Cost-minimizing Mobile Access Point Deployment in Workflow-based Mobile Sensor Networks

Haiming Jin[1], He Huang[1], Lu Su[2], and Klara Nahrstedt[1]

[1]Department of Computer Science, University of Illinois at Urbana-Champaign, IL, USA
[2]Department of Computer Science and Engineering, State University of New York at Buffalo, NY, USA
Email: {hjin8, hehuang2}@illinois.edu, lusu@buffalo.edu, klara@illinois.edu

***Abstract*—In mission-based mobile environments such as airplane maintenance, workflow-based mobile sensor networks emerge, where mobile users (MUs) with sensing devices visit sequences of mission-driven locations defined by workflows, and demand the gathering of sensory data within mission durations. To satisfy this demand in a cost-efficient manner, mobile access point (AP) deployment needs to be part of the overall solution. Therefore, we study the mobile AP deployment in workflow-based mobile sensor networks. We categorize MUs' workflows according to a *priori* knowledge of MUs' staying durations at mission locations into complete and incomplete information workflows. In both categories, we formulate the cost-minimizing mobile AP deployment problem into multiple (mixed) integer optimization problems, satisfying MUs' QoS constraints. We prove that the formulated optimization problems are NP-hard and design approximation algorithms with guaranteed approximation ratios. We demonstrate using simulations that the AP deployment cost calculated using our algorithms is 50-60% less than the stationary baseline approach and fairly close to the optimal AP deployment cost. In addition, the run times of our approximation algorithms are only 10-25% of those of the branch-and-bound algorithm used to derive the optimal AP deployment cost.**

I. Introduction

With the advances in mobile networks, we see the emergence of mobile sensor networks [1–8] that enable flexible and wide monitoring and collection of sensory data in diverse indoor and outdoor environments. Furthermore, we observe a wide spread of mobile sensor networks, where sensing devices such as smartphones, tablets or other types of sensors are carried by robots or people (MUs) that accomplish mission-critical tasks, hence move according to predefined workflows. Note that workflows define mobility patterns, i.e., sequences of mission locations that MUs are scheduled to visit. Typical examples of **workflow-based mobile sensor networks** are as follows.

- Recent years have witnessed the growing trend of utilizing handheld devices to facilitate the airline ground operations at airports (see Boeing digital airline project [9]). The handheld devices carried by airline workers, including mobile phones or tablets equipped with multiple sensors, not only provide instant access to technical and regulatory publications, but also enable real-time data capture during aircraft maintenance [10, 11]. These workers move according to their own workflows around the aircrafts and collect sensory data in form of images, text, audio or video files for further statistical analysis about conditions of the aircrafts.
- In infrastructure monitoring such as corrosion detection [12, 13] of water, petroleum or other chemical pipelines, mobile sensors carried by people or robots are usually deployed to monitor and collect data about the pipeline conditions to detect pipeline leak. These mobile sensors move along the monitored infrastructure following predefined trajectories and form workflow-based mobile sensor networks.
- In military surveillance [14–18], groups of mobile sensors carried by patrolling soldiers, unmanned vehicles or aircrafts carry out the surveillance of a particular area to detect enemy activities and intruders. These mobile sensors also move according to predefined trajectories and form similar workflow-based mobile sensor networks as the previous two examples.
- Another example of workflow-based mobile sensor networks exist in the scenario of industrial production line management [19–21]. Recent advances in robotic technology enable the deployment of mobile robots equipped with various diagnostic sensors to carry out the testing of product quality and the diagnosis of potential malfunctioning of the production line. These mobile diagnostic robots typically move along predefined trajectories during their mission time, which form workflow-based mobile sensor networks.

In other scenarios such as environmental monitoring, disaster recovery and etc., similar workflow-based mobile sensor networks also exist. The efficient and effective gathering of sensory data in the aforementioned scenarios is of great importance. One possible solution is to utilize existing commercial cellular networks to collect these data. However, in many of the aforementioned scenarios such as infrastructure monitoring and military surveillance, cellular network coverage is typically not available. Even in scenarios where cellular coverage is guaranteed, it is highly costly to purchase long-term unlimited data plans for MUs because of the large amount of data that need to be gathered. In this paper we consider a potentially much more cost efficient alternative that deploys wireless access points working as sink nodes to gather data from MUs. Thus, it is necessary to study the efficient deployment of APs for data collection from sensor nodes.

The problems of deploying stationary and mobile wireless APs for data collection in sensor networks have been widely

978-1-4799-6204-4/14 $31.00 © 2014 IEEE
DOI 10.1109/ICNP.2014.29

studied. The **stationary AP deployment problem** has been extensively studied in several previous works [22–31] to provide coverage to both mobile and stationary sensor nodes in a cost and energy efficient manner. However, stationary AP deployment, supporting mobile users, leads to AP-resource overprovisioning, hence great inefficiencies since when mobile devices move out of APs' coverage regions, those stationary APs become heavily underutilized. Therefore, the idea of placing APs on vehicles or mobile robots to make APs mobile so as to alleviate such underutilization has already been introduced by [32–36]. Among them [36] has built a prototype of mobile APs carried by mobile robots called WiFibots. However, different from the existing **mobile AP deployment** schemes that minimize data collection latency [32], maximize throughput of the sensor network [33, 35] and minimize energy consumption [34], our work takes a very different perspective on the mobile AP deployment problem. We take into consideration MUs' workflows and explore the cost-minimizing mobile AP deployment problem in workflow-based mobile sensor networks. Note that such a cost consists of the AP purchasing cost and the movement-incurred cost (e.g. gas or maintenance of an AP mobile carrier) proportional to the distance that APs travel. The major impact of this work is that it will bring significant saving in deployment cost for the entities that deploy APs to collect the sensory data. The contributions of this paper are as follows.

- Our work is the first one to study the cost-minimizing mobile AP deployment problem in workflow-based mobile sensor networks.
- We formulate the cost-minimizing mobile AP deployment problem as meaningfully solvable (mixed) integer optimization problems and prove that the formulated optimization problems are NP-hard. Further, we design efficient approximation algorithms with guaranteed approximation ratios.
- We show through our simulation results that the mobile AP deployment cost, calculated using our algorithms, is 50-60% less than that of the stationary AP baseline approach and fairly close to the optimal AP deployment cost. In addition, the run times of our approximation algorithms are only 10-25% of those of the branch-and-bound algorithm used to derive the optimal AP deployment cost.

The rest of this paper is organized as follows. Section II discusses the related work. Section III states the problem description and the system model. Section IV gives a detailed explanation of the mathematical formulation of our optimization problem. In Section V, we describe the solution techniques and analysis of our proposed algorithms. In Section VI, we elaborate on the incomplete information scenario. After describing our simulation results in Section VII, we conclude this paper in Section VIII.

II. Related Work

Sensor Deployment. Sensor deployment is an important issue in wireless sensor networks, since efficient sensor deployment is highly needed to ensure the coverage of the area of interest and connectivity among sensors. Furthermore, sensor deployment is closely related to AP deployment because the deployment of APs highly depends on the sensor topologies. [28, 37, 38] consider the deployment of stationary wireless sensors to achieve both coverage and connectivity. [39] tackles the problem of optimal sensor deployment on 3D surfaces which aims to achieve the highest overall sensing quality. [40] studies sensor deployment that minimizes the number of sensor nodes in network-structured environments. [41] studies the sensor reclamation and replacement for long-lived sensor networks. [42, 43] leverage on the mobility of mobile sensors to ensure connectivity, coverage and load balancing of the sensor network.

Stationary AP Deployment. Typically a hierarchical architecture is utilized for data collection in sensor networks, where sensory data are aggregated to APs working as sink nodes. The problem of stationary AP deployment has been extensively studied in previous literatures. [22, 25–27] explore AP placement to offer coverage to stationary devices. [22, 26, 27] explore AP deployment schemes that satisfy the connectivity requirement of the sensor nodes. [25] minimizes the deployment cost of stationary APs that cover stationary devices. [23, 24] explore the problem one step further and study AP placement to cover mobile devices. [30, 31] provide AP deployment algorithms to minimize sensor nodes' energy consumption and prolong their network life-time. [44] studies distributed placement of APs which serve as in-network processing operators and data caches in sensor networks.

Mobile AP Deployment. Due to the the AP-resource overprovisioning of stationary AP deployment, [32–35] propose to deploy mobile APs for data collection in sensor networks. [32] proposes a rendezvous based data collection approach that balances the energy consumption and data collection delay. [33, 35] provide theoretical results about the optimal AP movement that maximizes the overall lifetime. [34, 45] bundle data collection and wireless energy transfer and minimizes the energy consumption of the entire system.

Our paper is primarily different from the aforementioned related work in the following two aspects. Firstly, we take into consideration MUs' workflow information, which is readily available in the scenario of mission-driven mobile sensor networks studied in this paper. Based on such information, we are able to make better decisions about the AP movement and data collection schedules. Secondly, we seek to minimize a different objective function compared to the related work. Previous work minimizes the number of deployed APs, maximizes the network lifetime, minimizes data collection delays and so forth. However, our work minimizes the overall AP deployment cost consisting of the AP purchasing cost and the movement-incurred cost, which will bring significant cost saving for the entities that deploy the APs and collect the sensory data. Moreover, the mobile AP deployment scheme designed in this paper can be readily integrated with existing mechanisms and protocols that deal with routing [46–48] and data aggregation [49, 50] in wireless sensor networks to constitute a complete mobile sensing system.

III. System Model and Problem Description

A. Problem Description

We consider a set of P ($P \in \mathbb{Z}^+$) mobile users (MUs), $\mathcal{P} = \{1, \cdots, P\}$ carrying out particular missions in a 2D

TABLE I. NOTATIONS

$\mathcal{P}$	MU set
$\mathcal{A}$	AP set
$\mathcal{T}$	Sequence of time slots
$\mathcal{N}$	Candidate AP deployment location set
c_s	Unit cost of purchasing one stationary AP
c_{m1}	Unit cost of purchasing one mobile AP
c_{m2}	Unit cost for one AP moving one unit distance
Δt	Duration of one time slot
t_m	APs' moving duration in one time slot
t_s	APs' stationary duration in one time slot
R	APs' coverage radius
$\mathcal{B}_p$	MU p's QoS requirements along the time line
$\mathcal{M}_p^t$	MU p's possible positions in time slot t
Θ_i	APs' coverage region at candidate location i
G^t	Network topology graph in time slot t
G	AP position transition graph
Γ_i	Reachable position set for an AP at location i
v_p	Moving speed for MU p
v_{ap}	Moving speed for APs
d_{ij}	An AP's moving distance from location i to j

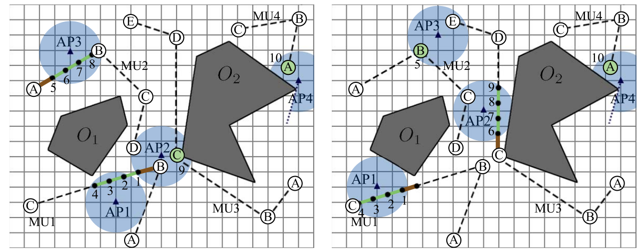
mission space with obstacles. Each of them carries a mobile sensing device to collect sensory data about the environment. The mobility of MUs follows predefined workflows, which define sequences of mission locations that MUs are scheduled to visit during their mission periods. A set of mobile APs, $\mathcal{A}$ is sent out from the beginning of the mission time line into the mission space to gather the sensory data from the MUs. The entire movement and data collection schedules of APs are computed offline before they are sent out. Our objective is to minimize the overall AP deployment cost consisting of two parts, namely the AP purchasing cost and AP moving cost. We denote the unit cost of AP purchasing as c_{m1} ($) and the unit cost of AP moving as c_{m2} ($\$/km$). The moving cost of an AP which is proportional to the distance that it travels is incurred by the gas and maintenance cost, necessary to maintain the AP's regular service. Note that the cost incurred by the communication between APs and MUs are negligible compared to the purchasing cost and moving cost. Thus, the communication cost is not taken into consideration in our objective function.

One of the major constraints that we take into consideration is MUs' QoS requirements. MUs have data transmission requirements not only at their mission locations, but also between any two consecutive mission locations during their movement. Therefore, by dividing the overall time line into a sequence of T ($T \in \mathbb{Z}^+$) time slots, $\mathcal{T} = \{1, \cdots, T\}$, we define MU p's ($\forall p \in \mathcal{P}$) QoS requirements over the time line as $\mathcal{B}_p = \{B_p^1, \cdots, B_p^T\}$, where B_p^t ($\forall t \in \mathcal{T}$) is MU p's average bandwidth requirement in time slot t.

Therefore, our overall goal is planning the number and movement of APs to minimize the mobile AP deployment cost such that MUs' QoS requirements are satisfied. Furthermore, we compare the cost of mobile AP deployment with the baseline case proposed in [24] which is a typical stationary AP deployment to provide coverage to MUs. We introduce the formal mathematical formulation in Section IV.

B. Mobility Model

In this subsection, we introduce MUs' and APs' mobility models. As mentioned earlier, the mobility of MUs follows predefined workflows. A workflow for MU p ($\forall p \in \mathcal{P}$) defines a sequence of mission locations the MU is scheduled to visit



(a) Time slot t (b) Time slot $t+1$

Fig. 1. An example of MUs' trajectories and APs' positions and coverage regions in two consecutive time slots t and $t+1$

and the corresponding durations the MU remains stationary at mission locations. Also, we assume that MU p ($\forall p \in \mathcal{P}$) moves between any two consecutive mission locations with constant speed known v_p on the straight line segment connecting the two mission locations. Then, we define the trajectory of MU p ($\forall p \in \mathcal{P}$) as Q_p consisting of the sequence of mission locations defined in the workflow and the line segments connecting every two consecutive mission locations. For example, in Fig. 1 the MU set is $\mathcal{P} = \{1, 2, 3, 4\}$. MU 3's workflow defines the mission location sequence that the MU is scheduled to visit, $\{A, B, C, D, E\}$ and the corresponding duration that MU 3 stays at every mission location, $\{T_A, T_B, T_C, T_D, T_E\}$. Hence, the trajectory for MU 3, Q_3 consists of the five mission locations and the line segments $\{\overline{AB}, \overline{BC}, \overline{CD}, \overline{DE}\}$. The aforementioned workflow is a complete information workflow since the durations that MUs stay at mission locations are known a *priori*. One way to obtain such knowledge is to carry out statistical analysis of MUs' working histories and predict with high accuracy the duration a MU takes to work at every mission location. Incomplete information workflows with predefined MUs' mission locations but unknown staying durations at these locations are defined in Section VI.

APs follow a periodically move-and-stay mobility model, i.e., in every time slot APs move in the first t_m duration with constant speed v_{ap} and remain stationary in the remaining t_s time duration ($t_m \ll t_s$). Thus, the duration of one time slot is $\Delta t = t_m + t_s$. APs can only remain stationary at a set of N ($N \in \mathbb{Z}^+$) candidate deployment locations, $\mathcal{N} = \{1, \cdots, N\}$ and we use $\mathcal{N}^t$ to denote the candidate deployment location set in time slot t, which has the same elements as $\mathcal{N}$. We divide the overall mission space into equal-sized grids and define candidate deployment locations as grid intersections which are not covered by obstacles. Furthermore, we assume that APs move along the grids which serve as APs' dedicated moving lanes between candidate deployment locations i and j ($\forall i, j \in \mathcal{N}$) in the corresponding shortest path. During the t_m duration, the maximum distance an AP travels is $v_{ap}t_m$. Then, the reachable location set for an AP placed at location i ($\forall i \in \mathcal{N}$) is $\Gamma_i = \{j | \forall j \in \mathcal{N}$ and $d_{ij} \leq v_{ap}t_m\}$ which contains the candidate deployment locations that an AP at candidate deployment location i ($\forall i \in \mathcal{N}$) is able to reach during the t_m duration within every time slot. In the example illustrated in Fig. 2 the 4 candidate deployment locations that the AP stays in the 4 times slots are candidate deployment locations 118, 94, 97 and 100. Since $v_{ap}t_m$ equals to 4 times the grid size in this example, the reachable locations for the AP at candidate deployment locations 118 and 100 are the grid intersections inside Γ_{118} and Γ_{100}.

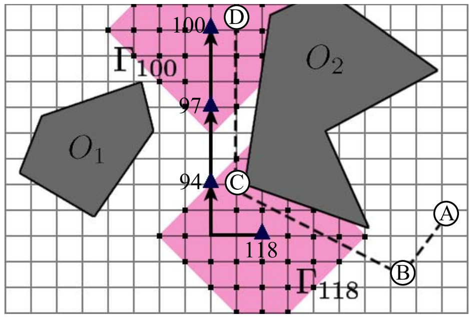


Fig. 2. An example of an AP's mobility in four consecutive time slots

We define the AP position transition graph as an undirected graph $G = (V, E)$, in which the vertex set $V = \bigcup_{t=1}^{T} \mathcal{N}^t$ and the edge set $E = \{\widehat{ij} | \forall i \in \mathcal{N}^t,\ \forall j \in \mathcal{N}^{t+1} \bigcap \Gamma_i$ and $\forall t \in \mathcal{T} \backslash \{T\}\}$. The vertex set is the combination of all the candidate location sets of all time slots and an edge $\widehat{ij}$ exists if and only if candidate deployment location j $(\forall j \in \mathcal{N}^{t+1})$ belongs to the reachable location set of location i $(\forall i \in \mathcal{N}^t)$ for any two consecutive time slots t and $t+1$. An example of an AP transition graph is shown in Fig. 3. Let's take the edges between $\mathcal{N}^1$ and $\mathcal{N}^2$. Since $\Gamma_1 = \Gamma_2 = \{1, 2\}$, edges $\widehat{11}$, $\widehat{12}$, $\widehat{22}$ and $\widehat{21}$ exist between $\mathcal{N}^1$ and $\mathcal{N}^2$.

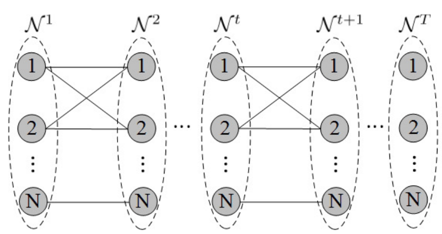


Fig. 3. An example of AP position transition graph G

C. Coverage Model

In this subsection, we first introduce the coverage model for one single AP. We assume all the communication devices use omnidirectional antennas and the transmission region in 2D open space can be regarded as circles. We define APs' coverage radius as R, which is the minimum between an AP's and a MU's transmission distance. The coverage region for an AP at location i $(\forall i \in \mathcal{N})$, Θ_i is defined as the region inside the radius-R circle centered at candidate deployment location i and within the line-of-sight region of the AP placed at location i. Note that the issue of channel allocation is beyond the scope of this paper. As long as APs with overlapping coverage regions operate on different channels, interference among APs can be minimized or avoided, which can be realized using existing channel allocation and access schemes [51–53].

Within every time slot, the trajectory of MU p $(\forall p \in \mathcal{P})$ is either a line segment (moving) or a point (stationary). To make the problem more tractable, we further divide the line segment that any MU p travels within the t_s portion of the time slot into series of shorter equal-length line segments if the MU is moving in the time slot, illustrated in Fig. 1(a). In this example, the line segment $\overline{14}$ on MU 1's trajectory is the line segment that MU 1 travels within the t_s duration in time slot t. And this line segment is further divided into line segments $\overline{12}$, $\overline{23}$ and $\overline{34}$. We construct an index set $\mathcal{M}^t$ containing indices of all end points of these line segments and those of the corresponding mission locations if the particular MUs are stationary in the time slot. Thus, MU p $(\forall p \in \mathcal{P})$ has a set of possible positions denoted as $\mathcal{M}_p^t$ in time slot t. And within every time slot t, we deploy APs to provide coverage to every position in $\mathcal{M}^t = \bigcup_{p=1}^{P} \mathcal{M}_p^t$. Based on the previous definition, in Fig. 1(a) $\mathcal{M}_1^t = \{1, 2, 3, 4\}, \mathcal{M}_2^t = \{5, 6, 7, 8\}$, $\mathcal{M}_3^t = \{9\}$ and $\mathcal{M}_4^t = \{10\}$ and in Fig. 1(b) $\mathcal{M}_1^{t+1} = \{1, 2, 3, 4\}, \mathcal{M}_2^{t+1} = \{5\}$, $\mathcal{M}_3^{t+1} = \{6, 7, 8, 9\}$ and $\mathcal{M}_4^{t+1} = \{10\}$. In time slot t, AP 1 covers {2,3,4}, AP 2 covers $\{1, 9\}$, AP 3 covers $\{5, 6, 7, 8\}$ and AP 4 covers $\{10\}$. Then in time slot $t+1$, the four APs move to new candidate deployment locations and cover $\{1, 2, 3, 4\}$, $\{5\}$, $\{6, 7, 8, 9\}$ and $\{10\}$.

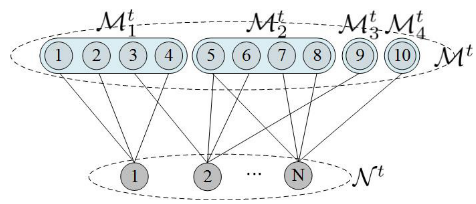


Fig. 4. One example of the network topology graph G^t in time slot t

We define the network topology graph in time slot t as a bipartite undirected graph $G^t = (V^t, E^t)$, in which $V^t = \mathcal{N}^t \bigcup \mathcal{M}^t$ is the vertex set and $E^t = \{\widehat{ij} | \forall i \in \mathcal{N}^t, \forall j \in \mathcal{M}^t$ and position j belongs to $\Theta_i\}$ is the edge set. G^t is a graph between $\mathcal{N}^t$ and $\mathcal{M}^t$ and an undirected edge $\widehat{ij}$ exists between vertex i $(\forall i \in \mathcal{N}^t)$ and vertex j $(\forall j \in \mathcal{M}^t)$ if candidate deployment location j $(\forall j \in \mathcal{M}^t)$ belongs to the coverage region of an AP staying at candidate deployment location i $(\forall i \in \mathcal{N}^t)$. An example of the bipartite representation of G^t in time slot t in Fig. 1(a) is shown in Fig. 4.

IV. Mathematical Formulation

In this section, we describe in detail our mathematical formulation of the cost-minimizing AP deployment problem for complete information workflows. We formulate the problem as a global optimization problem along the time line with two subproblems including minimum AP deployment (MinAD) and cost minimization for complete information workflows (CMinC). For every time slot t $(\forall t \in \mathcal{T})$, we carry out the MinAD to derive the minimum AP deployment while satisfying MUs' QoS constraints. Then, we use the outputs of the MinAD problem in all time slots as inputs to the CMinC problem to derive the cost-minimizing AP deployment. An illustration of the overall optimization framework is demonstrated in Fig. 5.

A. Subproblem I: Minimum AP Deployment (MinAD)

Given MUs' workflows, we can derive the network topology graph G^t at every time slot t. For every time slot t, we use G^t as an input to the MinAD problem to get the AP deployment that minimizes the number of deployed APs

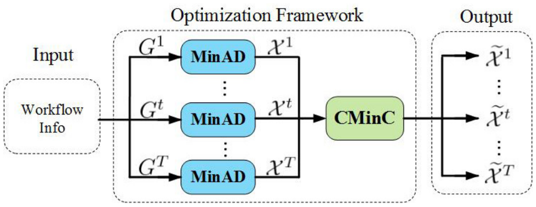


Fig. 5. Optimization framework for complete information workflows

while MUs' QoS constraints and APs' bandwidth constraints are satisfied. Therefore, in every time slot t ($\forall t \in \mathcal{T}$), the MinAD problem is formulated as follows.

Optimization Problem I (MinAD):

$$\min \sum_{i \in \mathcal{N}^t} x_i^t \tag{1}$$

$$\text{s.t.} \sum_{j \in \mathcal{M}_p^t} \sum_{i:\widehat{ij} \in E^t} y_{ij}^t \geq B_p^t t_s, \ \forall p \in \mathcal{P} \tag{2}$$

$$\sum_{j:\widehat{ij} \in E^t} y_{ij}^t \leq x_i^t B_{ap} t_s, \ \forall i \in \mathcal{N}^t \tag{3}$$

$$y_{ij}^t \in [0, +\infty), \ \forall i \in \mathcal{N}^t, \ j \in \mathcal{M}^t, \ \widehat{ij} \in E^t \tag{4}$$

$$x_i^t \in \{0, 1\}, \ \forall i \in \mathcal{N}^t \tag{5}$$

Variables: MinAD is a mixed integer programming problem, which has real positive variables $\{y_{ij}^t | \forall i \in \mathcal{N}^t, \forall j \in \mathcal{M}^t, \text{and } \widehat{ij} \in E^t\}$ and binary variables $\mathcal{X}^t = \{x_i^t | \forall i \in \mathcal{N}^t\}$. y_{ij}^t denotes the amount of data (Bytes) that a MU at candidate deployment location j ($\forall j \in \mathcal{M}^t$) transmits to the AP placed at candidate deployment location i ($\forall i \in \mathcal{N}^t$). Binary variable x_i^t ($\forall i \in \mathcal{N}^t$) represents whether it is necessary to place an AP at candidate deployment location i ($\forall i \in \mathcal{N}^t$). $x_i^t = 1$ represents that it is necessary that an AP placed at candidate deployment location i ($\forall i \in \mathcal{N}^t$) in time slot t and $x_i^t = 0$ indicates that an AP is not necessary to be placed at location i ($\forall i \in \mathcal{N}^t$) in time slot t.

Constants: t_s is APs' stationary duration within time slot t, B_p^t is the QoS (average bandwidth) requirement of MU p ($\forall p \in \mathcal{P}$) within time slot t and B_{ap} is the maximum bandwidth that an AP can support within a single time slot.

Objective function: In MinAD, the objective function to minimize is the number of necessarily deployed APs within time slot t represented by the summation of all the decision variables in $\mathcal{X}^t$.

Constraints: The MUs' QoS constraint (2) means that MU p's ($\forall p \in \mathcal{P}$) average bandwidth is no less than its minimum average bandwidth requirement within time slot t. The APs' bandwidth constraint (3) indicates that if an AP is placed at candidate deployment location i ($\forall i \in \mathcal{N}^t$), the total bandwidth requirement from all the MUs serviced by the AP should not be larger than its maximum allowable bandwidth.

B. Subproblem II: Cost Minimization for Complete Information Workflows (CMinC)

After we carry out the MinAD optimization in every time slot, we get series of solutions $\{\mathcal{X}^1, \cdots, \mathcal{X}^T\}$. Then we use $\bigcup_{t=1}^{T} \mathcal{X}^t$ as inputs to the CMinC problem to construct the MUs' QoS constraints. In fact, the solution of the CMinC problem indicates the location transition of any deployed AP between any two consecutive time slots.

Optimization Problem II (CMinC):

$$\min \left(c_{m1} \sum_{i \in \mathcal{N}^1} \sum_{j:\widehat{ij} \in E} z_{ij}^1 + c_{m2} \sum_{t=1}^{T-1} \sum_{i \in \mathcal{N}^t} \sum_{j:\widehat{ij} \in E} d_{ij} z_{ij}^t \right) \tag{6}$$

$$\text{s.t.} \sum_{j:\widehat{ij} \in E} z_{ij}^t \geq x_i^t, \ \forall i \in \mathcal{N}^t, \ t \in \mathcal{T} \setminus \{T\} \tag{7}$$

$$\sum_{i:\widehat{ij} \in E} z_{ij}^{T-1} \geq x_j^T, \ \forall j \in \mathcal{N}^T \tag{8}$$

$$\sum_{j:\widehat{ij} \in E} z_{ij}^t \leq 1, \ \forall i \in \mathcal{N}^t, \ t \in \mathcal{T} \setminus \{T\} \tag{9}$$

$$\sum_{i:\widehat{ij} \in E} z_{ij}^{t-1} = \sum_{k:\widehat{jk} \in E} z_{jk}^t, \forall j \in \mathcal{N}^t, t \in \mathcal{T} \setminus \{1, T\} \tag{10}$$

$$\sum_{i:\widehat{ij} \in E} z_{ij}^{T-1} \leq 1, \ \forall j \in \mathcal{N}^T \tag{11}$$

$$z_{ij}^t \in \{0, 1\}, \tag{12}$$

$$\forall i \in \mathcal{N}^t, \ j \in \mathcal{N}^{t+1}, \ t \in \mathcal{T} \setminus \{T\}, \ \widehat{ij} \in E$$

Variables: CMinC is an integer programming problem with binary decision variables $\mathcal{Z} = \{z_{ij}^t | \forall i \in \mathcal{N}^t, \ \forall j \in \mathcal{N}^{t+1}, \ \forall t \in \mathcal{T} \setminus \{T\}, \text{ and } \widehat{ij} \in E\}$ and $z_{ij}^t = 1$ if and only if an AP stays at candidate deployment location i in the t_s interval of time slot t and transits to candidate deployment location j in time slot $t+1$. Hence, with the AP position transition variables within $\mathcal{Z}$, we are able to calculate the values of another set of binary variables $\widetilde{\mathcal{X}}^t = \{\widetilde{x}_1^t, \cdots, \widetilde{x}_N^t\}$. The definition of $\widetilde{x}_i^t$ ($\forall i \in \mathcal{N}^t, \ \forall t \in \mathcal{T}$) is shown in equation (13), in which $\widetilde{x}_i^t = 1$ if and only if there is an AP staying at candidate deployment location i during the t_s time duration in time slot t.

$$\widetilde{x}_i^t = \begin{cases} \sum_{j:\widehat{ij} \in E} z_{ij}^t, & \text{if } t \in \mathcal{T} \setminus \{T\} \\ \sum_{j:\widehat{ij} \in E} z_{ij}^{T-1}, & \text{if } t = T \end{cases} \tag{13}$$

Note that $\mathcal{X}^t$ derived from the MinAD problem is the minimum-number AP deployment that is able to satisfy MUs' QoS constraints in time slot t. Therefore, to ensure that mobile APs reach the locations specified by $\mathcal{X}^t$ in every time slot t, the actual number of deployed mobile APs specified by $\widetilde{\mathcal{X}}^t$ should be larger than or equal to the largest number of APs specified by $\mathcal{X}^t$ along the time line.

Constants: As mentioned in Section III, c_{m1} is the cost of purchasing one AP and c_{m2} is the cost incurred if one AP travels one unit distance. The weight constant d_{ij} is the distance that an AP travels from candidate deployment location i to j. Constants $\mathcal{X} = \bigcup_{t=1}^{T} \mathcal{X}^t = \{x_i^t | \forall i \in \mathcal{N}^t, \ \forall t \in \mathcal{T}\}$, are the outputs of the MinAD problem from time slot 1 to time slot T.

Objective function: In the CMinC problem, we aim at minimizing the overall cost of mobile AP deployment, which consists of two parts, namely AP purchasing cost proportional to the number of purchased APs and AP moving cost proportional to the overall distance that all the APs travel along the time line. Since $\sum_{i\in\mathcal{N}^1}\sum_{j:\widehat{ij}\in E} z_{ij}^1$ represents the number of deployed APs and $\sum_{t=1}^{T-1}\sum_{i\in\mathcal{N}^t}\sum_{j:\widehat{ij}\in E} d_{ij}z_{ij}^t$ represents the overall distance that APs travel, by multiplying them with related unit cost c_{m1} amd c_{m2} we get the objective function (6) which is the overall cost of mobile AP deployment.

Constraints: Constraints (7) and (8) ensure that feasible solutions of CMinC satisfy MUs' QoS and APs' bandwidth constraints specified in the MinAD problem, because constraints (7) and (8) ensure that there is an AP staying at candidate deployment location i ($\forall i \in \mathcal{N}^t, \forall t \in \mathcal{T}$) if $x_i^t = 1$. Constraint (9) ensures that one AP can only transits to one candidate deployment location in one transition and constraint (10) ensures that the number of deployed APs is the same in different time slots. Constraints (10) and (11) as a whole ensure that there is at most one AP staying at one candidate deployment location in every time slot.

V. Solution and Analysis

A. Computational Complexity of MinAD and CMinC

In this subsection, we prove that the MinAD and the CMinC problems are both NP-hard in Theorems 1 and 2.

Theorem 1. *The MinAD problem is NP-hard.*

Proof: We prove that the set cover problem (SCP) which has already been proved to be NP-Complete is polynomial-time reducible to the MinAD problem. We set $B_p^t t_s = B_{ap} t_s = 1$ in constraints (2) and (3) to get a special instance of the MinAD problem. Then, by taking summation over $i : \widehat{ij} \in E^t$ on both sides of constraint (3), we have:

$$\sum_{i:\widehat{ij}\in E^t} x_i^t \geq \sum_{i:\widehat{ij}\in E^t}\sum_{j:\widehat{ij}\in E^t} y_{ij}^t \geq \sum_{j\in\mathcal{M}_p^t}\sum_{i:\widehat{ij}\in E^t} y_{ij}^t \geq 1.$$

Thus, we get an optimization problem from the MinAD problem represented by (14), (15) and (16):

$$\min \sum_{i\in\mathcal{N}} x_i^t \tag{14}$$

$$\text{s.t.} \sum_{i:\widehat{ij}\in E^t} x_i^t \geq 1 \tag{15}$$

$$x_i^t \in \{0,1\},\ \forall i \in \mathcal{N} \tag{16}$$

(14), (15) and (16) represent the integer programming formulation of a SCP, where the universe of elements is $\mathcal{U} = \{j|\forall j \in \mathcal{M}^t\}$, subsets $\mathcal{S} = \{S_1, \cdots, S_N\}$ satisfy that $S_i = \{j|\widehat{ij} \in E^t\}$ ($\forall i \in \mathcal{N}^t$) and the objective is to select the minimum number of subsets from $\mathcal{S}$ such that all elements in $\mathcal{U}$ are contained in the union of the selected subsets. Hence, the NP-complete problem SCP is polynomial-time reducible to the MinAD problem. Thus, the MinAD problem is NP-hard. ■

Theorem 2. *The CMinC problem is NP-hard when the number of time slots $T \geq 3$.*

Proof: We prove that the minimum cost 3 dimensional perfect matching problem (MC3DPM) which has already been proved to be NP-complete [54] is polynomial-time reducible to the CMinC problem. An instance of the MC3DPM problem consists of 3 sets B, C and D of the same cardinality and a cost function $f : B \times C \times D \to \mathbb{R}^+$. An assignment A, such that $A \subseteq B \times C \times D$ and each element of $B \bigcup C \bigcup D$ belongs to exactly one triple in the assignment A. Thus, the objective is to find the assignment that minimizes the cost which is defined as $\sum_{(b,c,d)\in B\times C\times D} f(b,c,d)$. In our CMinC problem, we set the number of time slots to be $T = 3$ and relax constraints (7) and (8) by setting $x_i^t = 0$ ($\forall i \in \mathcal{N}^t, \forall t \in \{1,2,3\}$). Then, the CMinC problem formulation can be represented by the following optimization problem:

$$\min \sum_{j\in\mathcal{N}^2}\sum_{i:\widehat{ij}\in E}\sum_{k:\widehat{jk}\in E} \Big((c_{m1} + c_{m2}d_{ij}) z_{ij}^1 + c_{m2}d_{jk}z_{jk}^2 \Big) \tag{17}$$

$$\text{s.t.} \sum_{j:\widehat{ij}\in E} z_{ij}^t \leq 1,\ \forall i \in \mathcal{N}^t,\ t \in \{1,2\} \tag{18}$$

$$\sum_{i:\widehat{ij}\in E} z_{ij}^1 = \sum_{k:\widehat{jk}\in E} z_{jk}^2, \forall j \in \mathcal{N}^2 \tag{19}$$

$$\sum_{i:\widehat{ij}\in E} z_{ij}^2 \leq 1,\ \forall j \in \mathcal{N}^3 \tag{20}$$

$$z_{ij}^t \in \{0,1\}, \tag{21}$$

$$\forall i \in \mathcal{N}^t,\ j \in \mathcal{N}^{t+1},\ t \in \{1,2\},\ \widehat{ij} \in E$$

(17), (18), (19), (20) and (21) represent the integer programming formulation of a MC3DPM problem defined on sets $\mathcal{N}^1$, $\mathcal{N}^2$ and $\mathcal{N}^3$ with cost function $f(i,j,k) = c_{m1} + c_{m2}(d_{ij} + d_{jk})$ ($\forall j \in \mathcal{N}^2, \widehat{ij} \in E, \widehat{jk} \in E$). Thus the NP-complete MC3DPM problem is polynomial-time reducible to a special case of the CMinC problem when $T = 3$, which is a special case of the CMinC problem when $T \geq 3$. Hence, the CMinC problem is NP-hard when $T \geq 3$. ■

B. Solution Techniques

Since the MinAD problem is a mixed-integer programming problem and the CMinC problem is a binary integer programming problem, we are unable to use standard linear programming techniques to derive solutions of these problems. Utilizing the branch-and-bound approach [55], we can get optimal solutions of these problems. However, the branch-and-bound algorithm takes an exponential number of iterations indicating excessively high computational complexity. Even though the optimization problems MinAD and CMinC are computed off-line, it still takes tremendous amount of time to obtain optimal solutions using the branch-and-bound algorithm if the input sizes are large. Thus, efficient approximation algorithms are highly necessary to derive near-optimal results with less number of iterations. We take the approach of linear relaxation based iterative rounding (LR-IR) [56] to derive approximate solutions of the aforementioned optimization problems. In such approaches, we first relax the integral constraints of a (mixed) integer optimization problem and round a subset of the fractional solution to integers. Then, we solve the residual problem iteratively until all variables take integral values. Based on this idea, we design Algorithm 1 and Algorithm 2 to tackle the MinAD and the CMinC problems.

Algorithm 1 returns approximate solutions for the MinAD

Algorithm 1: LR-IR for MinAD

```
Input: G^t, B_p, B
Output: AP placement solution {x_1^t, ..., x_N^t}
while true do
    solve the linear programming relaxation of MinAD
    with x_i^t ∈ [0,1] and get optimal solution
    {x_1^*, ..., x_N^*};
    round the largest fractional solution within
    {x_1^*, ..., x_N^*} to 1;
    if x_i^* ∈ {0,1} (∀i ∈ N^t) then
        return {x_1^*, ..., x_N^*};
    end
end
```

Algorithm 2: LR-IR for CMinC

```
Input: G, {X^1, ..., X^T}
Output: AP position transition solution Z
while true do
    solve the linear programming relaxation of CMinC
    with z_ij^t ∈ [0,1] and get optimal solution Z^*;
    round the largest fractional solution z_ij^{t*} within Z^*
    to 1 and round other variables with same t and i
    indices or the same t and j indices to 0;
    if z_ij^{t*} ∈ {0,1} (∀i ∈ N^t, j ∈ N^{t+1}, t ∈ T \ {T})
    then
        return Z^*;
    end
end
```

problem. In every iteration we round the largest fractional solution to 1 until all variables take the value 0 or 1. Furthermore, using Algorithm 2 we can derive approximate solutions for the CMinC problem. To ensure the feasibility of the approximate solutions, apart from rounding the largest fractional solution to 1 in every iteration, we also round other variables with the same t and i or t and j indices to 0.

C. Performance Analysis

For Algorithms 1 and 2, the number of iterations cannot be larger than the number of variables. Thus in terms of the number of iterations, Algorithms 1 and 2 are both $O(n)$ in which n is the number of variables. Since within each iteration, the simplex algorithm [55] is utilized to derive the optimal solution of the linear relaxation of the MinAD or CMinC problem, we cannot guarantee Algorithms 1 and 2 have polynomial time complexity in the worst case. However, our algorithms are actually efficient in practice as shown in our simulation in Section VII. Next, we introduce the analysis of the approximation ratios of Algorithms 1 and 2. Recall that B_{ap} denotes the maximum average bandwidth supportable by one AP in one time slot, B_p^t is the average QoS (bandwidth) requirement for MU p ($\forall p \in \mathcal{P}$). We define the number of edges that have $j \in \mathcal{M}^t$ as one of the end points in graph G^t as Λ_j^t.

Theorem 3. *Algorithm 1 is a factor $\frac{\gamma}{\alpha}$ approximation algorithm in which $\alpha = \frac{\sum_{p\in\mathcal{P}} B_p^t}{B_{ap}}$ and $\gamma = \max_{j\in\mathcal{M}^t} \Lambda_j^t$.*

Proof: We prove that in every iteration there is at least one fractional variable $x_i^t \geq \frac{\alpha}{N}$. By taking summation on both sides of constraint (3) over $i : \widehat{ij} \in E^t$ and on both sides of constraint (2) over $p \in \mathcal{P}$, we have:

$$\begin{aligned} \sum_{i:\widehat{ij}\in E^t} x_i^t &\geq \frac{1}{B_{ap} t_s} \sum_{i:\widehat{ij}\in E^t} \sum_{j:\widehat{ij}\in E^t} y_{ij}^t \\ &\geq \frac{1}{B_{ap} t_s} \sum_{p\in\mathcal{P}} \sum_{j\in\mathcal{M}_p^t} \sum_{j:\widehat{ij}\in E^t} y_{ij}^t \\ &\geq \frac{\sum_{p\in\mathcal{P}} B_p^t}{B_{ap}} \end{aligned} \tag{22}$$

In our formulation, we have $\sum_{p\in\mathcal{P}} B_p^t \geq B_{ap}$. Therefore, when the number of iterations $m \geq \frac{\sum_{p\in\mathcal{P}} B_p^t}{B_{ap}}$, the algorithm terminates and we get a set of approximate solutions for the MinAD problem. Thus, we consider m^{th} round iteration with $1 \leq m < \frac{\sum_{p\in\mathcal{P}} B_p^t}{B_{ap}}$. Then in the previous $m-1$ iterations, $m-1$ variables have been rounded to 1. Therefore, from constraint (22) we have at least one x_i^t such that

$$\begin{aligned} x_i^t &\geq \frac{\frac{\sum_{p\in\mathcal{P}} B_p^t}{B_{ap}} - (m-1)}{\Lambda_j^t - (m-1)} \geq \frac{\sum_{p\in\mathcal{P}} B_p^t}{\Lambda_j^t B_{ap}} \\ &\geq \frac{\sum_{p\in\mathcal{P}} B_p^t}{B_{ap} \max_{j\in\mathcal{M}^t} \Lambda_j^t} \end{aligned}$$

Since $\alpha = \frac{\sum_{p\in\mathcal{P}} B_p^t}{B_{ap}}, \gamma = \max_{j\in\mathcal{M}^t} \Lambda_j^t$, then at every iteration at least one fractional variable $x_i^t \geq \frac{\alpha}{\gamma}$, which implies that Algorithm 1 is a factor $\frac{\gamma}{\alpha}$ approximation algorithm. ■

Next, we introduce the conclusion of the approximation ratio of Algorithm 2. Recall that Γ_i is the reachable position set for an AP staying at candidate location i ($\forall i \in \mathcal{N}$) with $|\Gamma_i|$ being its corresponding cardinality. Then, we have the following Theorem 4 about the approximation ratio of Algorithm 2.

Theorem 4. *Algorithm 2 is a factor $\max_{i\in\mathcal{N}} |\Gamma_i|$ approximation algorithm.*

Proof: We prove that in every iteration there are at least one fractional variable $z_{ij}^t \geq \frac{1}{\max_{i\in\mathcal{N}} |\Gamma_i|}$ in the basic feasible solution of the linear relaxation of the CMinC problem.

1). In the first iteration, we suppose that a particular $x_i^t = 1$ ($i \in \mathcal{N}^t, t \in \mathcal{T} \setminus \{T\}$). Then from constraint (7), we have at least one $z_{ij}^t \geq \frac{1}{|\Gamma_i|}$ ($\forall j : \widehat{ij} \in E$). Otherwise, $\sum_{j:\widehat{ij}\in E} z_{ij}^t < \sum_{j:\widehat{ij}} \frac{1}{|\Gamma_i|} = 1$. For the same reason, if there exists at least one $x_j^T = 1 (\forall i \in \mathcal{N}^T)$, from constraint (8), we have at least one $z_{ij}^{T-1} \geq \frac{1}{|\Gamma_j|}$.

2). In the following iterations, since we have rounded at least one variable $z_{ij}^t = 1$ in previous iterations, either constraint (10) or constraints (7) and (8) will result in at least one fractional variable $z_{ij}^t \geq \frac{1}{|\Gamma_i|}$ ($\forall t \in \mathcal{T} \setminus \{T\}$) or $z_{ij}^{T-1} \geq \frac{1}{|\Gamma_j|}$.

From $|\Gamma_i| \leq \max_{i\in\mathcal{N}} |\Gamma_i|$, we arrive at the conclusion that in every iteration there exists at least one fractional $z_{ij}^t \geq \frac{1}{\max_{i\in\mathcal{N}} |\Gamma_i|}$. This implies that Algorithm 2 is a factor $\max_{i\in\mathcal{N}} |\Gamma_i|$ approximation algorithm. ■

VI. Incomplete Information Workflows

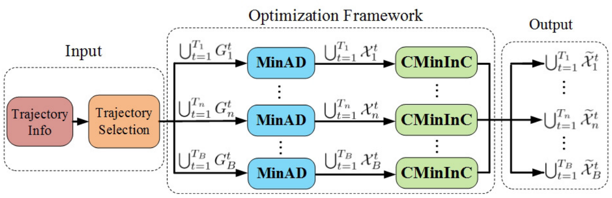


Fig. 6. Optimization framework for incomplete information workflows

In this section, we extend our analysis to the scenario with incomplete information workflows where the durations that MUs stay at mission locations are not a *priori* known information. The challenge to deal with this scenario is that the exact mapping between MUs' positions and their corresponding time instances is not known beforehand. Hence, it is impossible to carry out a global optimization along the time line to minimize the AP deployment cost. However, in this scenario we still have information including MUs' trajectories and MUs' moving speeds between consecutive mission locations. To address this problem, we divide any MU's trajectory into *dedicated-AP trajectory segments* and *shared-AP trajectory segments*. On the former the MU does not share APs with others, whereas on the latter the MU shares APs with others. We develop a hybrid approach to deploy stationary APs using existing algorithm [24] on shared-AP trajectory segments and deploy mobile APs on dedicated-AP trajectory segments. On dedicated-AP trajectory segments, it is possible that due to the limited AP speeds and the large QoS requirements of MUs, multiple APs might be needed to provide coverage for one MU. Therefore, we formulate the problem as a combination of optimization problems defined on every dedicated-AP trajectory segment. The overview of the optimization framework in this case is illustrated in Fig. 6.

A. Trajectory Selection

The first stage of the cascaded optimization framework is also the MinAD problem formulated in Section IV-A. However, the inputs to the MinAD problem in this scenario are the series of network topology graphs constructed on dedicated-AP trajectory segments. In this subsection, we introduce the trajectory selection process.

On dedicate-trajectory segments, MUs do not share APs with others. Therefore, we realize the trajectory selection process by drawing circles with radius R (APs' coverage radius) on every point of every trajectory in $\{Q_1, \cdots, Q_P\}$. Then we select the trajectory segments that do not have overlapping circles with other trajectories. One example of trajectory selection is demonstrated in Fig. 7. We draw circles with radius R centered on the four MUs' trajectories $\{Q_1, Q_2, Q_3, Q_4\}$. For MU 4, since it does not have overlapping circles with other trajectories, MU 4's overall trajectory is its dedicate-AP trajectory segment. For MU 3, $\widetilde{Q}_1$ and $\widetilde{Q}_2$ are its dedicated-AP trajectory segments because they do not share APs with other MUs' trajectories.

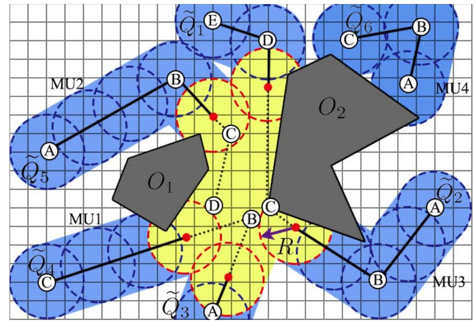


Fig. 7. An example of trajectory selection

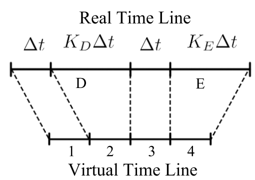


Fig. 8. An example of virtual time line construction

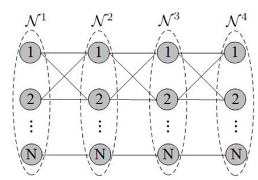


Fig. 9. An example of AP transition graph

B. Network Topology Graph and AP Position Transition Graph

In this subsection, we introduce the construction of the network topology graphs and AP position transition graph for every dedicated-AP trajectory segment. We denote the index set of dedicated-AP segments as $\mathcal{S}$. Since for incomplete information workflows, the duration that MU p $(\forall p \in \mathcal{P})$ stays in mission locations are not known a *priori*, it is impossible to get the real time line for selected trajectory segments beforehand. However, we construct a virtual time line corresponding to every dedicated-AP trajectory segment $\widetilde{Q}_n$ $(\forall n \in \mathcal{S})$ by setting the duration that one MU stays at one mission location to be one time slot. One example of such virtual time line construction is illustrated in Fig. 8. On the trajectory segment $\widetilde{Q}_1$, MU 3 moves for 1 time slot, stays at mission location D for K_D $(K_D \in \mathbb{Z}^+)$ time slots, moves for 1 time slot and then stays at mission location E for K_E $(K_E \in \mathbb{Z}^+)$ time slots. Although K_D and K_E are not known a *priori*, we set K_D and K_E to be 1 and obtain a virtual time line consisting of 4 time slots shown in Fig. 8. The constructed virtual time line will yield the same result in terms of the number of deployed mobile APs and the overall distance the APs travel.

Then w.r.t. trajectory segment $\widetilde{Q}_n$ $(\forall n \in \mathcal{S})$, we construct a set of T_n network topology graph $\{G_n^1, \cdots, G_n^{T_n}\}$ on the virtual time line using the method introduced in Section III-C. Furthermore, an AP position transition graph $G_n = (V_n, E_n)$ $(\forall n \in \mathcal{S})$ can also be constructed w.r.t. the virtual time line of trajectory segment $\widetilde{Q}_n$ $(\forall n \in \mathcal{S})$ consisting of vertices $\bigcup_{t=1}^{T_n} \mathcal{N}^t$ and the edges between vertices in $\bigcup_{t=1}^{T_n} \mathcal{N}^t$ in the AP position transition graph G defined Section III-B. In Fig. 9, we show an example of the AP position transition graph corresponding to the virtual time line constructed in Fig. 8. Since the virtual time line has 4 time slots, the AP position transition graph has vertices $\bigcup_{t=1}^{4} \mathcal{N}^t$ and edges between $\mathcal{N}^t$ and $\mathcal{N}^{t+1}$ $(t \in \{1, 2, 3\})$.

C. Mathematical Formulation, Solution Techniques and Analysis

In this subsection, we first introduce the mathematical formulation of the cost-minimizing mobile AP deployment problem for incomplete information workflows. Shown in Fig. 6, the first stage of the optimization is also the MinAD formulation defined in Section IV-A. In this scenario, for trajectory segment $\widetilde{Q}_n$ $(\forall n \in \mathcal{S})$ MinAD is calculated w.r.t. all time slots along the corresponding virtual time line and derive the solution sets $\{\mathcal{X}_n^1, \cdots, \mathcal{X}_n^{T_n}\}$. The solution sets together with the AP position transition graph G_n $(\forall n \in \mathcal{S})$ are used as inputs to the cost minimization problem for incomplete information workflows (CMinInc) defined on dedicated-AP trajectory segment Q_n $(\forall n \in \mathcal{S})$.

Optimization Problem III (CMinInC):

$$\min\ c_{m1} \sum_{i \in \mathcal{N}^1} \sum_{j:\widehat{ij} \in E_l} z_{ij}^{n1} + c_{m2} \sum_{t=1}^{T_n-1} \sum_{i \in \mathcal{N}^t} \sum_{j:\widehat{ij} \in E_l} d_{ij} z_{ij}^{nt} \tag{23}$$

$$\text{s.t.} \sum_{j:\widehat{ij} \in E_n} z_{ij}^{nt} \geq x_i^{nt},\ \forall i \in \mathcal{N}^t,\ t \in \mathcal{T}_n \setminus \{T_n\} \tag{24}$$

$$\sum_{i:\widehat{ij} \in E_n} z_{ij}^{n(T_n-1)} \geq x_j^{nT_n},\ \forall j \in \mathcal{N}^{T_n} \tag{25}$$

$$\sum_{j:\widehat{ij} \in E_n} z_{ij}^{nt} \leq 1,\ \forall i \in \mathcal{N}^t,\ t \in \mathcal{T}_n \setminus \{T_n\} \tag{26}$$

$$\sum_{i:\widehat{ij} \in E_n} z_{ij}^{n(t-1)} = \sum_{k:\widehat{jk} \in E_n} z_{jk}^{nt}, \tag{27}$$
$$\forall j \in \mathcal{N}^t,\ t \in \mathcal{T}_n \setminus \{1, T_n\}$$

$$\sum_{i:\widehat{ij} \in E_n} z_{ij}^{n(T_n-1)} \leq 1,\ \forall j \in \mathcal{N}^{T_n} \tag{28}$$

$$z_{ij}^{nt} \in \{0,1\}, \tag{29}$$
$$\forall i \in \mathcal{N}^t,\ j \in \mathcal{N}^{t+1},\ t \in \mathcal{T}_n \setminus \{T_n\},\ \widehat{ij} \in E_n$$

Since CMinInC and CMinC share some constants and constraints, we will only elaborate on the ones that are different from those in CMinC. Note that constants that have the same notations in CMinC and CMinInC have the same meanings.

Variables: CMinInC is a binary integer programming with decision variables $\mathcal{Z} = \{z_{ij}^{nt} | \forall i \in \mathcal{N}^t,\ \forall j \in \mathcal{N}^t,\ \forall t \in \mathcal{T}_n \setminus \{T_n\},\ \widehat{ij} \in E_n\}$, in which $z_{ij}^{nt} = 1$ if and only if an AP stays at position i $(\forall i \in \mathcal{N}^t)$ in time slot t and transits to position $j \in \mathcal{N}^{t+1}$ in time slot $t+1$.

Constants: Constants x_i^{nt} $(\forall i \in \mathcal{N}^t,\ \forall t \in \mathcal{T}_n)$ in constraints (24) and (25) are the outputs of the MinAD problem which uses network topology graphs G_n^t $(\forall n \in \mathcal{S}, \forall t \in \mathcal{T}_n)$ as inputs.

Objective function: In CMinInC, we aim at minimizing cost of mobile AP deployment on the trajectory segment $\widetilde{Q}_n$ $(\forall n \in \mathcal{S})$ represented by objective function (23).

Constraints: Similar to CMinC, constraints (24) and (25) ensure the minimum AP deployment is satisfied along the virtual time line, constraint (26) ensures that one AP can only transits to one candidate location in one single transition, constraint (27) ensures that the number of deployed APs for the coverage of trajectory segment $\widetilde{Q}_n$ $(\forall n \in \mathcal{S})$ is the same in different time slots and constraints (27) and (28) ensure that there is at most one AP staying at one location in every time slot for the coverage of $\widetilde{Q}_n$ $(\forall n \in \mathcal{S})$.

Corollary 5. *The CMinInC problem is NP-hard when the number of time slots of the virtual time line corresponding to trajectory segment Q_n $(\forall n \in \mathcal{S})$ $T_n \geq 3$.*

Proof: The CMinInc problem is equivalent to CMinC with T_n, E_n and z_{ij}^{nt} corresponds to T, E and z_{ij}^t. Based on Theorem 2 the CMinC problem is NP-hard when $T_n \geq 3$. ■

Since the CMinInc problem is equivalent to CMinC with only minor modifications to several constants, the CMinInc problem can also be solved using Algorithm 2.

VII. Simulation Results

A. Experimental Settings

TABLE II. Scenarios for complete information workflows

Scenario	Mission Space	AP Candidate Positions	Time Line	MUs
I	500m×500m	34	3000s	2∼16
II	1000m×1000m	114	3000s	8∼30

For complete information workflows, we first carry out our simulation in the two scenarios described in Table II. In both scenarios we set the grid size to be 100m and APs can only remain stationary at grid intersections. Thus, considering obstacles, scenario I and scenario II has 34 and 114 AP candidate positions respectively. In scenario I, we compare the AP deployment cost and the number of deployed APs calculated using our approximation algorithms, the branch-and-bound (optimal) algorithm and the baseline stationary AP deployment algorithm [24]. Furthermore, we also compare the run times between our approximation algorithms and the branch-and-bound algorithm in scenario I. To demonstrate that our approximation algorithms are highly scalable in terms of the size of inputs, we carry out simulation in scenario II in which the branch-and-bound algorithm is unable to return a solution in finite time. The run times of the algorithms are measured on a machine with a $2.0GHz \sim 2.5GHz$ CPU and 8GB memory.

TABLE III. Parameters for complete information workflows

c_s	c_{m1}	c_{m2}	Δt	t_m	R	v_p	v_{ap}
6000$	10000$	$\frac{1}{11.035}$$/km	60s	10s	100m	1.5m/s	10m/s

The value of other parameters used during the simulation are demonstrated in Table III, from which we have $c_s < c_{m1}$. We set the duration of a single time slot to be 60s and the moving duration for an AP to be $t_m = 10s$. Thus APs move for at most 100m within every time slot and then remain stationary for the remaining $t_s = 50s$. We set the number of time slots along the time line to be $T = 50$ and plan the AP deployment for 52560 periodic recurring time lines. For an average vehicle, its average cost per kilometer is $\frac{1}{11.035}$$.

For incomplete information workflows, we carry out our simulation in a 1000m×1000m mission space with 114 candidate AP locations. We set the number of MUs to be 8 and

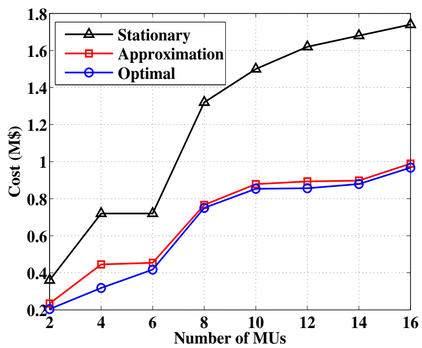


Fig. 10. Cost comparison for scenario I

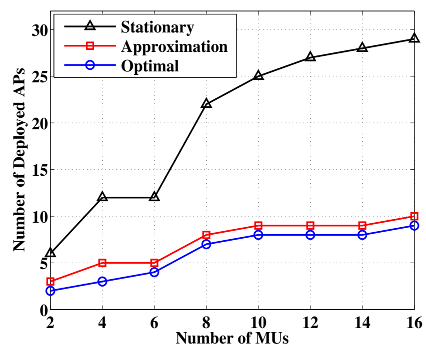


Fig. 11. Deployed AP number comparison for scenario I

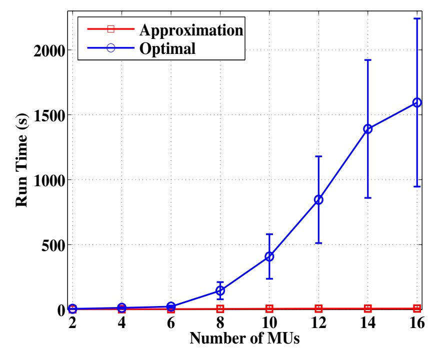


Fig. 12. Algorithm 1 run time comparison for scenario I

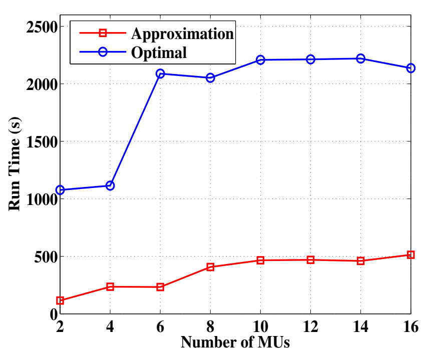


Fig. 13. Algorithm 2 run time comparison for scenario I

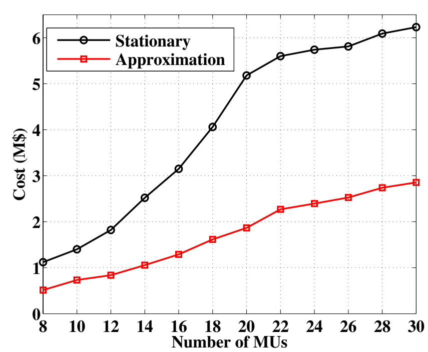


Fig. 14. Cost comparison for scenario II

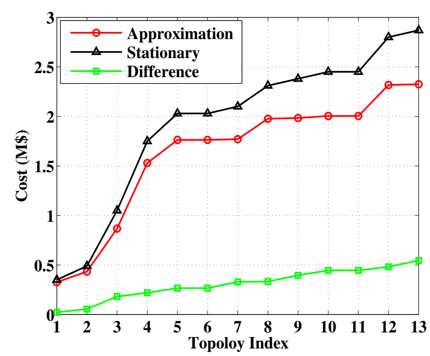


Fig. 15. Cost comparison for incomplete information scenario

study the benefit in the saving of the AP deployment cost of our hybrid approach compared to the baseline approach. The parameter that varies among groups is the percentage of selected trajectory segments β defined as $\beta = \frac{\sum_{m \in \mathcal{S}} T_m}{\sum_{p=1}^{8} T'_p}$ in which T'_p $(\forall p \in \{1, \cdots, 8\})$ is the number of time slots of the virtual time line constructed on trajectory Q_p. In our experiment, we study a group of 13 different topologies with the same $\sum_{p=1}^{8} T'_p$ values but different β values shown in Table IV.

TABLE IV. PERCENTAGE OF SELECTED TRAJECTORY SEGMENTS

Topology Index	1	2	3	4	5	6	7
β (%)	12.5	19.6	33.9	60.7	69.6	71.4	73.2
Topology Index	8	9	10	11	12	13	
β (%)	80.4	83.9	85.7	87.5	98.2	100.0	

B. Simulation Results and Discussions

In this subsection, we introduce the simulation results and discussions. In Figs. 10 and 11, we compare the AP deployment cost and the number of deployed APs among the stationary case, our approximation algorithms and the brand-and-bound (optimal) algorithm. Fig. 10 illustrates that when we introduce mobility to APs, the cost of AP deployment can be remarkably reduced compared to the baseline case in which APs are stationary. And our approximation algorithms yield AP deployment cost fairly close to the optimal cost derived using the brand-and-bound algorithm. Fig. 11 shows by exploiting APs' mobility we can reduce the number of deployed APs compared to the the baseline stationary AP scenario. Such reduction in the number of APs is a major reason of the decrease in the AP deployment cost when the overall moving cost of all APs is much less than the AP purchasing cost which is true in most practical scenarios.

In terms of the computational performance of our algorithms, we measure and compare the run time of our approximate algorithms and the branch-and-bound algorithm on the same machine. Fig. 12 illustrates the run time of Algorithm 1 used to solve the MinAD problem. we calculate the average value and the standard deviations of the run time of Algorithm 1 among the 50 time slots along the time line with different number of MUs in scenario I. In Fig. 12, the run time of the optimal algorithm increases exponentially as the number of MUs increases. This is because the number of variables increases linearly w.r.t. the number of MUs and the branch-and-bound algorithm has to use exponential number of iterations to reach the optimal solution. However, Algorithm 1 is able to converge in much less time than the optimal algorithm and increases only linearly with small slope w.r.t. the increasing of the number of MUs. In Fig. 13, the run time of the branch-and-bound algorithm does not increase exponentially w.r.t. the number of MUs because the number of MUs does not affect the number of variables and constraints in the CMinC problem. However, the Algorithm 2 still converges in much less time than that of the optimal branch-and-bound algorithm.

In Fig. 14, we demonstrate that our approximation algorithms still work in scenario II with a much larger input size than scenario I, in which the optimal branch-and-bound algorithm is not able to return a solution in finite time. In Fig. 15, we plot the cost of mobile AP deployment, the cost of stationary AP deployment and their difference w.r.t. different percentages of selected trajectory segments β. In our hybrid AP deployment approach, the difference of the AP deployment cost on selected trajectory segments is the difference between the cost of the baseline approach and that of our hybrid AP deployment approach. From Fig. 15, we can arrive at the conclusion that the difference between baseline cost and our hybrid approach increases w.r.t. trected trajectory segments.

VIII. Conclusions

We study the cost-minimizing mobile AP deployment problem in workflow-based mobile sensor networks, which we formulate as multiple (mixed) integer optimization problems. We prove that the formulated optimization problems are NP-hard and design efficient approximation algorithms with guaranteed approximation ratios. We demonstrate through our simulation results that the mobile AP deployment cost, calculated using our algorithms, is 50-60% less than that of the stationary AP baseline approach and fairly close to the optimal AP deployment cost. In addition, the run times of our approximation algorithms are only 10-25% of those of the branch-and-bound algorithm used to derive the optimal AP deployment cost.

Acknowledgment

This material is based upon work supported in part by Boeing and the Ralph and Catherine Fisher grant.

References

[1] S. Zhang, J. Cao, C. Li-Jun, and D. Chen, "Accurate and energy-efficient range-free localization for mobile sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 6, pp. 897–910, 2010.

[2] K. Ma, Y. Zhang, and W. Trappe, "Managing the mobility of a mobile sensor network using network dynamics," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 19, no. 1, pp. 106–120, 2008.

[3] G. Wang, M. J. Irwin, H. Fu, P. Berman, W. Zhang, and T. L. Porta, "Optimizing sensor movement planning for energy efficiency," *ACM Trans. Sen. Netw.*, vol. 7, no. 4, pp. 33:1–33:17, Feb. 2011.

[4] B. Liu, O. Dousse, P. Nain, and D. Towsley, "Dynamic coverage of mobile sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 2, pp. 301–311, Feb 2013.

[5] N. Bartolini, T. Calamoneri, T. F. L. Porta, and S. Silvestri, "Autonomous deployment of heterogeneous mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 10, no. 6, pp. 753–766, 2011.

[6] W. Liu, L. Sun, Y. Liu, H. Zhu, Z. Zhong, and T. He, "Exploiting ephemeral link correlation for mobile wireless networks," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '12. New York, NY, USA: ACM, 2012, pp. 239–252.

[7] D. Zhang, T. He, F. Ye, R. Ganti, and H. Lei, "Eqs: Neighbor discovery and rendezvous maintenance with extended quorum system for mobile sensing applications," in *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*, June 2012, pp. 72–81.

[8] M. Li, W. Cheng, K. Liu, Y. He, X. Li, and X. Liao, "Sweep coverage with mobile sensors," *Mobile Computing, IEEE Transactions on*, vol. 10, no. 11, pp. 1534–1545, Nov 2011.

[9] P. Noren, "The digital airline by boeing," 2011. [Online]. Available: http://aerosociety.com/Assets/Docs/Events/673/The_Digital_Airline.pdf

[10] P. Niklaus, "Using handheld devices for real-time data capture reduces cycle time," in *Specialists in Air Transport Communications and IT Solutions (SITA)*, 2010.

[11] M. Lampe, M. Strassner, and E. Fleisch, "A ubiquitous computing environment for aircraft maintenance," in *Proceedings of the 2004 ACM Symposium on Applied Computing*, ser. SAC '04. ACM, 2004, pp. 1586–1592.

[12] N. Sridhar, C. S. Brossia, A. Sabata, and G. Tormoen, "Development and application of mobile sensor network to monitor corrosion in pipelines," *CORROSION 2006*, 2006.

[13] J.-H. Kim, "Sensor-based autonomous pipeline monitoring robotic system," Ph.D. dissertation, Louisiana State University, 2011.

[14] S. Kumar, T. H. Lai, and A. Arora, "Barrier coverage with wireless sensors," in *MobiHoc*, 2005.

[15] A. Chen, S. Kumar, and T. H. Lai, "Designing localized algorithms for barrier coverage," in *MobiCom*, 2007.

[16] A. Chen, T. H. Lai, and D. Xuan, "Measuring and guaranteeing quality of barrier-coverage in wireless sensor networks," in *MobiHoc*, 2008.

[17] Y. Wang and G. Cao, "Barrier coverage in camera sensor networks," in *MobiHoc*, 2011.

[18] X. Gong, J. Zhang, D. Cochran, and K. Xing, "Barrier coverage in bistatic radar sensor networks: Cassini oval sensing and optimal placement," in *Proceedings of the Fourteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '13. New York, NY, USA: ACM, 2013, pp. 49–58.

[19] A. Cesetti, C. P. Scotti, G. Di Buo, and S. Longhi, "A service oriented architecture supporting an autonomous mobile robot for industrial applications," in *Control Automation (MED), 2010 18th Mediterranean Conference on*, June 2010, pp. 604–609.

[20] C. Cristalli, G. Angione, L. Lattanzi, B. Graf, F. Weisshardt, and G. Arbeiter, "Mo.di.bot - mobile diagnostic robot." in *ECAI*, ser. Frontiers in Artificial Intelligence and Applications, vol. 242. IOS Press, 2012, pp. 1007–1008.

[21] R. Luo, T. Hsieh, K. Su, and C. Tsai, "An intelligent remote maintenance and diagnostic system on mobile robot," in *IECON 02 [Industrial Electronics Society, IEEE 2002 28th Annual Conference of the]*, vol. 4, Nov 2002, pp. 2675–2680 vol.4.

[22] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang, "Two-tiered constrained relay node placement in wireless sensor networks: Computational complexity and efficient approximations," *Mobile Computing, IEEE Transactions on*, vol. 11, pp. 1399–1411, 2012.

[23] T. Wang, W. Jia, G. Xing, and M. Li, "Exploiting statistical mobility models for efficient wi-fi deployment," *Vehicular Technology, IEEE Transactions on*, vol. 62, no. 1, pp. 360–373, 2013.

[24] Y. Huang, Y. Gao, and K. Nahrstedt, "Relay placement for reliable base station connectivity in polymorphous networks," in *SECON*, 2010.

[25] J. Zhang, X. Jia, Z. Zheng, and Y. Zhou, "Minimizing cost of placement of multi-radio and multi-power-level access points with rate adaptation in indoor environment," *Wireless Communications, IEEE Transactions on*, vol. 10, pp. 2186–2195, 2011.

[26] Z. Zheng, P. Sinha, and S. Kumar, "Sparse wifi deployment for vehicular internet access with bounded interconnection gap," *Networking, IEEE/ACM Transactions on*, vol. 20, no. 3, pp. 956–969, 2012.

[27] S. Misra, S. D. Hong, G. Xue, and J. Tang, "Constrained relay node placement in wireless sensor networks: Formulation and approximations," *Networking, IEEE/ACM Transactions on*,

vol. 18, no. 2, pp. 434–447, 2010.
[28] X. Cheng, D.-Z. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *Wirel. Netw.*, vol. 14, no. 3, pp. 347–355, Jun. 2008.
[29] L. Liao, W. Chen, C. Zhang, L. Zhang, D. Xuan, and W. Jia, "Two birds with one stone: Wireless access point deployment for both coverage and localization," *Vehicular Technology, IEEE Transactions on*, vol. 60, no. 5, pp. 2239–2252, 2011.
[30] K. Xu, H. Hassanein, G. Takahara, and Q. Wang, "Relay node deployment strategies in heterogeneous wireless sensor networks," *Mobile Computing, IEEE Transactions on*, pp. 145–159, 2010.
[31] D. Li, W. Zhong, Q. Shi, H. Zhao, and B. Huang, "Secondary access points placement in cognitive radio networks: A spatial game model with power discrimination," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 6, pp. 2729–2739, 2012.
[32] G. Xing, T. Wang, W. Jia, and M. Li, "Rendezvous design algorithms for wireless sensor networks with a mobile base station," in *MobiHoc*, 2008.
[33] Y. Shi and Y. Hou, "Theoretical results on base station movement problem for sensor network," in *INFOCOM*, 2008.
[34] L. Xie, Y. Shi, Y. Hou, W. Lou, H. D. Sherali, and S. F. Midkiff, "Bundling mobile base station and wireless energy transfer: Modeling and optimization," in *INFOCOM*, 2013.
[35] W. Liu, K. Lu, J. Wang, L. Huang, and D. Wu, "On the throughput capacity of wireless sensor networks with mobile relays," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 4, pp. 1801–1809, 2012.
[36] M. Erdelj, T. Razafindralambo, and D. Simplot-Ryl, "Covering points of interest with mobile sensors," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 1, pp. 32–43, 2013.
[37] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," in *MobiHoc*, 2006.
[38] X. Bai, D. Xuan, Z. Yun, T. H. Lai, and W. Jia, "Complete optimal deployment patterns for full-coverage and k-connectivity ($k \leq 6$) wireless sensor networks," in *MobiHoc*, 2008.
[39] M. Jin, G. Rong, H. Wu, L. Shuai, and X. Guo, "Optimal surface deployment problem in wireless sensor networks," in *INFOCOM*, 2012.
[40] S. Xiong, L. Yu, H. Shen, C. Wang, and W. Lu, "Efficient algorithms for sensor deployment and routing in sensor networks for network-structured environment monitoring," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 1008–1016.
[41] B. Tong, G. Wang, W. Zhang, and C. Wang, "Node reclamation and replacement for long-lived sensor networks," in *SECON*, 2009.
[42] S. Chellappan, X. Bai, B. Ma, D. Xuan, and C. Xu, "Mobility limited flip-based sensor networks deployment," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 18, no. 2, pp. 199–211, 2007.
[43] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," *Mobile Computing, IEEE Transactions on*, vol. 5, no. 6, pp. 640–652, 2006.
[44] L. Ying, Z. Liu, D. Towsley, and C. Xia, "Distributed operator placement and data caching in large-scale sensor networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, April 2008.
[45] L. Xie, Y. Shi, Y. T. Hou, W. Lou, and H. D. Sherali, "On traveling path and related problems for a mobile station in a rechargeable sensor network," in *Proceedings of the Fourteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '13. New York, NY, USA: ACM, 2013, pp. 109–118.
[46] L. Su, B. Ding, Y. Yang, T. Abdelzaher, G. Cao, and J. Hou, "ocast: Optimal multicast routing protocol for wireless sensor networks," in *Network Protocols, 2009. ICNP 2009. 17th IEEE International Conference on*, Oct 2009, pp. 151–160.
[47] L. Su, C. Liu, H. Song, and G. Cao, "Routing in intermittently connected sensor networks," in *Network Protocols, 2008. ICNP 2008. IEEE International Conference on*, Oct 2008, pp. 278–287.
[48] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 56–67.
[49] L. Su, Y. Gao, Y. Yang, and G. Cao, "Towards optimal rate allocation for data aggregation in wireless sensor networks," in *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '11. New York, NY, USA: ACM, 2011, pp. 19:1–19:11.
[50] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tag: A tiny aggregation service for ad-hoc sensor networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 131–146, Dec. 2002.
[51] S. Ahmad, M. Liu, T. Javidi, Q. Zhao, and B. Krishnamachari, "Optimality of myopic sensing in multichannel opportunistic access," *Information Theory, IEEE Transactions on*, vol. 55, no. 9, pp. 4040–4050, 2009.
[52] Y. Ding, K. Pongaliur, and L. Xiao, "Channel allocation and routing in hybrid multichannel multiradio wireless mesh networks," *Mobile Computing, IEEE Transactions on*, vol. 12, no. 2, pp. 206–218, 2013.
[53] Z. Li, Y. Peng, D. Qiao, and W. Zhang, "Joint aggregation and mac design to prolong sensor network lifetime," in *ICNP*, 2013.
[54] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering," in *INFOCOM*, 2004.
[55] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*, 1st ed. Athena Scientific, 1997.
[56] J. Cheriyan, S. Vempala;, and A. Vetta, "Network design via iterative rounding of setpair relaxations," *Combinatorica*, vol. 26, no. 3, pp. 255–275, Jun. 2006.