

Mobility Coordination of Participants in Mobile CrowdSensing Platforms with Spatio-Temporal Tasks

Christine Bassem
Computer Science Department
Wellesley College
Wellesley, MA
cbassem@wellesley.edu

ABSTRACT

With the increased adoption of Mobile CrowdSensing platforms in urban environments, the mobility of participating self-motivated crowds already roaming in the field can be coordinated to assist in completing spatio-temporal sensing tasks. In this work, we propose to utilize the mobility predictability within such platforms, coupled with the availability of spatio-temporal tasks on the mobility field, to recommend routes for participants in real-time. In this paper, we define the graph-theoretic measure of *Temporal Coverage Centrality*, as well as adopt techniques from distributed systems to define three online route recommendation mechanisms, which can be used to coordinate the mobility of participants within MCS platforms to optimize system performance. Moreover, we empirically evaluate the efficiency of the proposed mechanisms with an insight on their benefits.

KEYWORDS

mobility coordination; routing; spatio-temporal; online allocation; crowdsensing; node centrality

ACM Reference Format:

Christine Bassem. 2019. Mobility Coordination of Participants in Mobile CrowdSensing Platforms with Spatio-Temporal Tasks. In *17th ACM International Symposium on Mobility Management and Wireless Access (MobiWac '19)*, November 25–29, 2019, Miami Beach, FL, USA. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3345770.3356734>

1 INTRODUCTION

Mobile CrowdSensing (MCS) platforms are increasingly used to inexpensively collect sensing data in urban environments, as they leverage the help of participating self-motivated crowds already roaming in a mobility field to assist in completing spatio-temporal sensing tasks, as shown in Figure 1. Sensing tasks can vary from taking images with the camera, to measuring acoustics with the microphone, to answering questions on a mobile application [9].

This work is supported by NSF grant #1755788.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MobiWac '19, November 25–29, 2019, Miami Beach, FL, USA

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-6905-3/19/11...\$15.00

<https://doi.org/10.1145/3345770.3356734>

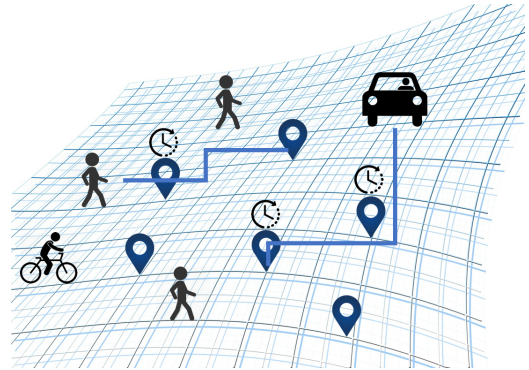


Figure 1: An example of a typical MCS platform, with spatio-temporal tasks and participants roaming around the mobility field using various transportation modes.

In most Mobile CrowdSensing (MCS) platforms, participants have access to the sensing tasks through a central service, which has various degrees of involvement in the process of task allocation. Task allocation is either agent-initiated, in which the participating agents pick the tasks to complete independently [16], or server-initiated, in which the service provider assigns tasks to roaming agents based on system needs and the agents' location, availability, and sensing capabilities [14]. In this work, we envision an MCS platform that coordinates the mobility of its participants, according to their personal mobility constraints, to improve the overall platform's performance, namely *Coordinated-Mobile CrowdSensing (C-MCS)* platforms.

The main premise of C-MCS platforms is that a human agent / participant willingly joins the platform by sharing their journey information, in the form of their two spatio-temporal end-points; representing the start and end locations, and the earliest departure time and the latest arrival time. By sharing their spatio-temporal constraints on their mobility, agents implicitly identify their flexibility, i.e., willingness to deviate from their regular paths to participate in task completion. Agents have an envelope of behaviors according to their degree of flexibility, and they are assumed to be at the disposal of the platform as long as the decisions made are within that envelope, giving the platform the chance to coordinate the mobility of all participants, as shown in Figure 2.

Paper Contribution.

In this paper, we investigate the problem of mobility coordination within C-MCS platforms, as agents¹ join the platform in an online

¹The terms agent and participant are used interchangeably throughout this paper.

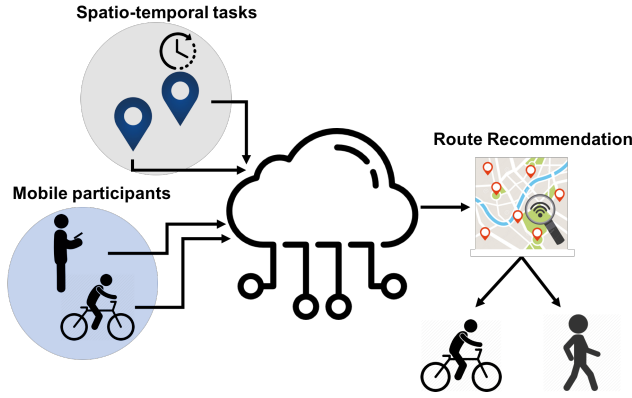


Figure 2: Paths are recommended for participants via a central task allocation service, which takes as input the availability information of spatio-temporal tasks and mobile participants.

model, *i.e.*, their spatio-temporal journey information is available in real-time. We define various online route recommendation mechanisms to coordinate participants mobility, using techniques from graph-theory and distributed systems. Moreover, we develop an evaluation platform to benchmark their performance using both real and synthesized mobility traces.

Paper Outline.

In Section 2, we present our the C-MCS system model, and in Section 3 we define our online route recommendation mechanisms for mobility coordination within such C-MCS platforms. In Section 4, we present the developed evaluation platform, and then discuss the evaluation results. Finally, we conclude with a description of related work in Section 5, and a summary with discussion of future and current work in Section 6.

2 COORDINATED MOBILE CROWDSENSING

In a typical Mobile CrowdSensing (MCS) platform, sensing tasks are distributed over the mobility field, be it a city, campus, or within a smart building, and participants roam freely over that mobility field, according to their personal hidden schedules. On the other hand, in Coordinated MCS (C-MCS) platforms, participants share information about their schedules, giving permission to the platform to coordinate their mobility, in return for profitable path recommendations. In this section, we present the system model of such C-MCS platforms.

2.1 Graph and Tasks

The mobility field can be modeled as a graph with a set of predefined vertices V and edges E , such as a Manhattan graph that can be used to model a structured urban setting as shown in Figure 1. The sensing period varies based on the application, but it is reasonable to assume a day-long period, during which the mobility field graph would typically remain static.

Two types of sensing tasks are considered; spatial and spatio-temporal tasks. Spatial tasks are typically seen in MCS applications when time is not a factor, as opposed to spatio-temporal tasks

that are time-sensitive, such as those within health-based [16] and delivery service [1] applications. We define tasks by the general definition of $(v, [t], [p])$; in which $v \in V$ represents the location of the task, $t \in N$ represents the specified time of completion, and $p \in R$ represents the payoff to be received for completing the task. The values of t and p are optional in the task definition, to allow for one definition for all types of tasks; spatial, spatio-temporal, and free tasks. For the purposes of this paper, we assume that sensing tasks are known by the beginning of the sensing period.

To further optimize this model, we adopt the definition of Weight Evolving temporal (WET) graphs [2] to model both the graph and the spatio-temporal tasks within it. In a WET graph, $G = (V, E, w)$, the temporal weight function, w , represents the reward associated with completing tasks on the mobility field graph. It associates a location, $v \in V$, and time, $t > 0$, with a reward, which represents the payoff collected by completing the tasks with the defined spatio-temporal constraints, (v, t) .

The novelty of the WET graph model lies in the temporal weight function, which links the weight of a vertex to the reward of a specific task at some point in time. This allows for various scoring models that are typically encountered in MCS platforms, such as instantaneous, steady, and time-window scoring. In the *instantaneous* model the reward is valid at a single time step, in the *steady* model the reward is valid for the entire time horizon, *i.e.*, spatial tasks, and in the *time-window* model the reward is valid during a specific time window.

2.2 Mobile Participants

Participants in MCS platforms have a wide range of mobility flexibility based on their personal schedules and needs, which is depicted in Figure 3. Participants with no flexibility always expect to be routed through the shortest path to their destinations, whereas, participants with extreme flexibility can go to any location of their choosing at any time.

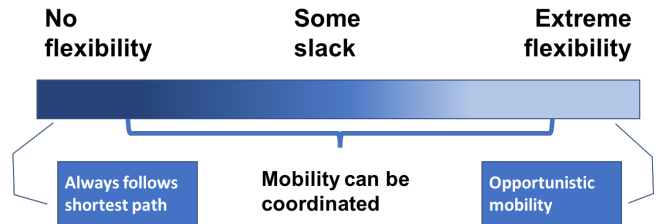


Figure 3: Participants have a wide range of constraints on their mobility; ranging from none flexible, in which strict shortest paths are required, to extremely flexible, in which the participant has no limitations on their mobility. This work considers the middle range of that spectrum, in which participants have some slack in their mobility constraints.

In Coordinated-MCS, we assume that participants have some spatio-temporal constraints, and are willing to share them with the platform in return for profitable path recommendations. In this work, we assume that participants need to only share their spatio-temporal endpoints with the platform, in the form of origin and

destination locations as well as their earliest departure and latest arrival times. This information is then used by the platform for task allocation and path recommendations.

Such an assumption is not very far fetched, since the information required is minimal, and can be easily obtained from the participants beforehand. Participants can provide their temporal constraints accompanied with their spatial constraints through their GPS navigation applications, or can share their schedules with the platform at the beginning of the sensing period. Moreover, a participant's trajectory can be predicted from their repeated behavior, such as going from home to work and vice versa every day.²

Participants join the C-MCS platform by sharing their spatio-temporal endpoints with the platform, and they can join at any time within the sensing period. Given these endpoints and the list of sensing tasks, the platform attempts to allocate tasks to the participant, as shown in Figure 2, with an objective to maximize the reward/payoff collected from completed tasks.

3 ROUTE RECOMMENDATION IN C-MCS

The goal of C-MCS platforms is to coordinate the mobility of its participants to optimize the platform's performance. In this section, we define various online route recommendation algorithms, with an objective to maximize the rewards collected from tasks completed within the platform.

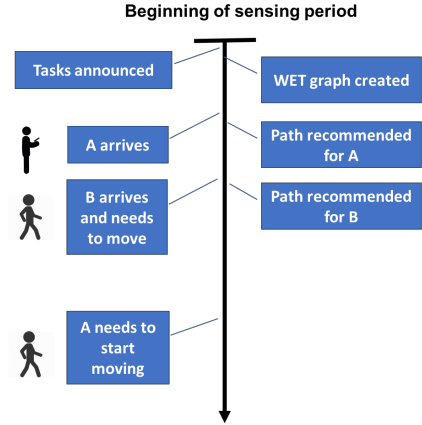
Before defining our online route recommendation algorithms, we differentiate between a participant's time of arrival, *i.e.*, when they share their endpoints, and their earliest time of departure, *i.e.*, when they can start moving towards their destination. Route recommendation can be performed *on-arrival*, regardless of the participant's expected departure time, or *on-move*, delaying the decision as much as possible. We explain this further using the example in Figure 4, in which agents A and B are the participants. Agent A arrives earlier than their earliest departure time, while Agent B arrives exactly when they are able to start moving. For agent B, the routing decision is always made on-arrival. However, for agent A, the routing decision can be made either on-arrival or on-move.

Since the tasks allocated to the participants dictate how they should move along the mobility field, the process of mobility coordination in C-MCS is equivalent to that of task allocation. Algorithm 1 defines the actions that would be taken by the platform throughout the sensing period, with the on-move model of allocation. The on-arrival mode of allocation would be similar, but with the decision made on the arrival of the participant.

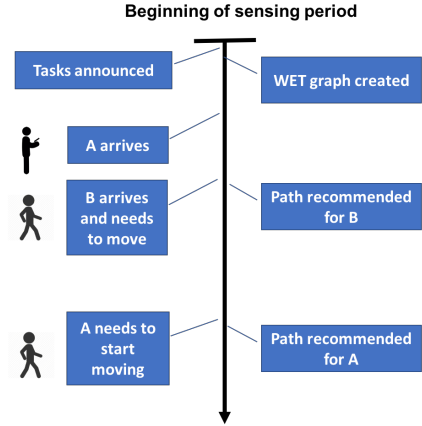
3.1 Individually-Optimal Route Recommendation

An initial straight-forward approach would be to find the optimal set of tasks to be allocated to each individual participant as needed, whether on-move or on-arrival. Given a WET graph, G , and a participant's spatio-temporal endpoints, the objective of an optimal path is to maximize the reward collected from tasks completed on the way. In other words, and without loss of generality, the

² For the purposes of this paper, it is assumed that the spatio-temporal endpoints are known regardless of the method of collection. The ideas mentioned in this paragraph are part of our current and future work.



(a) In this example, the routing decision is made once the participant joins the platform. When agent A moves later in the timeline, no further action needs to be made by the platform.



(b) The same example as above, but with on-move allocation. The routing decision is made just as the agent is ready to move.

Figure 4: Example for different modes of allocation.

objective is to maximize the total reward collected from visiting various WET graph vertices, which could be achieved using any of the optimal routing algorithm defined in [2].

Algorithm 2 takes as input the WET graph and the participant's spatio-temporal endpoints, $(v_{start}, v_{end}, t_{start}, T_{max})$, and finds the optimal path that starts at v_{start} and ends at v_{end} in at most T_{max} time units. The algorithm's optimality is derived from the recurrence of the dynamic program on which it is based, and its running time complexity is $O(T_{max}|E|)$ [2].

3.2 Backfilling Coordination

The Backfilling technique has long been used for scheduling online jobs in distributed systems [3, 25], in which online jobs are grouped together into batches of jobs, and scheduling decisions are made on the batch as a whole to improve the overall performance of the

Algorithm 1 Route recommendation as participants move in C-MCS

Input: $G = (V, E)$ in

- 1: Collect task information (T)
- 2: Generate WET graph $G = (V, E, w)$
- 3: **while** within sensing period **do**
- 4: **for** each arriving participant a **do**
- 5: Mark a 's earliest departure time
- 6: **end for**
- 7: **for** each participant's a departure time **do**
- 8: **Decide on tasks to be allocated**
- 9: **for** each task t chosen **do**
- 10: Allocate t to a
- 11: Remove t from T
- 12: **end for**
- 13: **end for**
- 14: **end while**

Algorithm 2 Optimal reward-maximizing routing on WET graphs with instantaneous tasks.

Input: $G = (V, E, w)$, (v_{start}, v_{end}) , T_{max} in

Output: Optimal routes from v_{start} to all destinations out

- 1: Create the time-ordered graph $G^{T_{max}}$ from G
- 2: Create array M of size $(|V| \times T_{max})$
- 3: **for** $t = 2$ to T_{max} **do**
- 4: **for** each vertex $v \in V^t$ **do**
- 5: **for** each vertex $u \in V^{t-1}$ s.t. $(u, v) \in E^{t-1, t}$ **do**
- 6: Choose u with maximum $M[u, t - 1]$
- 7: **end for**
- 8: $M[v, t] = M[u, t - 1] + \sum_{v_r} w(r, v, t)$
- 9: **end for**
- 10: **end for**
- 11: **return** M

system. The efficiency of this approach relies greatly on how long a job can be forced to wait to place it with a larger batch of jobs. However, in crowdsensing platforms, forcing a participant to wait and delaying them from reaching their destination is not an option.

Therefore, in this mechanism, we heavily rely on the differentiation between arrival time and earliest departure time, and all decisions can only be made on-move. As shown in Algorithm 3, the allocation is decided based on the available set of participants. A participant is available if they have already joined the system, by submitting their spatio-temporal constraints, but is not ready to move yet. The algorithm defined is a $\frac{1}{2}$ -approximation greedy algorithm[4], in which the utility of an agent/participant can be defined as the total path length taken, the total revenue collected, or some other sub-modular cost function. Although the mechanism below decides on how to distribute the tasks to all available participants, only the participant moving at the moment of executing that mechanism is provided with task recommendations.

The running time complexity of this mechanism is $O(|T| * |V| * cost)$, in which $cost$ represents the complexity of computing the utility function of the participant, based its definition. Although the algorithm itself is a $\frac{1}{2}$ -approximation of the optimal, the whole

Algorithm 3 Backfilling-based route recommendation mechanism for a specific participant (theAgent)

- 1: Sort tasks in decreasing order by their payoff/reward
- 2: **for** each task t **do**
- 3: **for** each available agent a **do**
- 4: **if** t is feasible for a **then**
- 5: Calculate the utility of a if t is allocated to them
- 6: **end if**
- 7: **end for**
- 8: Find the agent a^* with highest utility
- 9: Mark t for a^*
- 10: **end for**
- 11: **return** the tasks marked for theAgent

route recommendation process isn't guaranteed to be as efficient, since only a single agent follows the allocation decided by the algorithm at a time.

3.3 Centrality-based Coordination

Although the backfilling-based recommendation is very efficient, it is highly dependent on participants arriving a good amount of time before their earliest departure time, which is not always the case in C-MCS applications. If all participants arrive exactly before their earliest departure time, as in the case of using GPS navigation applications, then all allocations are focused on increasing the utility of each individual participant, leading to a performance similar to that of the Individually-Optimal route recommendation mechanism defined above.

To overcome this limitation, we propose using the node coverage centrality metric, as defined in Equation 1, which is a novel centrality measure that captures the frequency of traversing a node given a set of spatio-temporal tasks on a WET graph [2].

The definition of *Coverage Centrality*, as shown in Equation 1, is adopted from the classic definition of Betweenness Centrality [8]. A node's coverage centrality is defined as the ratio between the number of optimal routes between all pairs of spatio-temporal endpoints on the WET graph that pass through that node, and the total number of optimal routes between all pairs of spatio-temporal endpoints on the graph.

$$coverage(v) = \frac{\sum_{v(i,j,t,T_{max})} \pi_v(i,j,t,T_{max})}{\sum_{v(i,j,t,T_{max})} \pi(i,j,t,T_{max})} \quad (1)$$

In which (i, j, t, T_{max}) represents a set of spatio-temporal endpoints and can be further expanded to $\sum_{i \in V} \sum_{j \in V} \sum_{1 \leq t \leq T} \sum_{1 \leq T_{max} \leq T}$. The function $\pi(i, j, t, T_{max})$ represents the number of optimal routes between the spatio-temporal endpoints, (i, t) and $(j, t + T_{max})$, and $\pi_v(i, j, t, T_{max})$ represents the number of optimal routes between the spatio-temporal endpoints, (i, t) and $(j, t + T_{max})$, which includes the vertex v .

This coverage centrality metric can be computed at the beginning of the sensing period, as the task information becomes available. However, this metric, as defined in [2] is very inefficient to compute frequently, since it's running the optimal Algorithm 2 for every possible set of spatio-temporal endpoints.

Algorithm 4 Centrality-based route recommendation algorithm

```
1: Sort tasks in  $T$  in increasing order based on the coverage centrality of their locations
2: for each task  $t$  in  $T$  do
3:   if  $t$  is on  $a$ 's route to destination and feasible then
4:     Allocate  $t$  to  $a$ 
5:   end if
6: end for
```

In urban environments, we have the advantage that participant mobility is fairly predictable [7, 13]. Therefore, we propose that the metric be computed based on the history of spatio-temporal endpoints encountered by the system, as shown in Equation 2.

$$coverage(v) = \frac{\sum_{(i,j,t,T_{max}) \in H} \pi_v(i,j,t,T_{max})}{\sum_{(i,j,t,T_{max}) \in H} \pi(i,j,t,T_{max})} \quad (2)$$

This temporal definition of the coverage centrality metric reduces the complexity of computing the centrality value, while maintaining the most relevant information on the participants' most common routes. Various improvements can be further done to this metric, such as to include randomly chosen spatio-temporal endpoints in the computation. However, our evaluation results show that the effect of these improvements is minimal on the task allocation process.

4 PERFORMANCE EVALUATION

In this section, we present our evaluation platform, and present the results obtained from evaluating the defined online route recommendation mechanisms.

4.1 Evaluation Platform

A discrete event simulator has been developed with the purpose of evaluating the effectiveness of various task allocation mechanisms on the performance of C-MCS platforms. The simulator has a modular design, which allows for the addition of new allocation mechanisms. Moreover, all activities occurring in the simulated environment are traced to allow for new metrics to be measured in the future.

Task Model. Due to the lack of crowdsensing datasets, with recorded sets of spatio-temporal tasks, we generate our own random sets of spatio-temporal tasks on various graph structures. Tasks are created with randomly chosen locations, spatially distributed uniformly over the mobility field, and with temporal demand that is generated according to an exponential distribution with a mean that is a parameter of the simulation.

Agent Model. For the purposes of this paper, we generate pairs of spatio-temporal endpoints to mimic participants' behaviors within the platform. The actual mobility pattern of the participants is a result of the allocation mechanism adopted in the simulation, which greatly depends on the distribution of the spatio-temporal tasks on the mobility field.

Participants' spatio-temporal endpoints are generated using two models; random and urban. In the random model, the spatial endpoints are chosen uniformly at random, while in the urban model, typical daily mobility behavior is modeled, in which agents move

from home to the city center in the morning, and move back at the end of the day. For both models, the temporal demand is generated according to an exponential distribution with a mean that is a parameter of the simulation.

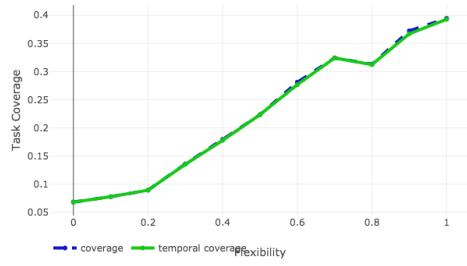
Evaluation Metrics. In the experiments shown below, we measure the system's participation ratio, task coverage, rewards ratio, and agent deviation. The system's participation ratio is the ratio between participants generating revenue and the total number of participants, its task coverage is the ratio between completed tasks and all available tasks, its revenue collected is the ratio between rewards collected and maximum possible rewards, and its agent deviation is the ratio between the agent's final path length and shortest path length. All results shown below are averages of 30 executions of the simulation with varying random seeds.

4.2 Effectiveness of Temporal Coverage

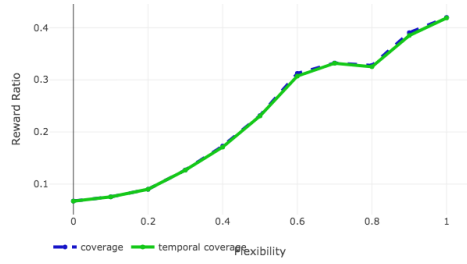
Initially, we evaluate the effectiveness of the defined temporal coverage metric, when compared to the classical coverage centrality metric. Due to the lack of any optimizations to the classical metric, it is computationally inefficient to execute it as defined for larger graphs. Therefore, all experiments in this subsection are performed on a 7x7 Manhattan Grid, with a simulation time of 6 hours, or 360 time units, and 300 spatio-temporal tasks uniformly distributed over the grid.

In the results shown in Figure 5 represent the results obtained by varying the flexibility factor of a 50 agents. An agent's flexibility factor indicates the increase in path length that an agent is willing to accept. A flexibility factor of 0.5 means that the agent might accept paths that are 1.5 the length of their shortest paths. These results show that the temporal coverage metric performs as well as the classical coverage metric, even as the flexibility factor is increased. They also indicate the power of flexibility, as the task coverage ratio more than doubled with a small increase of flexibility from 0.4 to 1, while maintaining a participation ratio of less than 15%.

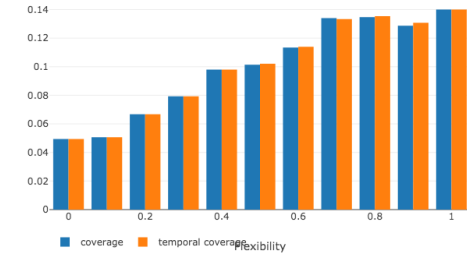
Another interesting pattern is that the agent deviation ratio remains at 1, *i.e.*, no deviation, until an adequate flexibility value is reached, which highly increases the task coverage and rewards ratio, along with a slight increase in participation. This due to the coordination aspect of the allocation, as participants have an increased envelope of choices, due to the increased flexibility, longer routes are chosen to optimize for task coverage, to avoid congestion around most popular tasks.



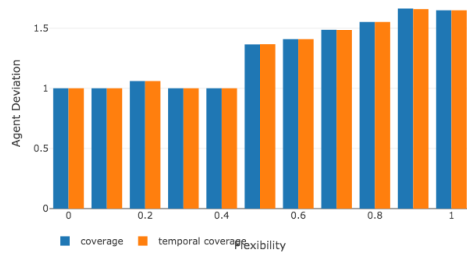
(a)



(b)



(c)



(d)

Figure 5: Task coverage and rewards increase at a high rate with the increase of flexibility, even as participation ratio remains at a low value.

In Figure 6, represents the results of varying the number of participants roaming the field, with uniform mobility models, and a flexibility factor of 0.5. The task coverage ratio increases at a similar rate as the increased number of agents, which is due to the rigid schedules of users that make deviations less likely. This scenario resembles current MCS platforms, in which participants aim to minimize their trip duration, thus limiting the possibility of mobility coordination and improved task coverage with lower participation rates.

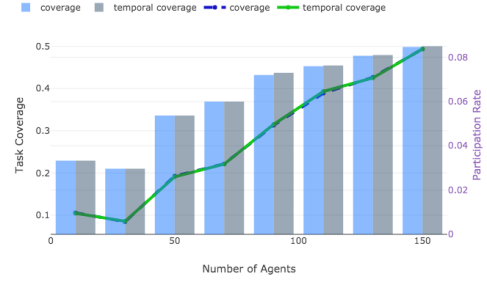


Figure 6: With less flexibility, many participants are needed to coordinated to cover as many tasks as possible. Even with the increased number of participants, the participation ratio keeps increasing to cover more tasks.

This is not the case in Figure 7, which has the same exact simulation parameters except for a flexibility factor of 2. In this setting, participants can be coordinated to take longer routes, thus covering more tasks with less participation rate. Moreover, we note that the task coverage ratio seems to plateau after a certain number of agents, which indicates a saturation in the mobility field.

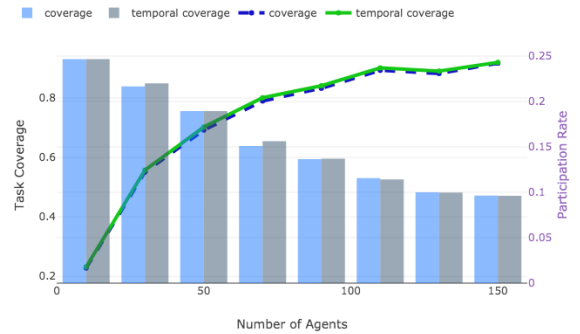


Figure 7: The effect of increasing the flexibility factor of the agents can be seen in the decreasing participation ratio, even with similar task coverage rates.

4.3 Optimal Routing vs. Backfilling

In the next set of experiments, we evaluate the performance of all proposed mechanisms against each other. Since we have shown that

temporal coverage can be used in place of the classical coverage metric, we can increase the graph sizes. All experiments in this subsection are performed on a 20x20 Manhattan Grid, with a simulation time of 6 hours, or 360 time units, and 3000 spatio-temporal tasks uniformly distributed over the grid.

In these experiments we compare the performance of all proposed mechanisms against each other, as well as against **shortest path**, in which the shortest path is chosen for all agents, and **individual-revenue**, in which the agent picks one task at a time based on its revenue. We vary the value of the flexibility factor with the same number of agents of 1000.

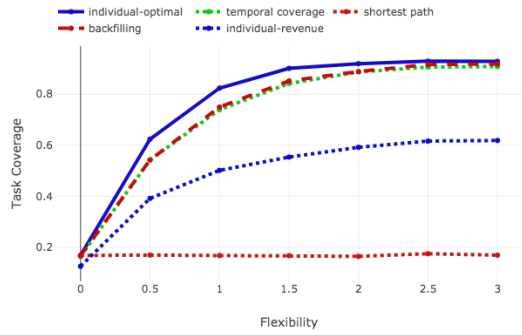


Figure 8: The performance of the temporal coverage and backfilling mechanisms keep improving, until it almost reaches that of the optimal.

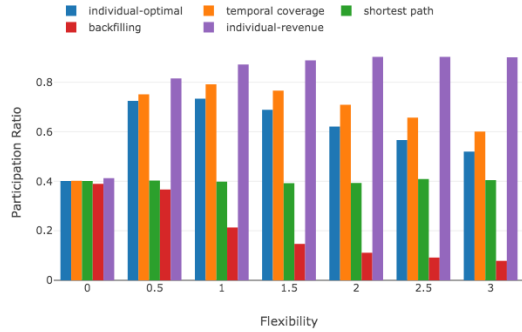


Figure 9: An advantage of the backfilling technique is that it employs the help of less participants while gaining almost the same task coverage as other efficient mechanisms.

All of the results obtained in these experiments have shown that the temporal coverage coordination mechanism performs as well as the backfilling mechanism (with 30 minute earlier arrival rate). They indicate that the individual optimal is superior to all other

mechanisms no matter the value of flexibility, which is due to the uniformity of task distribution over space and time³

Moreover, for the backfilling-based mechanism, we note the decreased participation rate accompanied with the increased task coverage as the flexibility increases. This is due to the nature of the backfilling technique itself, as it groups participants together before making allocation decisions, thus further optimizing the allocation.

5 RELATED WORK

Task Allocation.

Task allocation in MCS platforms can be generally classified into the two categories of single-task and multi-task allocation. Both of which can be used to optimize various system objectives, such as participation, quality of work, energy efficiency, and task completion among many.

In single-task allocation mechanisms, participants are typically assigned a single task in each allocation cycle, such as in [6], in which auctions are used to decide on optimal and truthful task allocation. In [12], the privacy of bidders is a concern of the system, and a group-based bidding approach is adopted. In work done by Xiong et al. in [24], the objective was to increase task coverage while minimizing the energy consumed by the participating devices, while using the Piggyback approach as defined in their earlier work on CrowdTasker [23].

On the other hand, multi-task allocation mechanisms attempt to coordinate the allocation of tasks to optimize for the system as a whole, such as in PSAllocator [19], in which participants movement is predicted, and then the allocation problem is solved as a bipartite matching problem. Similarly in TRACCS [5] and [22], optimization techniques are used to coordinate participant mobility to maximize the total payoff from assigned tasks. Multi-task allocation can also be used for assigning interdependent tasks as in [20], in which the quality of each task is ensured as interdependent tasks are being allocated.

Node Centrality.

For classically defined temporal graphs/networks, the centrality of a node was measured based on number of shortest paths that pass through that node in [18]. In [10], temporal graphs are modeled as a sequence of static graphs, labeled with the time the edge existed. In [11], the authors presented time-ordered graphs, which is a powerful model for characterizing a temporal graph, and redefined basic temporal centrality measures accordingly. In [17], the notion of temporal vertices is defined, which are robust against time scale changes, and efficiently compute various centrality values.

More centrality measures have been proposed for temporal graphs, as in [15, 21]. However, all of these measures are not suitable for identifying central nodes in a temporal graph model such as WET graphs, since their main focus is on the evolution of graph edges and the graph's connectivity over time. In [2], the notions of weight evolving temporal graphs and coverage centrality have been defined, but their effect on online mobility coordination has not been clearly investigated. In this work, we further develop

³Evaluating the performance of these mechanisms with various task distribution models is part of our future work.

these definitions, and compare them against existing optimal routing algorithms designed for offline route coordination in C-MCS platforms, such as those defined in [4].

6 CONCLUSION AND FUTURE WORK

In this paper, we presented the notion of using task allocation mechanisms to coordinate the mobility of online participants in Coordinated-MCS platforms. Given the difficulty of coordinating crowd mobility in general, the requirement of allocating spatio-temporal tasks to them adds to the complexity of the coordination problem. We defined several route coordination mechanisms, and evaluated them empirically. Moreover, we defined the temporal coverage centrality metric, and showed it to be equally effective for mobility coordination when compared to the classical coverage centrality metric.

This paper is a first step towards a deeper study of mobility coordination mechanisms for C-MCS platforms. Our future work is two parts; theoretical and practical. On the theoretical front, we plan to further investigate other coordination mechanisms, as well as define appropriate incentive mechanisms to accompany them. Moreover, we plan to investigate the usefulness of the temporal coverage metric in other aspects of MCS platforms, such as resource placement. On the practical front, we plan to evaluate the effectiveness of our mobility coordination mechanisms on real crowds, specifically in a college campus setting, in which mobility schedule can be defined beforehand.

REFERENCES

- [1] [n. d.]. Uber Eats - Your Favorite Restaurants, Delivered Fast</title>. u.ubereats.com. Accessed: 2019-06-30.
- [2] Christine Bassem. 2019. Redefining Node Centrality for Task Allocation in Mobile CrowdSensing Platforms. In *Smart Computing (SMARTCOMP), 2019 IEEE International Conference on*. IEEE.
- [3] Christine Bassem and Azer Bestavros. 2016. *Scheduling of data-intensive workloads in a brokered virtualized environment*. Technical Report. Computer Science Department, Boston University.
- [4] Christine Bassem and Azer Bestavros. 2017. GuideMe: Routes coordination of participating agents in mobile crowd sensing platforms. In *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 4043–4049.
- [5] Cen Chen, Shih-Fen Cheng, Aldy Gunawan, Archan Misra, Koustuv Dasgupta, and Deepthi Chander. 2014. Tracss: a framework for trajectory-aware coordinated urban crowd-sourcing. In *Second AAAI Conference on Human Computation and Crowdsourcing*.
- [6] Zhenni Feng, Yanmin Zhu, Qian Zhang, Lionel M Ni, and Athanasios V Vasilakos. 2014. Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing. In *INFOCOM, 2014 Proceedings IEEE*. IEEE, 1231–1239.
- [7] Laura Ferrari, Alberto Rosi, Marco Mamei, and Franco Zambonelli. 2011. Extracting urban patterns from location-based social networks. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*. ACM, 9–16.
- [8] Linton C Freeman. 1978. Centrality in social networks conceptual clarification. *Social networks* 1, 3 (1978), 215–239.
- [9] Bin Guo, Zhiwen Yu, Xingshe Zhou, and Daqing Zhang. 2014. From participatory sensing to mobile crowd sensing. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*. IEEE, 593–598.
- [10] David Kempe, Jon Kleinberg, and Amit Kumar. 2002. Connectivity and inference problems for temporal networks. *J. Comput. System Sci.* 64, 4 (2002), 820–842.
- [11] Hyounghick Kim and Ross Anderson. 2012. Temporal node centrality in complex networks. *Physical Review E* 85, 2 (2012), 026107.
- [12] Ting Li, Taeho Jung, Hanshang Li, Lijuan Cao, Weichao Wang, Xiang Yang Li, and Yu Wang. 2017. Scalable privacy-preserving participant selection in mobile crowd sensing. In *2017 IEEE International Conference on Pervasive Computing and Communications, PerCom 2017*. IEEE, 59–68. <https://doi.org/10.1109/PERCOM.2017.7917851>
- [13] Xiaolong Li, Gang Pan, Zhaohui Wu, Guande Qi, Shijian Li, Daqing Zhang, Wangsheng Zhang, and Zonghui Wang. 2012. Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science* 6, 1 (2012), 111–121.
- [14] Yan Liu, Bin Guo, Yang Wang, Wenle Wu, Zhiwen Yu, and Daqing Zhang. 2016. TaskMe: Multi-task Allocation in Mobile Crowd Sensing. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 403–414. <https://doi.org/10.1145/2971648.2971709>
- [15] Vincenzo Nicosia, John Tang, Cecilia Mascolo, Mirco Musolesi, Giovanni Russo, and Vito Latora. 2013. Graph Metrics for Temporal Networks. (jun 2013). https://doi.org/10.1007/978-3-642-36461-7_2 arXiv:1306.0493
- [16] Rüdiger Pryss, Manfred Reichert, Berthold Langguth, and Winfried Schlee. 2015. Mobile crowd sensing services for tinnitus assessment, therapy, and research. In *Mobile Services (MS), 2015 IEEE International Conference on*. IEEE, 352–359.
- [17] Taro Takaguchi, Yosuke Yano, and Yuichi Yoshida. 2016. Coverage centralities for temporal networks. *The European Physical Journal B* 89, 2 (2016), 35.
- [18] John Tang, Mirco Musolesi, Cecilia Mascolo, Vito Latora, and Vincenzo Nicosia. 2010. Analysing information flows and key mediators through temporal centrality metrics. In *Proceedings of the 3rd Workshop on Social Network Systems*. ACM, 3.
- [19] Jiangtao Wang, Yasha Wang, Daqing Zhang, Feng Wang, Yuanduo He, and Liantao Ma. 2017. PSAllocator: Multi-Task Allocation for Participatory Sensing with Sensing Capability Constraints. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing - CSCW '17*. ACM Press, New York, New York, USA, 1139–1151. <https://doi.org/10.1145/2998181.2998193>
- [20] Jiangtao Wang, Yasha Wang, Daqing Zhang, Feng Wang, Haoyi Xiong, Chao Chen, Qin Lv, and Zhaopeng Qiu. 2018. Multi-Task Allocation in Mobile Crowd Sensing with Individual Task Quality Assurance. *IEEE Transactions on Mobile Computing* (2018), 1–1. <https://doi.org/10.1109/TMC.2018.2793908>
- [21] Matthew J. Williams and Mirco Musolesi. 2015. Spatio-temporal networks: reachability, centrality and robustness. (jun 2015). <https://doi.org/10.1098/rsos.160196> arXiv:1506.00627
- [22] Mingjun Xiao, Jie Wu, Liusheng Huang, Yunsheng Wang, and Cong Liu. 2015. Multi-task assignment for crowdsensing in mobile social networks. In *Proceedings - IEEE INFOCOM*, Vol. 26. IEEE, 2227–2235. <https://doi.org/10.1109/INFOCOM.2015.7218609>
- [23] Haoyi Xiong, Daqing Zhang, Guanling Chen, Leye Wang, Vincent Gauthier, and Laura E Barnes. 2015. iCrowd: Near-Optimal Task Allocation for Piggyback Crowdsensing. (2015), 1536–1233. <https://doi.org/10.1109/TMC.2015.2483505>
- [24] Haoyi Xiong, Daqing Zhang, Guanling Chen, Leye Wang, Vincent Gauthier, and Laura E. Barnes. 2016. iCrowd: Near-Optimal Task Allocation for Piggyback Crowdsensing. *IEEE Transactions on Mobile Computing* 15, 8 (aug 2016), 2010–2022. <https://doi.org/10.1109/TMC.2015.2483505>
- [25] Yanyong Zhang, Hubertus Franke, JE Moreira, and Anand Sivasubramaniam. 2000. Improving parallel job scheduling by combining gang scheduling and backfilling techniques. In *Proceedings 14th International Parallel and Distributed Processing Symposium. IPDPS 2000*. IEEE, 133–142.