

# Preferential Field Coverage Through Detour-Based Mobility Coordination<sup>†</sup>

HANY MORCOS  
hmorcos@bu.edu

AZER BESTAVROS  
best@bu.edu

IBRAHIM MATTA  
matta@bu.edu

Computer Science Department  
Boston University

**Abstract**—Controlling the mobility of mobile nodes (*e.g.*, robots) to monitor a given field is a well-studied problem in sensor networks. In this setup, absolute control over the nodes’ mobility is assumed. In this paper, we address a more general setting in which mobility of each node is externally constrained by a *schedule* consisting of a list of locations that the node must visit at particular times. Typically, such schedules exhibit some level of *slack*, which could be leveraged to achieve a specific coverage distribution of a field. Such a distribution defines the relative importance of different field locations. We define the Constrained Mobility Coordination problem for Preferential Coverage (CMC-PC) as follows: given a field with a desired monitoring distribution, and a number of nodes  $n$ , each with its own schedule, we need to coordinate the mobility of the nodes in order to achieve the following two goals: 1) satisfy the schedules of all nodes, and 2) attain the required coverage of the given field. We show that the CMC-PC problem is NP-complete (by reduction from the Hamiltonian Cycle problem). Then we propose TFM, a distributed heuristic to achieve field coverage that is as close as possible to the required coverage distribution. We verify the premise of TFM using extensive simulations, as well as taxi logs from a major metropolitan area. We compare TFM to the random mobility strategy—the latter provides a lower bound on performance. Our results show that TFM is very successful in matching the required field coverage distribution, and that it provides, at least, two-fold query success ratio for queries that follow the target coverage distribution of the field.

## I. INTRODUCTION

Controlling mobility of a number of objects (*e.g.*, robots) in order to cover a given field is a well-studied problem in the literature. In this model, node mobility could be used to (i) circumvent low density of nodes, (ii) navigate to hard-to-reach areas (due to natural barriers) in order to achieve uniform coverage of the field, and/or (iii) react to some change in the environment (*e.g.*, forest fire), or address preferential coverage based on changing demands. In such a model, it is usually assumed that the mobile nodes are under the control of a single authority that decides the mobility pattern of each mobile node.

In this paper, we consider a model of *autonomous* mobile users (nodes / sensors). These autonomous mobile users are interested in monitoring a given field according to some target distribution—the distribution defines the percentage of time different field locations should be covered by, at least, one node (sensor). The field monitoring distribution stems from the inherent interest of users to query the state of different field locations. We also assume that mobility of each user is

coarsely directed by an *external* schedule. A node’s schedule defines a list of locations and a corresponding list of times (waypoints), such that, for a node to satisfy its schedule, it has to be present at the specified locations at the indicated times. An important attribute of such a schedule is how tight/relax are the consecutive journeys between waypoints. That is, if a schedule allows much more time, than the needed minimum, for a node to reach each waypoint, then it would be a relaxed schedule with plenty of *slack*, otherwise, it would be a tight schedule. The problem is then, *how to coordinate the mobility of nodes and manage their slacks so as to achieve the requested monitoring distribution.*

To see why this is the case, consider a situation where a user moving between two points A and B may have multiple choices of paths of almost equal expected quality (*e.g.*, in terms of traveled distance or time). Taking any of the alternative paths leads to monitoring different field locations. Such a scenario is particularly true for paths between locations in a dense urban setting. As an illustration, consider Figure 5, which shows paths followed by cabs on the streets of the San Francisco Bay area. The grid structure of the paths taken (underscoring the underlying city blocks in SF) demonstrates the existence of multiple routes of *indistinguishable lengths*, to travel between arbitrary points A and B on the grid. In such a case, it is perceivable that one might think that all nodes would satisfy their own schedules in one of the following manners: (1) Nodes would prefer paths leading to the monitoring of high-demand spots in the field, or (2) nodes would take random routes in each journey between each two consecutive waypoints in the schedule.

In the first scenario, if all users end up monitoring the same (highest-demand) field locations, the rest of the field would be left unmonitored, resulting in missing many of the users queries. On the other hand, if nodes take random paths, as we will show in the evaluation section (Section V), this will lead to poor coverage of the field, since the “importance” of each field location (indicated by the desired/target monitoring distribution) will be ignored when making random mobility decisions. This accentuates the importance of *coordinating* mobility of users, while ensuring that all schedules are satisfied.

Our contributions can be summarized as follows:

- We apply the above mobility model of autonomous nodes and its features (*i.e.*, slack) to the problem of distributed field coverage. We coin the problem of Constrained Mobility Coordination for Preferential field Coverage

<sup>†</sup> Supported in part by NSF awards #0720604, #0735974, #0820138, and #0952145.

(CMC-PC). We show that this problem is NP-complete (Section II), and argue that none of the existing research efforts is adequate to solve the problem (Section III).

- We develop TFM, the first mobility coordination strategy that aims to achieve a given distribution of field coverage (Section IV). Under TFM, in steady state, nodes are in a dynamic (*i.e.*, mobile) state. This salient characteristic of TFM enables it to achieve the required coverage distribution of a spatio-temporal field with a low-density network.
- Using extensive simulations, we compare TFM to the random mobility strategy — the latter provides a lower bound on performance. Our results indicate the significant performance gain attained by using TFM over random mobility (Section V-A). More importantly, TFM is shown to closely meet the target field coverage distribution.
- Furthermore, we perform a trace-driven evaluation of TFM and random mobility. We use cab traces from cabs in the San Francisco area. Results of the trace-driven evaluation underscore the effectiveness of TFM in practical settings (Section V-B).

## II. PROBLEM DEFINITION

In this section, we define the Constrained Mobility Coordination problem for Preferential Coverage (CMC-PC), then show that it is NP-complete.

**Definition 1: (Nodes):**  $N$  autonomously mobile nodes move in the target field. Mobility of each node is externally constrained by a schedule (Definition 2). The prime goal of these nodes is to satisfy their own schedules. While doing so, they also try to cooperatively cover the target field according to the required coverage distribution (Definition 4).

**Definition 2: (Schedule  $L$ ):** A schedule of node  $n_i$  is a list  $L(n_i)$  of tuples of the form  $u_{ij} = (\tau_{ij}, l_{ij})$ , where  $1 \leq j \leq |L(n_i)|$ . To satisfy a schedule entry  $u_{ij}$ , node  $n_i$  has to be at location  $l_{ij}$  at time  $\tau_{ij}$ . For  $n_i$  to satisfy its schedule, it has to satisfy  $u_{ij}$  for all  $1 \leq j \leq |L(n_i)|$ .

**Definition 3: (Field  $G$ ):** The target field is represented as a graph  $G = (V, E)$ , such that each vertex  $v \in V$  represents a field location, and each edge  $e \in E$  connects two vertices representing two field locations that could be directly reached from each other.

**Definition 4: (Coverage Distribution  $D$ ):** Coverage of a given field is defined by a target coverage distribution  $D$ , such that  $D(v)$  is the relative importance of field location  $v \in V$ . The coverage distribution  $D$  represents the preferential interest in covering different locations in the field, and is application-specific. Practically,  $D$  could be interpreted in a number of ways. For example, we could require that more important field locations be covered more *frequently* than less important ones. Another interpretation, is to require that more important locations be covered for *longer periods* compared to less important ones. In this paper, we adopt the latter interpretation. Specifically, we interpret  $D(v)$  as the required percentage of time, during which, field location  $v \in V$  should be covered, by at least one node. We also note that, at any time, a field location is either covered or not. Hence, covering a given location with only one node is exactly equivalent to covering it with more than one node.

**Definition 5: (Communication Range  $r$ ):** Any two nodes can communicate with each other only if the distance between them is less than or equal to a (given) fixed communication range  $r$ .

**Definition 6: (Speed of Motion  $\eta_i$ ):** The maximum speed of motion of a node  $n_i$  is  $\eta_i$ . Without loss of generality, we assume that  $\eta_i = \eta_{\max}$ ,  $1 \leq i \leq N$ .

**Definition 7: (The CMC-PC Problem  $P$ ):** The Constrained Mobility Coordination problem for Preferential Coverage CMC-PC is defined by the tuple  $P(G, D, N, L)$ , such that  $G$  is a given field to cover with a target distribution  $D$  using a set of  $N$  mobile nodes, each with its own schedule  $L(n_i)$ . In order to solve a given instance of the CMC-PC problem, we need to coordinate mobility of the  $N$  nodes in order to achieve two goals: 1) satisfy schedules  $L$  of all nodes, and 2) cover each field location,  $v \in V$ , the percentage of time indicated by the target distribution  $D(v)$ . Clearly, any feasible solution to the CMC-PC problem must satisfy the maximum speed requirement, *i.e.*, no node is allowed to move with a speed higher than  $\eta_{\max}$ .

Theorem 1 states that CMC-PC is NP-complete by reduction from the Hamiltonian Cycle Problem. A Hamiltonian Cycle is a cycle in an undirected graph which visits each vertex exactly once and then returns to the starting vertex. Determining (and finding) whether a Hamiltonian cycle exists in a given graph is NP-complete, so is our CMC-PC problem.

**Theorem 1:** The CMC-PC problem is NP-complete.

Due to lack of space, we refer the reader to [14] for the proof of this theorem.

## III. RELATED WORK

The problem we study here is mainly related to sensor deployment and redeployment, field coverage, and motion planning.

**Field Coverage With Static Nodes:** Multiple research efforts [13], [7], [5], [16] concentrated on calculating the coverage level attained by a *static* network. For example, Dhillon *et al.* [5] formulate the coverage problem as an optimization problem where they attempt to optimize placement of sensors in the field to maximize attained average coverage of the field.

**Field Coverage With Mobile Nodes:** Another group of research efforts concentrate on the effect of mobility on network coverage [17], [19], [6], [10], [15], [18]. Most efforts in this group start from a sub-optimal deployment of nodes in the field (*e.g.*, random), calculate an “optimal” deployment, and then move each node to its newly calculated location. These efforts differ, basically, in the way they calculate the new locations of sensors. So, the network starts from a static configuration, then nodes move once to reach another optimal *static* configuration.

Another group of research efforts concentrated on the attained dynamic coverage of a mobile network. For example, in [11], the authors study the efficacy of a mobile network in field surveillance. They gauge the ability of the network to detect a static and a mobile intruder. A common factor in these efforts is that the steady state of the network is a dynamic one, unlike previous research efforts. Our work resembles these efforts in this regard. However, our work addresses the

general problem of *constrained* mobility coordination of nodes in order to achieve some *given* monitoring distribution.

**Robotics Motion Planning:** Motion planning has been studied in the robotics field [9], [8]. Coupling robotics and sensor networks concepts has also been studied [4], [2], [12]. These efforts study problems of sensors carried by robots, and the required modifications in robots mobility planning in order to support tasks of sensor networks. Our work is also different from these efforts in that, we assume that sensors are embedded into platforms that are *autonomously* mobile by nature, and whose mobility has a limited degree of freedom (*i.e.*, slack) that could be planned to optimize performance of the embedded sensor network.

#### IV. PREFERENTIAL FIELD COVERAGE MOBILITY STRATEGY

In Section II, we showed that CMC-PC is NP-complete. In this section we propose the Targeted Field Monitoring (TFM) mobility strategy, a distributed heuristic to solve the problem. To execute this algorithm, each mobile node  $n_i$  needs to know its own schedule  $L(n_i)$ , and the target coverage distribution function  $D$ . This algorithm does not assume existence of a centralized decision-making facility nor knowledge about schedules of other nodes.

TFM uses another algorithm to assign a utility value to each field location, based on the coverage distribution  $D$ . Then, at each time unit, TFM plans node mobility by selecting the field location to be visited at the next time unit such that it maximizes the utility of visited field locations, satisfying the node's schedule at all times.

Specifically, let us denote the current location of node  $n_i$  as  $v_c$ . Let us also denote the immediately accessible field locations from the current location, *i.e.*, the set of neighboring locations, by  $N(v_c)$ .  $N(v_c)$  is the set of field locations  $v_f \in V$  such that there exists a direct edge between  $v_c$  and  $v_f$ . Formally,

$$N(v_c) = \{v_f \in V | v_f \neq v_c \text{ AND } e = (v_c, v_f) \in E\} \quad (1)$$

Figure 1 (left) shows field location  $v_c$ , and its neighboring locations  $N(v_c)$ . In order to plan its mobility, node  $n_i$  needs to decide at each time unit, which of the neighboring field locations,  $N(v_c)$ , it will move to next. To that end,  $n_i$  executes the Two-phase Utility Assignment (2UA) algorithm to assign a utility value to each location  $v_f \in N(v_c)$ . Node  $n_i$ , then, decides on the next step greedily to maximize the utility of visited locations. We will now describe the two phases of the 2UA algorithm.

**Phase One of the 2UA Algorithm:** During this phase,  $n_i$  assigns an *initial* utility value  $U_i(v)$  to each field location  $v \in V$ . Utility of each field location is a function of the popularity of this location (defined by  $D(v)$ ), the specific node carrying out the calculations  $n_i$ , and the time of performing this calculation.

More specifically,  $n_i$  keeps a local "view" of the field representing the last time each field location was last visited by *any* of the mobile sensors. Let us denote the local view of node  $n_i$  as  $C_i$ , where  $C_i(v)$  is the last time field location

$v$  was last visited by *any* node, according to node  $n_i$  (Figure 1 center). Node  $n_i$  updates its local view of the field at two occasions: 1) Whenever it visits a new field location, it updates the last time this location was visited to the current time, and 2) whenever it encounters another node  $n_j$ , the two nodes exchange their views of the field. The result of this exchange is that, each node keeps the most recent version of the two views.

Using its current view of the field  $C_i$ , node  $n_i$  calculates the utility of field location  $v$  as

$$U_i(v) = D(v) \times (t_c - C_i(v)) \quad (2)$$

where  $t_c$  is the actual time of performing the utility calculation (*i.e.*, the current time). Notice that Equation 2 is a linear function of the popularity of the location,  $D(v)$ , and the length of the interval since location  $v$  was last visited ( $t_c - C_i(v)$ ). Equation 2 is just an example for utility calculations, which could take any different form (*e.g.*, exponential in the location popularity). Notice also that this equation is related to our interpretation of  $D(v)$  as the required percentage of time during which field location  $v$  should be covered.

**Phase Two of the 2UA Algorithm:** In this phase, node  $n_i$  calculates a *coarse* utility value,  $\hat{U}_i(v_f)$ , for each of the directly neighboring locations,  $v_f \in N(v_c)$ . The coarse utility of  $v_f$  is calculated as the sum of utilities of field locations comprising the highest-utility path of length  $h$  that starts from  $v_f$ . More specifically, for each  $v_f \in N(v_c)$ , we find *all* paths of length  $h$  that start from  $v_f$ , called Potential Future Paths (PFP's). The utility of each PFP is the sum of initial utilities (calculated in Phase 1 of 2UA) of field locations comprising this path. The coarse utility of  $v_f$  is the highest utility of all such PFP's starting at  $v_f$ . Formally,

$$\hat{U}_i(v_f) = U^p(P_{best}(v_f, h)) = \sum_{v_x \in P_{best}(v_f, h)} U_i(v_x) \quad (3)$$

where function  $U^p(P)$  calculates the sum of utilities of locations in PFP  $P$ , and  $P_{best}(v_f, h)$  is the PFP with the highest utility and is defined as,

$$P_{best}(v_f, h) = \{P'(v_f, h) : P'(v_f, h) \in \mathcal{P}(v_f, h) \text{ AND } (U^p(P'(v_f, h)) \geq U^p(P(v_f, h)), \forall P(v_f, h) \in \mathcal{P}(v_f, h))\} \quad (4)$$

where  $\mathcal{P}(v_f, h)$  is the set of all PFP's of length  $h$  that start from location  $v_f$ , and  $P(v_f, h) \in \mathcal{P}(v_f, h)$  is defined as a series of  $h$  connected field locations, the first of which is  $v_f$ . Figure 1 (right) shows the current location  $v_c = (3,3)$ , its directly neighboring locations  $N(v_c) = \{(2,3), (4,3), (3,2), (3,4)\}$ , and the range of PFP's from the neighboring locations of length  $h = 1$ . For example, for  $v_f = (2,3)$ ,  $P(v_f, 1) = \{((2,3), (1,3)), ((2,3), (2,2)), ((2,3), (2,4))\}$ .

**Scope of PFP's:** In the model we described, PFP's constitute some form of lookahead in order to optimize performance, and  $h$  is a tunable parameter that defines the exact amount of lookahead to perform. Hence, we need to answer the

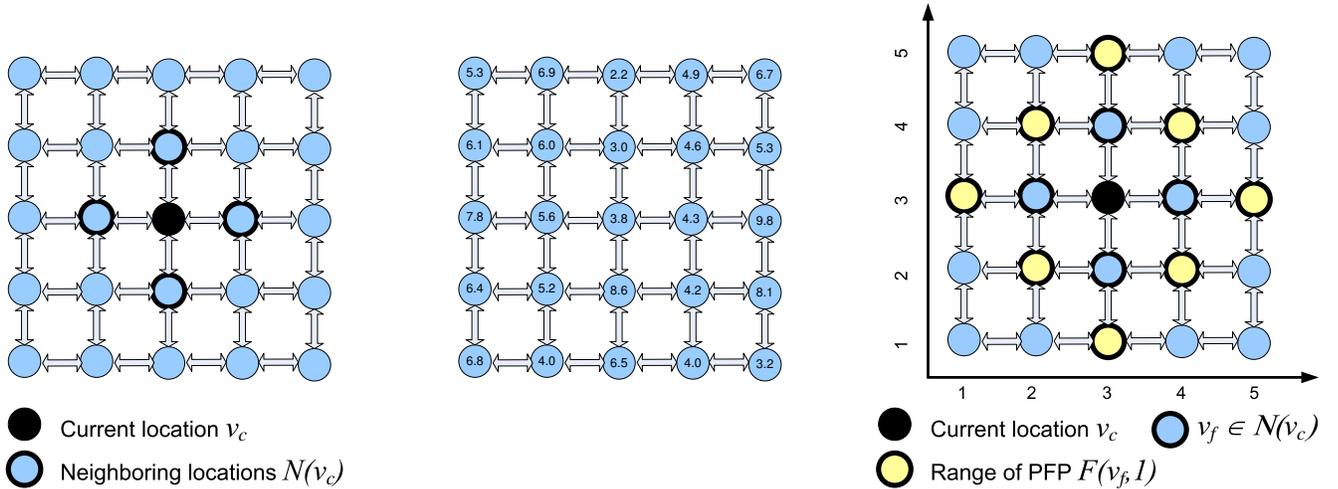


Fig. 1. Location  $v_c$ , and its neighboring locations (left),  $C_i$ : local field view of node  $n_i$  (center), and range of PFP  $P(v_f, h)$ , where  $h = 1$ .

TABLE I  
SCHEDULE OF  $n_1$ .

Time	Location
1	(2,4)
5	(4,2)

question: “What is the optimum range of PFP’s to consider? Consequently, what is the best value for  $h$ ?”. It is natural to think that the higher the value of  $h$ , the longer the range of considered PFP’s, the longer it takes to plan mobility, and the more optimal the mobility decisions are. There is, however, a dynamic restriction on the value of  $h$  to be used at any neighboring location  $v_f$ . Theorem 2 states this restriction, then we give an example to illustrate it.

*Theorem 2:* In a field coverage system with a single node, while determining the PFP’s  $P(v_f, h)$  of a location  $v_f$ , only field locations that could be actually reached by the node (due to scheduling constraints) should be included in PFP’s. Hence, the optimum value of the locale radius  $h$  is the amount of slack  $k$  available to the node at the current time. Using any other values of  $h$  could lead to sub-optimal decisions.

Due to lack of space, we refer the reader to [14] for the proof of this theorem, where we show that sub-optimal decisions result from: 1) not including all reachable field locations, and 2) including field locations that are not reachable.

The following example illustrates the idea of Theorem 2. It illustrates the case that not including all reachable field locations in deciding the PFP’s of each  $v_f$  results in sub-optimal coverage of the field.

Table I gives the schedule of node  $n_1$ , while Figure 2 (left) shows  $v_c$  the current location of node  $n_1$ , along with field locations directly accessible from  $v_c$ ,  $N(v_c) = \{(2, 3), (3, 4)\}$ . It also shows the initial *utility* of visiting each location, the result of phase one of the 2UA algorithm. Notice that locations (1, 4) and (2, 5) are not members of the set  $N(v_c)$ , because the schedule of  $n_1$  does not allow enough slack to visit any of these locations. Figure 2 (left) also highlights the set of field locations that could be reached given the schedule of  $n_1$ .

Node  $n_1$  needs to assign a coarse utility value  $\hat{U}_i(v_f)$  to each of the neighboring locations. Figure 2 (center) shows the PFP with the highest utility for each  $v_f$  such that  $h = 3$ , which matches the time needed to get to the destination.  $P_{\text{best}}((3, 4), 3) = ((3, 4), (4, 4), (4, 3), (4, 2))$ , while  $P_{\text{best}}((2, 3), 3) = ((2, 3), (3, 3), (4, 3), (4, 2))$ . In this case, both PFP’s visit the high-utility location (4, 3).  $U^P(P_{\text{best}}((3, 4), 3)) = 4.3$ , while  $U^P(P_{\text{best}}((2, 3), 3)) = 4.1$ . Based on these calculations,  $n_i$  moves to (3, 4) as its next step.

Figure 2 (right) shows the same paths when  $h = 1$ . Notice that in this case, not all reachable field locations are included in the range of PFP’s.  $P_{\text{best}}((3, 4), 1) = ((3, 4), (4, 4))$ , while  $P_{\text{best}}((2, 3), 1) = ((2, 3), (2, 2))$ , and  $U^P(P_{\text{best}}((3, 4), 1)) = 0.9$ , while  $U^P(P_{\text{best}}((2, 3), 1)) = 1.0$ . Based on these calculations,  $n_i$  moves to (2, 3) as its next step, which is clearly a sub-optimal decision.

Theorem 2 addresses the case of a field coverage system comprised of a single node. In the case of multiple nodes, mobility planning decisions made by one node in the past, could be “invalidated” in the future due to mobility of other nodes in the system. For example, a field location  $v_x$  of high initial utility according to node  $n_i$  in the past, could be visited by another node  $n_j$  causing its current utility to drop when actually visited by  $n_i$ . We argue, however, that Theorem 2 still holds in systems of sparse deployment, and we give evidence to this insight in our trace-driven evaluation in Section V-B.

## V. PERFORMANCE EVALUATION

In order to evaluate the efficacy of TFM in achieving a specific coverage distribution of a given field, we developed a mobility simulator. Our simulator models the mobility of nodes by keeping track of the location of each node at each time unit. It also models the exchange of local views between nodes upon an encounter. Since our goal is to evaluate the synthesized mobility of our detour-based techniques, we make simplifying assumptions about the communication model as we assume that nodes within a certain communication range could successfully exchange data. We assume that the size of

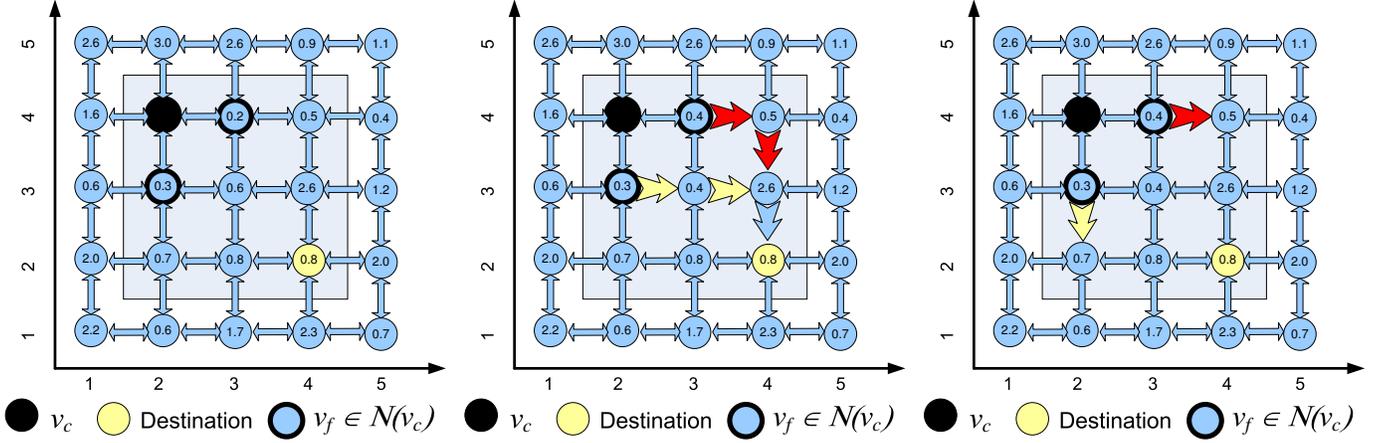


Fig. 2. Location  $v_c$ , its neighboring locations  $N(v_c)$ , the destination waypoint, and the utility of each location (left),  $P_{\text{best}(v_f,3)}$  (center) and  $P_{\text{best}(v_f,1)}$  (right).

exchanged messages is small with respect to the bandwidth in a single contact between two nodes. We also, willingly, overlook storage limitations. We do this motivated by current advances in storage technology that make memory devices of tens of gigabytes available off-the-shelf.

**Performance Metrics:** The performance metrics we use are the Kullback-Leibler (KL) distance, and the query success ratio (QSR). The KL distance is a measure of distance between distributions [3]. Having a true distribution  $P$ , and an approximated one  $Q$ , the KL-distance between  $P$  and  $Q$ ,  $KL(P||Q)$ , is calculated as:

$$KL(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \quad (5)$$

Mobility of nodes over the field induces a distribution  $Q$  of the length of periods during which each location is covered. We measure the distance between the required coverage distribution,  $D$ , and the induced distribution  $Q$ ,  $KL(D||Q)$ . Lower values of the KL distance indicate that the induced distribution is close to the required distribution  $D$ , which is the prime requirement in field coverage.

Then, we assume that nodes have unlimited storage in which they keep collected samples from the field. A node keeps a sample from each field location it visits. A sample is assumed to have a time-to-live (TTL), during which this sample is considered to be fresh, *i.e.*, an accurate representation of the target phenomenon at the field location where it was collected. Only fresh samples are kept in the local storage, while expired ones are evicted. Nodes are independently queried about the state of the field <sup>1</sup>. A query is defined by a tuple  $q = (v_q, e)$ .  $v_q$  is the query target, the location in which the inquirer is interested, while  $e$  is a measure of tolerable imprecision in the answer. The specific locations of query targets follow some spatial distribution over the field. The answer to a query  $q = (v_q, e)$  is a sample collected at

<sup>1</sup> We can think of this as if the user/owner hosting the mobile node is interested in the state of some location in the field, so she uses the local device, to which the sensor is attached, to submit queries and receive answers back from the distributed system.

location  $v$ , such that the distance between  $v$  and  $v_q$  is less than  $e$  (*i.e.*,  $|v - v_q| \leq e$ ). We can think of a query as a circle whose center is the query target  $v_q$ , and whose radius is the imprecision  $e$ . In this case, the query answer is a sample collected from within this circle.

In order to answer any query, a queried node searches its local storage to find a sample that could be used as an answer to the query. If found, then the query is counted as a success. Otherwise, the queried node forwards the query to its *direct* neighbors only. If one of these neighbors has an answer to the query, this neighbor sends the answer back to the queried node, and the query is counted as a success. If neither the queried node nor any of its neighbors has an answer to the query, it is counted as a missed query. In order to assess the efficacy of each mobility model in achieving the required probability distribution, we matched the distribution of query targets to that of the required coverage distribution  $D$  (*i.e.*, field locations that are required to be monitored for longer periods of time have higher probability of being query targets). We define the query success ratio (QSR) as the ratio between the number of successfully answered queries to the total number of queries.

The point of the two performance metrics is to gauge the degree to which each mobility strategy can match the required coverage distribution  $D$ . If a mobility strategy could closely match this distribution, this should be manifested in achieving a small KL distance, and high query success ratio.

**Competing Strategies:** We compare TFM to the random mobility strategy (RND), in which nodes move randomly between every two consecutive waypoints provided that the schedule is satisfied. In the trace-driven evaluation, we compare TFM to Wait-at-Destination (WAD), a variation of RND. Under WAD, nodes move to the destination waypoint using the shortest path, where they wait spending all the available slack, if any. Clearly, both RND and WAD represent lower bounds on performance, since they do not actively attempt to coordinate nodes' mobility to improve coverage of the field.

In the synthetic evaluations, we evaluated the performance of TFM and RND with respect to query handling in two differ-

ent setups, distributed and centralized. In the distributed setup, only nodes within communication range  $r$  could communicate so as to cooperate in query handling. This version is denoted as “DST” in the following graphs. In the centralized setup (denoted as “CTR” in the following graphs), query handling is done in a centralized facility. In this case, we simulate the case where all collected samples are forwarded to a central processing facility, and all queries are directed to this facility.

### A. Evaluation Using Synthetic Workloads

**Schedule Generation:** Every node starts at time  $= t_{\text{current}}$  (initially,  $t_{\text{current}} = 1$ ) at a random location in the field  $loc_1$ . The entry  $(t_{\text{current}}, loc_1)$  is added to the schedule. Then we randomly select another location  $loc_2$  in the field such that the minimum time to move from  $loc_1$  to  $loc_2$  is  $t$ . For location  $loc_2$ , we assign time  $t_s$

$$t_s = t_{\text{current}} + t + (\kappa \times \rho) \quad (6)$$

where  $\kappa$  is the *maximum slack* we allow in any journey, and  $\rho$  is a uniform random variable such that  $\rho \in [0, 1]$ . The entry  $(t_s, loc_2)$  is appended to the schedule. We repeat this process until the end of the simulation time is reached.

**Baseline Parameters:** We simulated a field of  $10 \times 10$  where nodes can communicate only when they are at the same field location. Each simulation runs for 100 seconds. In the following graphs, each point is the average of 20 simulation runs, with the 95% confidence intervals shown as well. The required field coverage distribution is assumed to be a symmetric bivariate normal distribution centered in the center of the field, with variance  $= 4 \times I$ , where  $I$  is the identity matrix of size  $2 \times 2$ . In the following experiments, unless otherwise stated, the default value of the maximum PFP’s  $h_{\text{max}} = 1$ , number of nodes  $N = 30$ , and query precision  $e = 1$ .

For lack of space, we show representative results and we refer the reader to [14] for more details.

**Effect of Number of Nodes:** Figure 3 shows the effect of increasing the number of nodes on the performance of the system in handling queries. TFM achieves between 2-fold to 3-fold improvement in QSR over RND. Increasing the TTL of collected samples or the maximum slack  $\kappa$  improves performance of both protocols, however, TFM is always superior to RND because nodes under RND do not coordinate usage of their slack to optimize coverage of the field. It is also interesting to notice that the distributed version of TFM achieves close performance to that of the centralized one. Increasing either TTL or maximum slack  $\kappa$  diminishes the gap between the two versions. This confirms the premise of TFM in a distributed practical setup. While, there is a noticeable gap between QSR of the distributed and centralized versions of RND, as expected of a naïve approach.

**Effect of Maximum Slack  $\kappa$ :** Figure 4 shows the KL distance of RND and TFM as a function of the maximum slack  $\kappa$  of the schedule, in a system of 5 nodes (Figure 4 left), and 15 nodes (Figure 4 right). Increasing  $\kappa$  allows TFM to match the required distribution resulting in a smaller value of the KL distance. This is true for systems of both high and low densities. Increasing the number of nodes has a more pronounced positive effect on the KL distance of RND compared to TFM.

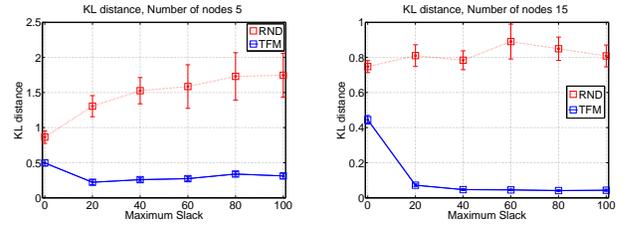


Fig. 4. KL distance of TFM and RND as a function of the maximum slack for different number of nodes.

**Effect of Partially Following Detours:** The goal of this experiment is to measure the effectiveness of TFM when only a given percentage of the nodes follow detour hints provided by TFM. This scenario is motivated by the observation that some nodes may not be willing to participate in the field coverage application, and opt to spend their slack in a different way. For lack of space we only summarize these results [14]. Obviously, as the percentage of nodes following TFM hints increases the resulting KL distance decreases. When 60% of the nodes follow TFM hints, the resulting KL distance decreases by up to 67% ( $\kappa = 100$ ).

In [14] we also show that, similar to the KL distance, increasing the percentage of nodes that follow TFM hints causes a linear increase in the QSR of the system. Interestingly, the slope of the linear increase is a function of the slack, and the “height” (*i.e.*, success level) is a function of the TTL of samples. With enough schedule slack, a density of TFM-compliant nodes as low as 0.3 could achieve QSR approaching 100%.

### B. Trace-Driven Evaluation

Following our motivating application, and to present even more realistic evaluation of the protocol we propose, we used taxi traces [1] for cabs in the San Francisco area as input to our models. The goal is to show that, with little coordination between cabs, they could function as an effective distributed field coverage system.

**Methodology:** For each cab, the traces show location updates of the cab. Each update is composed of latitude and longitude of the cab location, the time of the location update, along with the cab status: metered (hailed) or not (not hailed). We gathered more than a full day’s worth of data for more than 450 cabs. In the traces we collected, some cabs have as many as 400 location updates, while others have as few as 5 updates. We used all location updates for all cabs to construct a “map” of the San Francisco area. We represented the map as an undirected graph  $G = (V, E)$ .  $V$  is the set of all legitimate locations any cab can be in at any time, where a location is defined by its latitude and longitude coordinates. In the data we collected, the total number of locations is 40399, and the number of unique locations,  $|V| = 39,103$  locations. To determine the relation between different locations (*i.e.*, the edges,  $E$ ), we used a threshold-based neighborhood algorithm with a threshold value  $r_{th}$ . This means that, for any two locations  $a$  and  $b$ , such that the distance between them is  $Dist(a, b)$ , if  $Dist(a, b) \leq r_{th}$ , then we add an edge between  $a$  and  $b$  whose cost  $= Dist(a, b)$ . We used  $r_{th} = 200$  meters ( $\approx 0.12$  miles = 656 feet). This value of  $r_{th}$  partitioned the

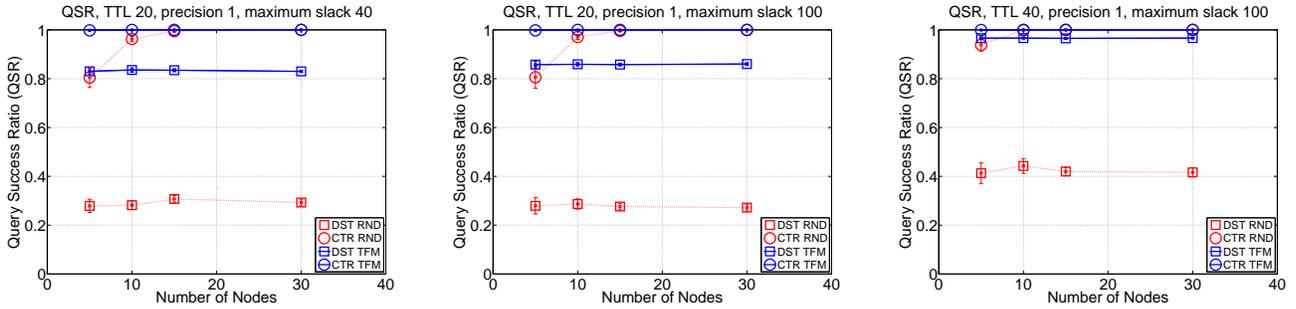


Fig. 3. Query success ratio of TFM and RND as a function of the number of nodes in the system for different slack values.

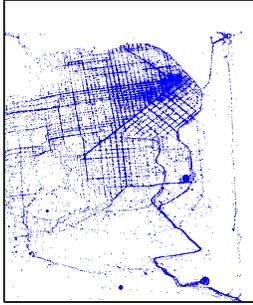


Fig. 5. Paths followed by cabs in the SF Bay area, from traces used to evaluate mobility coordination approaches.

unique field locations into different partitions, with the largest partition consisting of 36,368 unique locations. We used this partition as a representative of the map. A depiction of this map is given in Figure 5.

Finally, out of the 450 cabs, we selected the 50 cabs with the highest number of location updates. We mapped the location updates of the cabs to the map we generated, and used the map to “fill” in the gaps due to missing location updates, for the first 150 minutes. This is done by mapping each two consecutive updates to the map, and finding the shortest route between them. Next, we interpolate a number of locations along this route that is equal to the number of minutes between the location updates. This process allows us to infer the locations of cabs at one-minute granularities. The cab status for those interpolated locations is set to be its reported status in the last location update.

Based on each cab’s mobility profile (obtained as described above), we defined the schedule of the cab as follows: every time the cab is metered, its location is added to the schedule of the cab. This means that, if the cab is hailed (according to location updates), then it has to be in the indicated location at the indicated time. In other words, we can not change the trajectory of a hailed cab. This leaves room for offering mobility hints (detours) to the cab only when it is not hailed.

We compared TFM and Wait At Destination (WAD), a variant of the RND protocol. Under WAD, when not hailed, a cab moves to the next location where it picks up its next

customer, as early as possible, and spends its slack time there waiting for the customer. Throughout the trace evaluation, we assumed that cabs do not exceed speed of 30 mph, which is quite conservative.

**Target Distribution  $D$ :** We assume that the goal of these cabs is to cover the city to track a specific phenomenon that breaks out at random locations. For example, an Amber alert is issued specifying the break out location (the center of the phenomenon) which is given the highest level of attention. Attention awarded to neighboring locations is a function of their distance from the center of the phenomenon. To model this application, we define target coverage distribution  $D$  as follows: We start with a maximum utility value  $M$  and a utility decrement value per hop  $\delta$ . Then we randomly select a field location,  $v_1$ , to be the center of the distribution. We assign virtual utilities  $u_v$  as follows:

$$u_v(v_i) = M - \delta \times |v_1 - v_i| \quad (7)$$

where  $u_v(v_i)$  is the virtual utility assigned to field location  $v_i$ , and  $|v_1 - v_i|$  is the number of hops between  $v_1$  (the center of the distribution) and  $v_i$ . This function assigns to  $v_1$  the maximum value of utility,  $M$ . The utility value of field location  $v_i$  drops as a linear function of the number of hops between  $v_i$  and the center of the distribution,  $v_1$ . To get the required distribution  $D$ , which specifies the percentage of time  $D(v_i)$  that field location  $v_i$  should be covered, we use the following equation:

$$D(v_i) = u_v(v_i) / \sum_{v_x \in V} u_v(v_x) \quad (8)$$

Figure 6 illustrates an example distribution over a compact version of the map.

**Results:** In our comments on Theorem 2, we argued that in case of systems with single nodes, the longer the PFP’s used to estimate the coarse utility of directly neighboring locations, the better the performance, provided we limit our consideration to field locations that are reachable under the scheduling constraints of the node. In this experiment we aim to evaluate this Theorem in systems with multiple nodes, but low node density. Towards this end, we calculated the KL distance of TFM using different values of the maximum length of PFP’s  $h_{max}$ . For each value, we run 20 simulations using the inferred schedule and the generated map. Each simulation

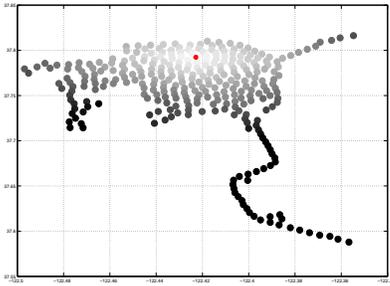


Fig. 6. Example of the distribution we used with the San Francisco cab traces. The lighter the area the higher the coverage percentages ( $D(v)$ ).  $v_1$  in this case is marked by the red (dark) dot in the light area.

TABLE II  
KL DISTANCE RESULTING FROM APPLYING TFM ON CAB TRACES WITH DIFFERENT VALUES OF MAXIMUM LENGTH OF PFP'S  $h_{max}$ .

$h_{max}$	0	1	2	3
KL distance	1.0986	0.8975	0.7441	0.6528

has a different distribution center  $v_1$ . Table II shows the average KL distance for different values of  $h$ .

It is clear that the performance improves by increasing  $h$ , provided that we only consider reachable field locations, confirming our expectations. Since  $h = 3$  yields the lowest value of KL distance, we use this value in the query-based performance evaluation. The KL value of WAD = 2.1, three times that of TFM with  $h = 3$ .

Figure 7 shows the query performance of the two mobility models. TFM achieves from 30% to 120% improvement in QSR over WAD. Increasing the communication range improves the performance of both protocols. However, we found out in another experiment in which we measured performance as a function of the communication range (results not shown here) that this improvement reaches a plateau very fast.

## VI. CONCLUSION

We considered a new mobility model whereby each node is coarsely constrained by an external schedule. These node schedules have slack which can be leveraged to coordinate mobility between nodes so as to satisfy a field monitoring objective. We coin the problem of Constrained Mobility Coordination for Preferential field Coverage (CMC-PC). We show that this problem is NP-complete and propose a distributed heuristic (TFM) that provides nodes with mobility hints (detours) so as to achieve field monitoring that is as close as possible to the required monitoring distribution. We verify the premise of our mobility planning technique using extensive simulations, as well as taxi logs from the San Francisco area. Our results underscore the evident performance improvement attained by TFM.

## REFERENCES

[1] Cabspotting. <http://www.cabspotting.org/api>.  
 [2] S. Bergbreiter and K. Pister. Cotsbots: an off-the-shelf platform for distributed robotics. *Intelligent Robots and Systems, 2003. (IROS 2003), Proceedings. 2003 IEEE/RSJ International Conference on*, 2:1632–1637 vol.2, Oct. 2003.

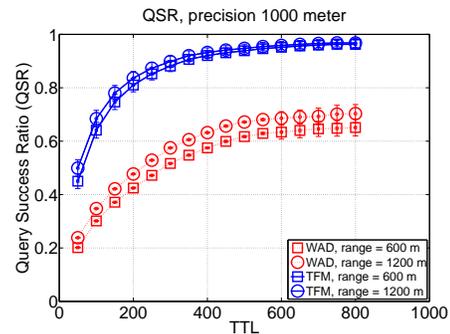


Fig. 7. QSR of TFM and WAD as a function of TTL of samples (TTL measured in minutes). The graph show results when using two values of the communication range: 600 and 1200 meters.

[3] T. M. Cover and J. A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.  
 [4] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme. Robomote: enabling mobility in sensor networks. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 55, Piscataway, NJ, USA, 2005. IEEE Press.  
 [5] S. Dhillon and K. Chakrabarty. Sensor placement for effective coverage and surveillance in distributed sensor networks. *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, 3:1609–1614 vol.3, March 2003.  
 [6] A. Howard, M. J. Mataric, and G. S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *6th International Symposium on Distributed Autonomous Robotics Systems (DASRO2)*, June 2002.  
 [7] C.-F. Huang and Y.-C. Tseng. The coverage problem in a wireless sensor network. In *WSNA '03: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 115–121, New York, NY, USA, 2003. ACM.  
 [8] J.-C. Latombe. *Robot Motion Planning*. Kluwer International Series in Engineering and Computer Science, Norwell, MA, USA, 1991.  
 [9] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, May 2006.  
 [10] J. Lee, A. Dharme, and S. Jayasuriya. Potential field based hierarchical structure for mobile sensor network deployment. *American Control Conference, 2007. ACC '07*, pages 5946–5951, July 2007.  
 [11] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley. Mobility improves coverage of sensor networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 300–308, New York, NY, USA, 2005. ACM.  
 [12] M. McMickell, B. Goodwine, and L. Montestrucque. Micabot: a robotic platform for large-scale distributed robotics. *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, 2:1600–1605 vol.2, Sept. 2003.  
 [13] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *INFOCOM*, pages 1380–1387, 2001.  
 [14] H. Morcos, A. Bestavros, and I. Matta. Preferential field coverage through detour-based mobility coordination. Technical report, Boston University, March 2009. <http://www.cs.bu.edu/research/treports.shtml>.  
 [15] P. D. O., S. H. E., H. Chad, and S. A. C. Robotic deployment of sensor networks using potential fields. In *IEEE International Conference on Robotics and Automation*, April 2004.  
 [16] S. Shakkottai, R. Srikant, and N. Shroff. Unreliable sensor grids: coverage, connectivity and diameter. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, 2:1073–1083 vol.2, March-3 April 2003.  
 [17] W. G. C. G. L. P. T.F. Movement-assisted sensor deployment. *Transactions on Mobile Computing*, 5(6):640–652, June 2006.  
 [18] Y. Yang and M. Cardei. Movement-assisted sensor redeployment scheme for network lifetime increase. In *MSWiM '07: Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, pages 13–20, New York, NY, USA, 2007. ACM.  
 [19] Y. Zou and K. Chakrabarty. Sensor deployment and target localization based on virtual forces. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, 2:1293–1303 vol.2, March-3 April 2003.