $p = 0.25$ corresponds to the case of equal moving probability.

Figure 10 shows the performance for a different AoI-Size by using random moving probabilities. The number of nodes was 100, the required coverage α was 0.5, and the total number of steps was 8. In Fig. 10 (a), the trend on the number of selected nodes was almost similar to the case of Fig. 5 (a), but more nodes were selected. This is because, in this scenario, the probability matrix is not uniform and there are a smaller number of nodes that visit some sensing locations in AoI. In Fig. 10 (b), the required coverage is almost satisfied by all algorithms.

Figure 11 shows the performance for different values of p. Where a sufficient number of nodes exists in the AoI, a clear impact of p value may not occur on the AoI coverage. So, it is preferable to evaluate the performance of the proposed algorithms when there is an insufficient number of nodes inside the AoI. Therefore, in this simulation, the number of nodes was 50. In Fig. 11 (a), the number of selected nodes decreased as p increased. This is because, when p increases, the expected number of different visited locations for each node increases. In Fig. 11 (b), the required coverage was satisfied only by EWOT. This is because, there is an insufficient number of nodes inside the AoI. Also, EWOT reduced the number of selected nodes by 66.74% of the number of candidate nodes inside and outside the AoI. The variance of ILB, IMTB, and random algorithms decreased as p increased. This is because, when p increases the nodes tend to move in different directions and the expected number of different covered locations increases. As a result, if there is an insufficient number of nodes inside the AoI and most of the nodes tend to move towards a specific direction, the EWOT algorithm is the best among all algorithms.

5.2.3 Traffic Overhead and Resource Consumption Ratio

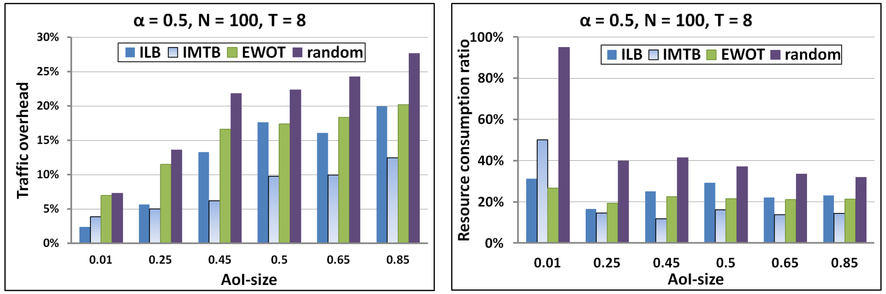
In this section, we show the traffic overhead and the efficiency in mobile nodes' resources consumption of the proposed algorithms.

The main objectives of the proposed node selection algorithms are (i) achieving AoI coverage ratio α in time period T and (ii) minimizing the overall cost consisting of (1) the incentive fees (determined depending on resource consumption of mobile nodes for sensing and uploading) paid to the nodes that perform sensing and uploading and (2) the total traffic amount.

Here, we defined the method in which the node selection is not performed and all candidate nodes are performing the sensing and uploading operations as the standard method. According to the query execution sequence in Fig. 4, to execute the standard method, only two processes (1) and (4) are needed. On the other hand, to execute the proposed algorithms, the four processes (1), (2), (3), and (4) are needed. Here, the traffic overhead of the proposed algorithms depends on the number of mobile nodes in each process of the query execution sequence. Now, we will explain how to measure the traffic overhead as follows.

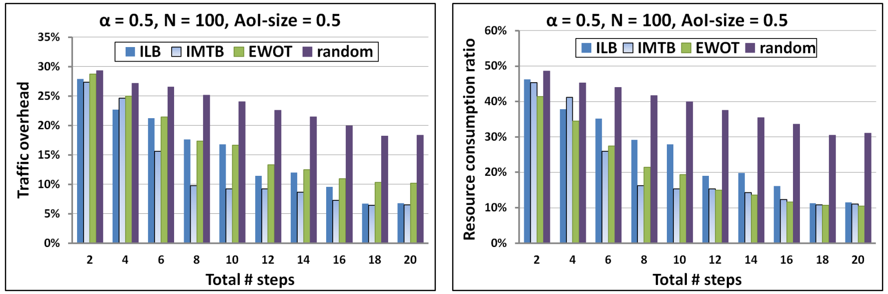
Let N denote the total number of mobile nodes in service area A, B denote the number of all candidates in the AoI, and C denote the number of selected candidates (for the proposed algorithms). In the case of the standard method, the number of mobile nodes in (1) and (4) are N and B, respectively. So, the total traffic (in terms of the number of transmitted messages) for the standard method is equal to $N+B$. In the case of the proposed algorithms, the number of mobile nodes in (1), (2), (3), and (4) are N, B, C, and C, respectively. So, the total traffic (in terms of the number of transmitted messages) for the proposed algorithms is equal to $N + B + 2C$. We will define the traffic overhead for the proposed algorithms as follows.

$$\textit{traffic overhead} = \frac{2C}{N+B} \tag{21}$$



(a) Traffic overhead (b) Resource consumption ratio

Fig. 12 Traffic overhead and resource consumption ratios for different AoI sizes.



(a) Traffic overhead (b) Resource consumption ratio

Fig. 13 Traffic overhead and resource consumption ratio for different values of steps.

Also, to show the efficiency in mobile nodes' resource consumption of the proposed algorithms compared to the standard method, we defined the resource consumption ratio of the number of selected candidates to the total number of candidates existing in AoI as follows.

$$Resource\ consumption\ ratio = \frac{C}{B} \tag{22}$$

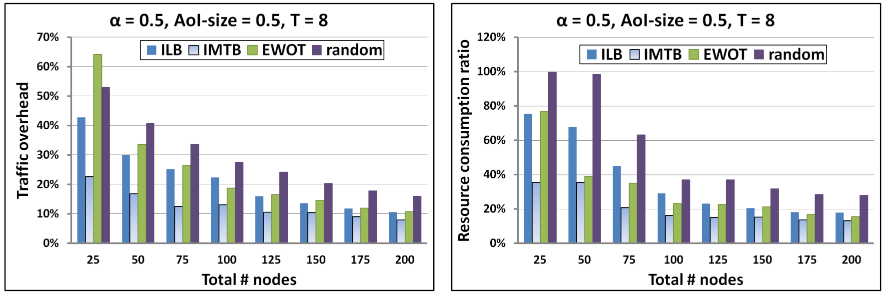
According to Eqs. (21) and (22), we measured the traffic overhead and the resource consumption ratio for ILB, IMTB, EWOT, and random algorithms compared to the standard method for different values of AoI size, number of steps, number of nodes, and required coverage in the case of the equal moving probabilities scenario.

Figure 12, shows the traffic overhead and the resource consumption ratio for a different AoI-Size. The number of nodes was 100, the required coverage α was 0.5, and the total number of steps was 8. In Fig. 12 (a), the traffic overhead increased as the AoI-Size increased. When the AoI-Size was 0.01, the traffic overhead of ILB was lower than other algorithms and it was 2%. For a larger AoI-Size, the traffic overhead of IMTB was lower than other algorithms and it increased from 5% to 12.4% as the AoI size increases. In Fig. 12 (b), the resource consumption ratio of the proposed algorithms was much lower than the standard method. When the AoI-Size was 0.01, the resource consumption ratio of ILB and EWOT was lower than other algorithms and it was 31% and 27%, respectively. For a larger AoI-Size, the resource consumption ratio of IMTB was lower than other algorithms and it was between 12% to 16%.

Figure 13 shows the traffic overhead and the resource consumption ratio for a different numbers of time steps. The AoI-Size was 0.5. In Fig. 13 (a), the traffic overhead decreased as the number of steps increased. The traffic overhead of IMTB was lower than other algorithms and it decreased from 27% to 6% as the number of steps increases. In Fig. 13 (b), the resource consumption ratio of the proposed algorithms was much lower than the resource consumption ratio in the case of the standard method. The resource consumption ratio of IMTB was lower than other algorithms in most values of time steps and it was between 11% and 45%.

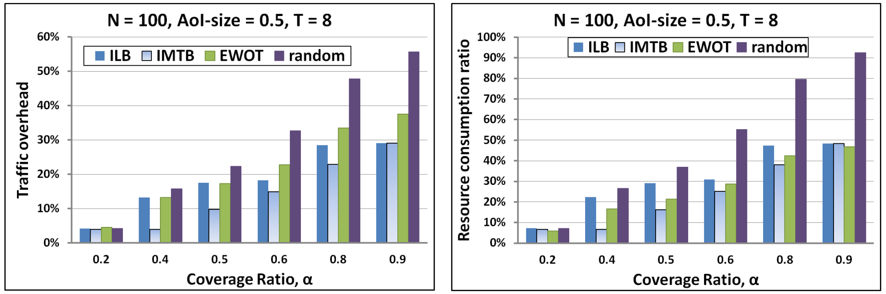
Figure 14 shows the traffic overhead and the resource consumption ratio for a different numbers of nodes. In Fig. 14 (a), the traffic overhead decreased as the number of nodes increased. The traffic overhead of IMTB was lower than other algorithms and it decreased from 23% to 8% as the number of nodes increases. In Fig. 14 (b), the resource consumption ratio of the proposed algorithms was lower than the standard method. The resource consumption ratio of IMTB was lower than other algorithms and it was between 13% and 36%.

Figure 15 shows the the traffic overhead and the resource consumption ratio for a different required coverage ratio. In Fig. 15 (a), the traffic overhead increased as the required coverage ratio increased. The traffic overhead of IMTB was lower



(a) Traffic overhead (b) Resource consumption ratio

Fig. 14 Traffic overhead and resource consumption ratio for different number of nodes.



(a) Traffic overhead (b) Resource consumption ratio

Fig. 15 Traffic overhead and resource consumption ratio for different values of required coverage, α.

than other algorithms and it increased from 4% to 29% as the required coverage ratio increases. In Fig. 15 (b), the resource consumption ratio of the proposed algorithms was lower than the standard method. The resource consumption ratio of IMTB was almost lower than other algorithms and it was between 7% and 48%.

5.2.4 Sensitivity of P Matrix

In this section, we study the sensitivity of matrix P on the performance of the proposed algorithms by adding a noise parameter called σ to the moving probability matrix P as follows.

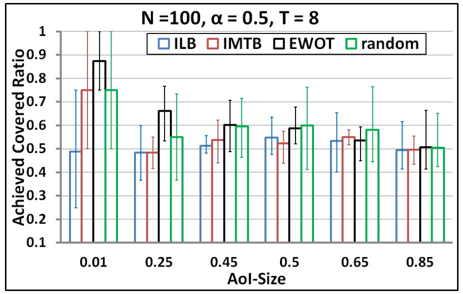


Fig. 16 Effect of σ on the achieved coverage ratio for different values of AoI size.

$$P = (p_{i,j} \pm \sigma_k), k = 0, 1, 2, \ldots, T-1, s.t. \sum_{j=1}^{M+L} (p_{i,j} \pm \sigma_k) = 1, 1 \leq i \leq M+L \tag{23}$$

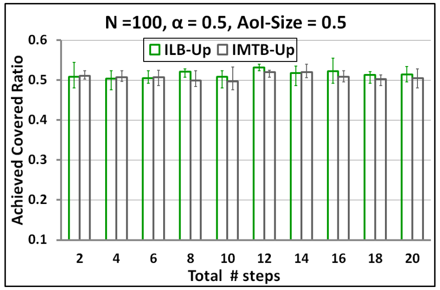
According to Eq. (23), we conducted simulation experiments to show the impact of σ on the performance of the proposed algorithms against the AoI-size. This is because the size of matrix P depends on the AoI-size and the impact of σ will be more visible. The value of σ was randomly selected between 0 and 1.

As shown in **Fig. 16**, the performance of all algorithms was affected by the value of σ where the variance of all algorithms was bigger than the variance of all algorithms in the case of Fig. 5 (b). In addition, the IMTB still achieved the lowest variance compared to other algorithms.

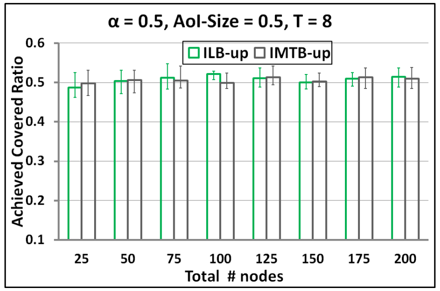
5.3 Simulation Results with Updating Mechanism

In this section, we show the simulation results which we conducted for ILB-up and IMTB-up. We measured the performance of ILB-up and IMTB-up in terms of the number of selected nodes, the achieved coverage ratio, the total number of sensing times, and the communication overhead. In the simulations, the required coverage α was 0.5 and the AoI-Size was 0.5. In order to evaluate the overhead of the updating mechanism, we define the total number of sensing times as the total number of times at which the selected nodes perform a sensing action. It is defined as follows.

$$totalSensingTimes = \sum_{u \in C} nst_u, \quad C = \bigcup_{t \in UT} C_t \tag{24}$$

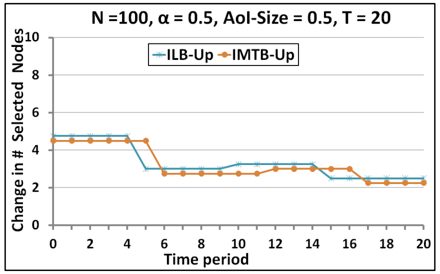


(a) Achieved coverage vs. Total # of steps

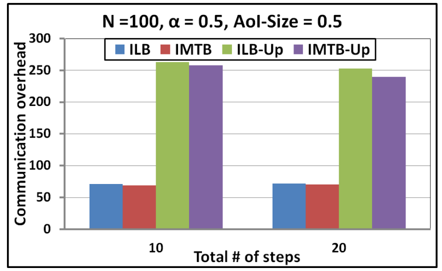


(b) Achieved coverage vs. Total # of nodes

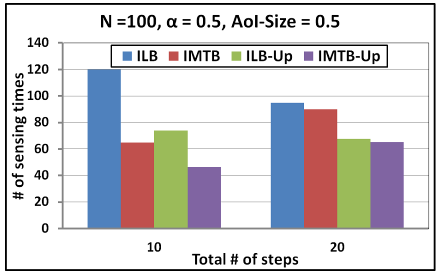
Fig. 17 Coverage performance for updating algorithms.



(a) Change in number of selected nodes during time period, total number of steps = 20



(b) Communication overhead when total number of steps is 10 and 20



(c) Number of sensing times when total number of steps is 10 and 20

Fig. 18 Change in number of selected nodes, communication overhead, and number of sensing times for updating algorithms.

where, nst_u is the number of sensing times of a node u, C is the set of all selected nodes during the time period T, C_t is the set of selected nodes at updating time t (C_0 represents the initial selected set), and UT is the set of updating times.

Also, we define the communication overhead as the total number of candidate nodes for all updating times during the time period. It is defined as follows.

$$ComOverhead = \sum_{t \in UT} candidates(t) \tag{25}$$

where, $candidates(t)$ is the number of candidate nodes at time t. Here, the candidate nodes are the nodes inside the AoI or within distance $(T-t)$ from the AoI border. We show the simulation results in **Figs. 17** and **18**.

Figure 17 (a) shows the performance for a different numbers of time steps with a medium size AoI and a medium required coverage ratio. The number of nodes was 100. As shown in Fig. 17 (a), the required coverage was satisfied by ILB-up and IMTB-up. The accuracy of ILB-up and IMTB-up was better than ILB and IMTB in Fig. 6 (b) and their variances were lower than ILB and IMTB. This is because, ILB-up and IMTB-up adapt the number of selected nodes during the time period and ILB and IMTB do not. Figure 17 (b) shows the performance for a different numbers of nodes with a medium size AoI and a medium required coverage ratio. The number of time steps was 8. While ILB and IMTB did not satisfy the required coverage ratio when the total number of nodes was 25 to 75 (Fig. 7 (b)), ILB-up and IMTB-up satisfied the ratio thanks to the update mechanism. For a larger number of nodes, all algorithms satisfied the required coverage.

Figure 18 (a) shows the change in the number of selected nodes during a time period when T was 20. As shown in Fig. 18 (a), ILB-up and IMTB-up adapt the number of selected nodes by adding or removing nodes to improve the accuracy as much as possible during the time period. Figures 18 (b) and 18 (c) show the communication overhead and the total number of sensing times when the time period was 10 and 20 steps. In Fig. 18 (b), the communication overhead for ILB-up and IMTB-up was larger than ILB and IMTB since ILB-up and IMTB-up requires all nodes to communicate in and near the AoI at each update time.

In Fig. 18 (c), the total number of sensing times for ILB-up and IMTB-up was smaller than ILB and IMTB since the update mechanism selects only necessary nodes taking into account the already covered sensing locations at each update time.

In conclusion, the update mechanism can be used for applications that require a high accuracy of AoI coverage and are not concerned with the communication overhead. On the other hand, if low communication overhead is required, it is better to use ILB and IMTB without the update mechanism.

6. Conclusion

In this paper, we tackled the (α, T)-coverage problem in people centric sensing with a motivating application scenario. We formulated this problem as an optimization problem with the objective of minimizing the number of selected nodes to meet the demanded coverage ratio α within a query interval time T. To resolve this problem, we proposed heuristic algorithms.

Our simulation results showed that the proposed algorithms achieved (α, T)-coverage with good accuracy for a variety of values of α, T, AoI size, and moving probability, and that the inter-meeting time based algorithm selects a smaller number of nodes without deteriorating coverage accuracy. Also, the proposed algorithms reduce the cost (number of sensing times) to a great extent compared to the case of selecting all nodes in the AoI. In addition, our updating mechanism adapts the number of selected nodes by removing useless nodes and adding some extra nodes that contribute more to AoI coverage.

In this paper, we considered only the case where a single query is issued at a time. In future work, we will try to make the proposed algorithms adaptive in the case of multiple simultaneous queries to minimize the overhead.

References

1) Campbell, A., Eisenman, S., Lane, N., Miluzzo, E., Peterson, R., Lu, H., Zheng, X., Musolesi, M., Fodor, K. and Ahn, G.: The Rise of People-Centric Sensing, *IEEE Internet Computing Special Issue on Mesh Networks*, Vol.12, No.4, pp.12–21 (2008).
2) Al-Karaki, J.N. and Kamal, A.E.: Routing Techniques in Wireless Sensor Networks: A survey, *IEEE Wireless Communications*, Vol.11, No.6, pp.6–28 (2004).
3) Akkaya, K. and Younis, M.: A survey on Routing Protocols for Wireless Sensor Networks, *IEEE Wireless Communications*, Vol.3, No.3, pp.325–349 (2005).
4) Lima, L. and Barros, J.: Random Walks on Sensor Networks, *Proc. 5th Intl. Syposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks*, pp.1–5 (2007).
5) Shah, R., Roy, S., Jain, S. and Brunette, W.: Data mules: Modeling a three-tier architecture for sparse sensor networks, *Proc. IEEE Workshop on Sensor Network Protocols and Applications*, pp.30–41 (2003).
6) Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K. and Ito, M.: Extending k-Coverage Lifetime of Wireless Sensor Networks Using Mobile Sensor Nodes, *Proc. 5th IEEE Intl. Conf. Wireless and Mobile Computing, Networking and Communications* (*WiMob'09*), pp.48–54 (2009).
7) Wang, W., Sirinivasan, V. and Chua, K.: Trade-offs Between Mobility and Density for Coverage in Wireless Sensor Networks, *Proc. 5th IEEE Intl. Conf. Mobile Computing and Networking* (*MobiCom'07*), pp.39–50 (2007).
8) Hull, B., Bychkovsky, V., Zhang, Y., Chen, K., Goraczko, M., Miu, A., Shih, E., Balakrishnan, H. and Madden, S.: CarTel: A Distributed Mobile Sensor Computing System, *Proc. 4th Intl. Conf. Embedded Networked Sensor Systems* (*SenSys'06*), pp.125–138 (2006).
9) Murty, R., Gosain, A., Tierney, M., Brody, A., Fahad, A., Bers, J. and Welsh, M.: CitySense: A Vision for an Urban-Scale Wireless Networking Testbed, *Proc. Intl. Conf. Technologies for Homeland Security*, pp.583–588 (2008).
10) Tuulos, V., Scheible, J. and Nyholm, H.: Combining Web, Mobile Phones and Public Displays in Large-Scale: Manhattan Story Mashup, *Proc. 5th Intl. Conf. Pervasive Computing*, pp.37–54 (2007).
11) Lua, H., Lane, N., Eisenmanb, S. and Campbella, A.: Bubble-sensing: Binding Sensing Tasks to The Physical World, *Journal of Pervasive and Mobile Computing*, Vol.6, No.1, pp.58–71 (2010).
12) Shi, J., Zhang, R., Liu, Y. and Zhang, Y.: PriSense: privacy-preserving data aggregation in people-centric urban sensing systems, *Proc. INFOCOM'10*, pp.758–766 (2010).
13) Cornelius, C., Kapadia, A., Kotz, D., Peebles, D., Shin, M. and Triandopoulos, N.: Anonysense: Privacy-aware people-centric sensing, *Proc. MobiSys'08*, pp.211–224 (2008).
14) Abdelzaher, T.: Mobiscopes for human spaces, *IEEE Pervasive Computing*, Vol.6, No.2, pp.20–29 (2007).
15) Ganti, R.K., Pham, N., Ahmadi, H., Nangia, S. and Abdelzaher, T.F.: GreenGPS: A participatory sensing fuel-efficient maps application, *Proc. MobiSys'10*, pp.151–164 (2010).
16) Ahmed, A., Yasumoto, K., Yamauchi, Y. and Ito, M.: Probabilistic Methods for Spatio-Temporal Coverage in People-Centric Sensing, *IPSJ SIG Technical Reports 2010-MBL-55*, Vol.3, pp.1–8 (2010).

17) Ahmed, A., Yasumoto, K., Yamauchi, Y. and Ito, M.: Distance and Time Based Node Selection for Probabilistic Coverage in People-Centric Sensing, *Proc. 8th Annual IEEE Intl. Conf. Sensor, Mesh and Ad Hoc Communications and Networks* (*SECON'11*), pp.134–142 (2011).
18) Vercellis, C.: A Probabilistic Analysis of The Set Covering Problem, *Annals of Operations Research*, Vol.1, No.3, pp.255–271 (1984).
19) Scalable Network Technologies Inc.: *Qualnet 4.0 network simulator* (2007).

(Received February 22, 2011)
(Accepted July 8, 2011)
(Released October 5, 2011)

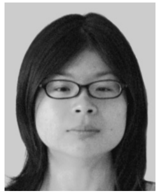
Editor's Recommendation

This paper aims at reducing the total cost for sensing a specified AoI (Area of Interest) in urban district by mobile users equipped with sensors based on People-Centric Sensing (PCS). Since mobile users mobility is uncontrollable, it is difficult to guarantee sensing coverage of a specified AoI in PCS. For this challenge, the authors modeled users mobility by discrete Markov chain and formulated the problem for covering each point of a specified AoI at specified probability in a specified time. The authors proposed two novel algorithms to solve the problem and showed that the proposed algorithms accurately achieve AoI coverage of specified probability in a specified time with much smaller number of users than selecting all users in AoI.

(Chairman of SIGMBL Atsushi Takeshita)

Asaad Ahmed received his B.S. degree in Computer Science from the Faculty of Science, Alexandria University, Egypt in 1999. He received his M.S. in Computer Science from the Faculty of Science, Cairo University, Egypt in 2008. Currently, he is a Ph.D. student candidate at the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. His current research interests include distributed systems, mobile ad hoc networks and ubiquitous computing.

Keiichi Yasumoto received his B.E., M.E., and Ph.D. degrees from Osaka University, Japan, in 1991, 1993 and 1996, respectively. He joined the faculty of Shiga University in 1995. Since 2011, he has been a professor of the Graduate School of Information Science at Nara Institute of Science and Technology. His current research interests include mobile computing, ubiquitous computing, and multimedia communication. He is a member of IEICE, ACM and IEEE/CS.



Yukiko Yamauchi received her master degree and Ph.D. degree in Information Science and Technology from Osaka University in 2006 and in 2009. She was an assistant professor of the Graduate School of Information Science, Nara Institute of Science and Technology from 2009 to 2011. She is now an assistant professor of the Graduate School of ISEE, Kyushu University, Japan. Her research interests include distributed algorithms. She is a member of IEEE and IEICE.

Minoru Ito received his B.E., M.E. and Ph.D. degrees in Information and Computer Sciences from Osaka University in 1977, 1979 and 1983, respectively. He joined the faculty of Osaka University in 1979. Since 1993, he has been a professor of the Graduate School of Information Science at Nara Institute of Science and Technology. He is a member of ACM and IEEE.