Redefining Node Centrality for Task Allocation in Mobile CrowdSensing Platforms

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

***Abstract*—With the recent developments in Mobile CrowdSensing, an interesting model of temporal graphs has emerged, in which node weights evolve over time, according to the availability of spatio-temporal tasks on the mobility field. The analysis and understanding of these types of graphs, namely *Weight Evolving Temporal (WET) graphs*, is critical for optimizing task allocation in such crowdsensing platforms. In this paper, we formally define WET graphs and their corresponding routing problem, in which the objective of the routing is to maximize the reward collected from vertices visited amid the graph traversal. By modeling a WET graph as a time-ordered graph, we define efficient and optimal routing algorithms, and theoretically analyze them. Moreover, we present a novel node centrality measure, namely *Coverage Centrality*, that captures the popularity of various nodes of the WET graph, and which we incorporate in an online crowdsensing task allocation mechanism to increase task coverage. Finally, we evaluate the efficacy of this novel centrality measure on different types of graphs, when compared to other centrality measures, and evaluate its effect on task coverage in online mobile crowdsensing platforms.**

***Index Terms*—routing, centrality, temporal graphs, mobile crowdsensing, task allocation**

I. Introduction

With the recent advances in Mobile CrowdSensing (MCS), participating self-motivated crowds already roaming in a mobility field can be used to assist in completing spatio-temporal sensing tasks, such as taking images with the camera or measuring acoustics with the microphone [8]. The advantages of such a model is that by employing the help of the crowd, massive sensory information can be collected without the cost of setting up a physical infrastructure [4, 11, 9]. Moreover, it allows for the development of new smart services and applications that involve the humans in the loop of the sensing process, which would have otherwise been impossible, such as environmental applications [14] and health-care applications [18].

In a typical Mobile CrowdSensing (MCS) platform, a service provider acts as an intermediary between the tasks that need to be completed and the mobile participants that can complete them, as shown in Fig. 1. The main challenges in such platforms are the spatio-temporal constraints of the tasks and the unpredictable participant participation and human mobility patterns, which affect the efficiency of task allocation; in terms of rate of task completion, the coverage of the field, and the quality of service guarantees of the tasks [1, 15, 22].

This work is supported by NSF grant #1755788.

Fig. 1. An example of a typical MCS platform, as depicted in [16]

In our work, we consider MCS platforms with spatio-temporal tasks that need to be completed at a their specified location and time. These types of tasks exist in various medical and environmental applications, as well as smart-city services, and they add to the complexity of task allocation in MCS platforms. The constraints of these spatio-temporal tasks create a novel temporal graph model, in which vertex weights evolve over time. In this paper, we formalize this definition of Weight Evolving Temporal (WET) graphs that allow for an accurate representation of the mobility field in MCS platforms.

This novel definition of WET graphs allows for the definition of efficient task allocation mechanisms within MCS platforms in two ways; by developing optimal routing algorithms to be used by participants in the platform, and by defining a novel centrality measure that represents the popularity of the graph nodes, *i.e.*, locations in the platform. In this paper, we define optimal spatio-temporal routing algorithms on WET graphs, and expand on them to define the *Coverage Centrality* measure that takes into account the effect of both the set of spatio-temporal tasks as well as the graph structure on the centrality of nodes in a WET graph.

Paper Contribution.
This paper focuses on defining new graph and algorithmic measures that address the problem of task allocation in Mobile CrowdSensing platforms with spatio-temporal tasks. This is achieved through;
1) Identifying a novel type of temporal graphs, namely Weight Evolving Temporal graphs.

2) Defining optimal routing algorithms on such WET graphs, for various reward models.
3) Defining a novel centrality measure on WET graphs.
4) Developing efficient online task allocation mechanisms.

Paper Outline.
In Section 2, we define the structure of Weight Evolving Temporal graphs. In Section 3, we define the spatio-temporal routing problem on WET graphs, with optimal algorithms to solve it, as well as the novel coverage centrality measure. In Section 4, we present efficient online task allocation mechanisms based on the definition of WET graphs, which we evaluate together with the proposed centrality measure in Section 5. Finally, we present a quick overview of related work in centrality measures and task allocation in MCS platforms in Section 6, and conclude the paper with a summary of results and our plans for future work.

II. Weight Evolving Temporal Graphs

With the evolving nature of spatio-temporal tasks in a mobile crowdsensing setting, a novel temporal graph model arises, one which effectively represents the mobility field and the constantly evolving task rewards, in both space and time. In this section, a novel temporal graph model, namely the Weight Evolving Temporal (WET) graph, is defined.

A. WET Graph Model

We define a Weight Evolving Temporal (WET) graph as $G = (V, E, w)$, in which the set of temporal vertices V represents the various landmarks in the mobility field, the set of edges E represents the links, *i.e.*, streets, between these landmarks, and the temporal weight function, w, represents the reward associated with completing tasks on the mobility graph. The weight function associates a location, $v \in V$, and time, $t > 0$, tuple with a reward, which represents the reward collected by completing a task with the defined spatio-temporal constraints, (v, t). We note that each vertex may have varying weights though time, depending on the available spatio-temporal tasks.

The definition of WET graphs can be expanded to encompass various realistic settings, in which locations and/or roads can become evolve and become inaccessible over various periods of time, by the use of time-ordered graphs[13]. However, for the purposes of this paper, since shorter windows of time are usually considered in the process of task allocation in MCS platforms, we focus on graphs with static sets of vertices and edges. Moreover, we assume that time is discrete and finite and that movement between landmarks, *i.e.* along an edge, is completed in a single discrete time step [1].

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, and in the *time-window* model the reward is valid during a specific time window.

To better explain these scoring models, we present the illustrative examples below, in which we follow a typical characterization of a temporal graph by splitting it into a sequence of static snapshots as in [1].

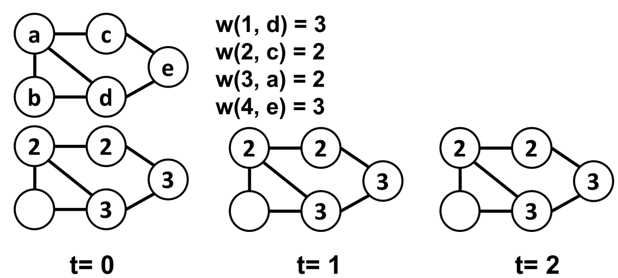


Fig. 2. A time series representation of a WET graph with steady tasks.

In Fig. 2, we present an example of the steady model, in which the reward of visiting a vertex is valid for the entire time horizon of the graph. An aggregated view of the graph would provide a correct representation of the state of the MCS platform.

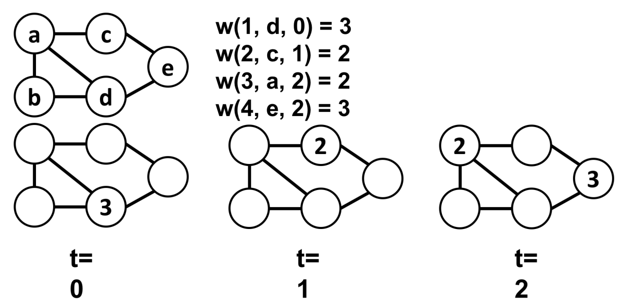


Fig. 3. A time series representation of a WET graph with instantaneous tasks.

In Fig. 3, we present an example of the instantaneous model, in which the reward of visiting a vertex is only valid for a single unit of time, depending on the constraints of the tasks. As opposed to the steady model, an aggregated view of the graph would not provide a correct representation of the available sensory tasks, but a detailed definition of the weight function is necessary. For the purposes of this paper, the weight function is stored in a dictionary-based data-structure, and optimizing this storage model is part of our future work.

Finally, for the time-window-reward model, the temporal weight function would need to include the task identifier, to allow for the differentiation between multiple tasks on the same location. The study of this graph variant is part of ongoing work, which is out of the scope of this paper.

[1] We expand on that last assumption in the discussion next section.

III. Routing and Centrality on Weight Evolving Temporal Graphs

Since Weight Evolving Temporal (WET) graphs present an efficient method to model mobility fields in a mobile crowd-sensing setting, appropriate routing and centrality models need to be defined for such graphs.

A. Spatio-Temporal Routing on WET Graphs

Consider an MCS platform participant, who has a predetermined itinerary, *i.e.*, start and end points, and who is willing to complete a few spatio-temporal tasks on the way. For such a participant, existing routing algorithms would not be suitable, since they would prefer a route that maximizes the reward collected from completing tasks, while remaining in the confines of their itinerary constraints.

1) Problem Definition: Given a WET graph, G, representing the mobility field as well as the rewards of tasks on that field, and a participant's spatio-temporal endpoints, (v_{start}, t_{start}) and (v_{end}, t_{end}), the objective of routing is to find the optimal route between these endpoints that maximizes the reward collected from tasks completed on the way. A spatio-temporal task can be completed only if its location is traversed within its specified time. In other words, and without loss of generality, the objective is to maximize the total reward collected from visiting various WET graph vertices.

Unlike the Orienteering problem [7], the optimal route in this model allows for cycles and staying at nodes, as long as the spatio-temporal constraints of the journey are not violated. In other words, the participant must leave from v_{start} not before t_{start}, and reach their destination, v_{end}, at a time no later than t_{end}. Thus, preventing endless cycles and starvation of nodes.

2) Routing Algorithms on WET Graphs: In this section, we define optimal routing algorithms for WET graphs with instantaneous-reward tasks and steady-reward tasks. The routing algorithm for WET graphs with time-window-reward tasks is part of previous work in [1].

1) Instantaneous-Reward Tasks.
For a WET graph with instantaneous tasks, a dynamic program can be defined to optimally solve the routing problem, with a recurrence as shown in (1).

$$\begin{aligned} OPT(v,t) = \sum_{1 \leq r \leq R} & w(r,v,t) + \\ & max\Big\{ OPT(v,t-1), \\ & max_{u \in Neighbor(v)}\{OPT(u,t-1)\}\Big\} \end{aligned} \tag{1}$$

where the optimal reward can be computed by evaluating $OPT(v_{end}, Tmax)$.

We define the bottom-up Algorithm 1 to solve the recurrence in (1), in which a 2-dimensional array can be used to keep track of the optimal reward collected with every time step on the route, requiring a space complexity of $O(T_{max}|V|)$. The time complexity of the algorithm can be improved from $O(T_{max}|V|^2)$ to $O(T_{max}|E|)$ by adopting constrained time-ordered graphs, which are used to effectively represent temporal graphs as directed graphs.

In a time-ordered graph [13], $G^T = (V^T, E^{T-1})$, vertices are tagged with the time step, and edges represent single time step traversals between adjacent vertices on the original graph. We slightly modify this definition to create a constrained time-ordered graph based on the spatio-temporal constraints of the route. For the spatio-temporal endpoints, $(v_{start}, 1)$ and (v_{end}, T_{max}), a graph with T_{max} time slices is generated. For $t = 1$, only v_{start} is included, and for each $t \in 2...T_{max}$, a vertex v^t is included iff it has edges from some node u^{t-1}, or it's the same vertex, and the distance from that vertex v to v_{end} is at most $T_{max} - t$. If the pair-wise distances is known, the complexity of constructing a constrained time-ordered graph is $O(T_{max}|E|)$.

Algorithm 1 Optimal reward-maximizing routing on WET graphs with instantaneous tasks.

```
Input: G = (V, E, w), (v_start, v_end) in
Output: Optimal routes from v_start to all destinations out
Create the time-ordered graph G^{T_max} from G
Create array M of size (|V| × T_max)
for t = 2 to T_max do
    for each vertex v ∈ V^t do
        for each vertex u ∈ V^{t-1} s.t. (u, v) ∈ E^{t-1,t} do
            Choose u with maximum M[u, t − 1]
        end for
        M[v, t] = M[u, t − 1] + Σ_{∀r} w(r, v, t)
    end for
end for
return M
```

As mentioned above, using constrained time-ordered graphs reduces the complexity of the Algorithm 1 to $O(T_{max}|E|)$.

2) Steady-Reward Tasks.
With steady tasks, a value is associated with a vertex in the graph for the duration of the entire time horizon. However, since the weight of a vertex should only be counted once on the journey, no matter how many times it has been visited, we need to incorporate an extra data structure to keep track of all vertices visited, *i.e.* tasks completed, throughout the journey. A modified version of the proposed algorithm is presented in Algorithm 2. The optimality of this algorithm can be proven similarly as above.

3) Discussion.
More Types of Graphs: For graphs with variable edge costs, *i.e.*, not of the same length, the recurrence defined above can be modified to,

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

Input: $G = (V, E, w)$, (v_{start}, v_{end}) in
Output: Optimal routes from v_{start} to all destinations out
Create the time-ordered graph $G^{T_{max}}$ from G
Create array M of size $(|V| \times T_{max})$
Create array N of size $(|V| \times T_{max})$
Set $N[v, t] = v$ for all v and t
for $t = 2$ to T_{max} **do**
  for each vertex $v \in V^t$ **do**
    for each vertex $u \in V^{t-1}$ s.t. $(u, v) \in E^{t-1,t}$ **do**
      Only consider u s.t. $v \notin N[u, t-1]$
      Choose u with maximum $M[u, t-1]$
    end for
    $M[v, t] = M[u, t-1] + \sum_{\forall r} w(r, v, t)$
    $N[v, t] = N[u, t-1] + N[v, t]$
  end for
end for
return M

$$OPT(v,t) = \sum_{1 \le r \le R} w(r,v,t) + max\Big\{OPT(v,t-1), max_{u \in Neighbor(v)}\{OPT(u, t - c(u,v))\}\Big\} \quad (2)$$

In which $c(u, v)$ represents the cost, in time steps, to traverse the edge between adjacent nodes u and v.

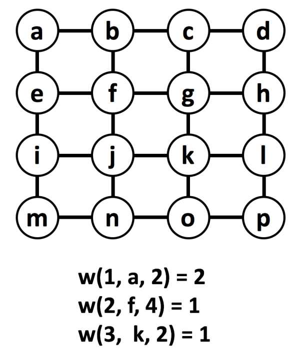
Optimality: All algorithms presented above always produce the optimal route, due to the nature of the dynamic program defined by the recurrences in Equations 1 and 2.

B. Centrality on WET Graphs

Since WET graphs are defined to capture the evolving nature of task rewards in an MCS platform, node / vertex centrality needs to be redefined to reflect its degree of popularity. In other words, given a WET graph representing available tasks with their associated rewards in an MCS platform with a set of roaming participants, more central nodes are those which have a higher chance of being traversed by these participants.

1) Coverage Centrality: An efficient centrality measure for WET graphs should compute centrality in terms of both the graph structure and the spatio-temporal tasks on that graph. Accordingly, we define the *Coverage Centrality* measure, which captures the frequency of traversing a node given a set of spatio-temporal tasks on a WET graph.

The definition of *Coverage Centrality*, as shown in (3), is adopted from the classic definition of Betweenness Centrality [6], which is defined as the ratio between the number of optimal routes that pass through the node, and the total number of optimal routes between all pairs of endpoints on the graph. However, *Coverage Centrality* differs from Betweenness Centrality in two main aspects due to the nature of WET graphs.



v	C.C.
a	0.732093
b	0.759632
c	0.637888
d	0.355465
e	0.311434
f	0.346938
g	0.384138
h	0.307209
i	0.224128
j	0.243062
k	0.257665
l	0.224128
m	0.115988
n	0.122965
o	0.122965
p	0.115988

Fig. 4. An example with a 2D grid graph and 3 tasks placed on vertices with equal closeness and betweenness centrality measures. The advantage of the coverage centrality measure is that not only does it consider the graph structure when measuring centrality, but also the distribution of spatio-temporal tasks on the graph.

First, WET graphs are temporal in nature and the routes needs to be spatio-temporal, not just spatial. Second, optimal routes on WET graphs are not simple shortest path routes, but reward-maximizing routes, since the main goal of an participant traversing the graph would be to complete tasks collecting the maximum possible reward.

$$coverage(v) = \frac{\sum_{\forall (i,j,t,T_{max})} \pi_v(i,j,t,T_{max})}{\sum_{\forall (i,j,t,T_{max})} \pi(i,j,t,T_{max})} \quad (3)$$

In which (i, j, t, T_{max}) represents a tuple of spatio-temporal endpoints and can be further expanded to $\sum_{i \in V} \sum_{j \in V} \sum_{1 \le t \le T} \sum_{1 \le T_{max} \le T}$. 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.

We illustrate the significance of this measure in the example in Fig. 4. In this small example, we present a 4×4 2D grid graph, with 3 tasks placed at three different locations that typically have equal values of betweenness and closeness centrality. However, the existence of these tasks leads to variable coverage centrality as shown.

IV. Online Task Allocation in Predictable Mobile CrowdSensing Platforms

In a predictable mobile crowdsensing setting, participating agents are assumed to have well-defined itineraries, in the form of spatio-temporal end-points. In such a setting, participants are constrained by their itineraries, and prefer to maximize the reward collected from completing tasks. To achieve maximum task coverage, *i.e.*, completion of tasks, MCS platforms allocate the spatio-temporal tasks to participants based on their itineraries[23, 1]. In this section, we propose two online task allocation mechanisms based on the WET graph model defined above, with an objective of maximizing task completion in predicatable MCS platforms.

A. Individual Task Allocation

In this task allocation mechanism, the platform assigns tasks to a participant as they would have picked them selfishly. In other words, when a participant is ready to start their journey, the mechanism computes the optimal spatio-temporal route, as presented in Algorithms 1 and 2 above, for the participant based on their itinerary.

The advantage of this mechanism is that it provides the participants with routes that they would have selfishly chosen themselves. However, it lacks in participant coordination, which might lead to less than desirable coverage of tasks.

B. Centrality-Based Task Allocation

An alternative mechanism is to consider the popularity of task locations before allocating them to nearby agents, which can be achieved by using the defined coverage centrality measure during the allocation process. In this task allocation mechanism, tasks are sorted, in increasing order, based on the coverage centrality measure of their location node. Then, each agent is assigned tasks, in order, if they are feasible, *i.e.*, the tasks can be completed within the participant's itinerary constraints.

By sorting the tasks in an increasing order according the centrality coverage of their corresponding locations, the platform can assure the allocation of tasks in unpopular locations/times, which are less likely to be picked by participants willingly, thus leading to higher task coverage.

V. Evaluation Results

In this section, we evaluate the proposed centrality measure, and its effectiveness on online task allocation in mobile crowd sensing systems.

A. Simulation Setting

Due to the lack of crowdsensing datasets, with recorded sets of spatio-temporal tasks, we generate random sets of spatio-temporal tasks on various graph structures. Thus, creating a diverse set of WET graph instances, on which we measure the coverage centrality of all nodes of the graph. Then, to evaluate the effectiveness of these metrics on optimizing task allocation, we generate a set of spatio-temporal end-points, representing roaming participants. All simulations were performed on a Windows 10 computer, with Intel Core i5 and 8GB of RAM.

Task Model. Tasks are created with randomly chosen locations, spatially distributed randomly 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. For the experiments in this paper, the mean used is 0.5.

Agent Model. For the purposes of this paper, we generate participants with randomly chosen locations, distributed 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.

Evaluation Metrics. For the first set of experiments, we measure the betweenness centrality and closeness centrality, as defined in [6]. We also measure the coverage centrality, as defined above, and task centrality, which we define as the ratio between the tasks incident on the node and all the tasks available. For the second set of experiments, we measure the system's participation rate, and task coverage. The system's participation rate is the ratio between participants generating revenue and the total number of participants, and its task coverage is the ratio between completed tasks and all available tasks.

B. Statistics of Coverage Centrality

In this set of experiments, we work with four different types of graphs, with properties presented in Table I. For each graph, we generate a set of tasks with varying spatio-temporal constraints. Then, we compute various centrality measures over those graphs and compare them against each other.

TABLE I
GRAPHS USED IN THE FIRST SET OF EXPERIMENTS.

Graph	Type	$\lvert V\rvert$	$\lvert E\rvert$
A	Grid graph	16 to 36	24 to 60
B	Krackhardt Kite Social Network	10	18
C	Chvátal graph	12	24
D	Tutte graph	46	69

We start our experiments with finding the correlation between all centrality measures over all these graphs. For each graph, we generate a set of 50 tasks uniformly distributed over space and time, except for the Tutte graph for which we generate 100 tasks due to its size. We compute the correlation between the various centrality measures, and present the results in Tables II - V. We note that the purpose of these experiments is to emphasize the effect of even the smallest set of tasks on node centrality, as well as its dependency on the graph structure.

TABLE II
COMPARISON BETWEEN CENTRALITY MEASURES FOR GRID GRAPH WITH 16 NODES.

	Closeness	Betweenness	Coverage	Task
Closeness C.		0.997	-0.02	0.486
Betweenness C.			-0.03	0.485
Coverage C.				0.335

TABLE III
COMPARISON BETWEEN CENTRALITY MEASURES FOR GRID GRAPH WITH 36 NODES.

	Closeness	Betweenness	Coverage	Task
Closeness C.		0.991	0.087	0.074
Betweenness C.			0.093	0.076
Coverage C.				0.42

According to the definition of the coverage centrality measure, it is expected not to correlate with other measures on structured graphs such as 2D grid graphs, as shown in Tables II and III. This is due to the fact that it doesn't only depend on the graph structure, but also the spatio-temporal properties of the tasks on the graph. However, the correlation increases

TABLE IV
COMPARISON BETWEEN CENTRALITY MEASURES FOR THE KRACKHARDT KITE SOCIAL NETWORK.

	Closeness	Betweenness	Coverage	Task
Closeness C.		0.519	0.772	0.422
Betweenness C.			0.502	-0.01
Coverage C.				0.239

TABLE V
COMPARISON BETWEEN CENTRALITY MEASURES FOR THE TUTTE GRAPH.

	Closeness	Betweenness	Coverage	Task
Closeness C.		0.956	0.7	0.2
Betweenness C.			0.7	0.184
Coverage C.				0.284

slightly for small non-structured graphs, as shown in Table IV, and then increases with larger less dense graphs, as shown for the Tutte graph in Table V. The increased correlation is due to the limited number of routes between pairs of nodes on the graph, leading to a higher similarity between the shortest paths and the reward-maximizing paths on the graph.

To better understand our proposed centrality measure, we vary the number of tasks on the graph from 5 tasks to 65 tasks. Then, we further compare the centrality measures against each other, on both a 2D grid graph of 16 nodes and the Kite graph. For all the results below, we show the measurements collected from single instances, since similar patterns of correlations have been observed in other instances of similar settings.

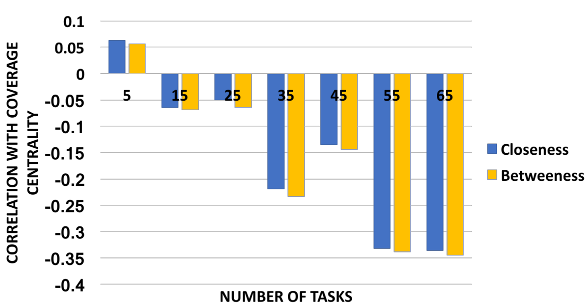


Fig. 5. In a 2D grid graph, the closeness and betweenness centrality measures fail to compare to the coverage centrality measure, since they ignore the evolving node weights on the graph.

In Fig. 5, we present the correlation between the closeness and betweenness centrality measures and the coverage centrality measure on a 2D grid graph with 16 nodes. Due to the patterned structure of grid graphs, the closeness and betweenness measures correlate highly with each other. However, they both do not correlate with the coverage centrality measure, which indicates the inadequacy of these measures in highly structured graphs, and the need for a centrality measure that also considers the spatio-temporal properties of tasks on these graphs.

To further evaluate these results we compare all three centrality measures to the task centrality on the graph, and present the results in Fig. 6. Although the coverage centrality measure doesn't capture the task centrality adequately for a very small

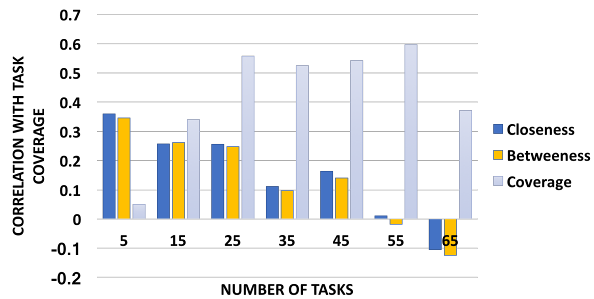


Fig. 6. The coverage centrality measure correlates the best with the task centrality on the graph, while keeping into consideration the structure of the graph itself.

number of tasks, its correlation improves dramatically with an increase in the number of tasks, while the other measures degrade in correlation. We note that the coverage centrality measure is not expected to correlate perfectly with the task centrality, since it also considers the graph structure in its computation.

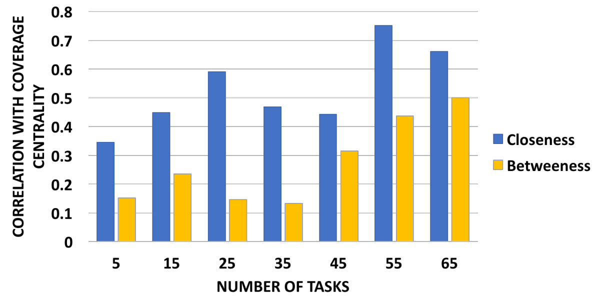


Fig. 7. In graphs with a less patterned structure, the correlation between the centrality measures is higher.

We repeat the experiment on the Krackhardt Kite Social Network graph, with results shown in Fig. 7 and Fig. 8. Due to the less structured nature of the kite graph, the correlation between the closeness and betweenness measure and the coverage measure is high. However, these classical centrality measures still do not correlate highly with the task centrality on the graph.

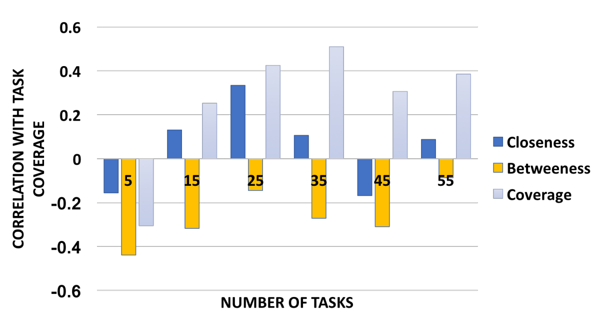


Fig. 8. However, the coverage centrality measure still has the highest correlation with the task centrality on the graph.

C. Effectiveness of Coverage Centrality on Task Allocation in MCS

According to the definition of coverage centrality, it should improve the task allocation process in an online MCS platform; specifically increasing the task coverage. To evaluate the effectiveness of the task allocation mechanisms defined above, we generate a small set of experiments to evaluate whether the proposed measure has any effect on task allocation.

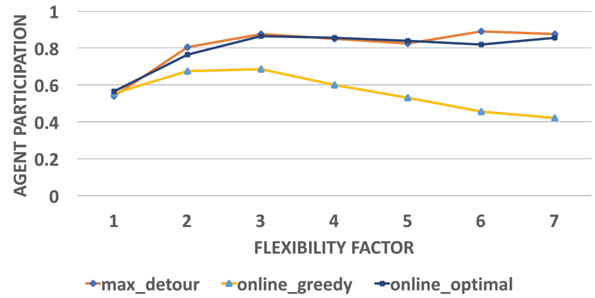
In these experiments, we generate a set of 100 spatio-temporal tasks on an 8×5 2D grid graph. Then, we generate a set of spatio-temporal endpoints to simulate the mobile participants available to complete the tasks in the platform. Each participant is defined by its endpoints, as well as its arrival time, to create an online dynamic system, with participants arriving to the system at different times. The arrival times of participants are randomized following an exponential distribution with a mean of 10. We use the flexibility factor as a method of relaxing the temporal constraints of an participant, *i.e.*, an participant with a flexibility factor of x has $x \times distance(v_{start}, v_{end})$ time steps to reach their destination.

We compare the performance of the proposed mechanisms, Individual Task Allocation (online-optimal) and Centrality-Based Task Allocation (online-greedy) against each other. Moreover, we compare them against an offline algorithm (max-detour)[2], in which all the information of participants and tasks are known a priori. In this mechanism, a participant is assigned tasks that allows them to stretch the length of their journey as much as possible. The intuition behind this approach is to maximize the length of participants' routes as much as possible, thus increasing task coverage.

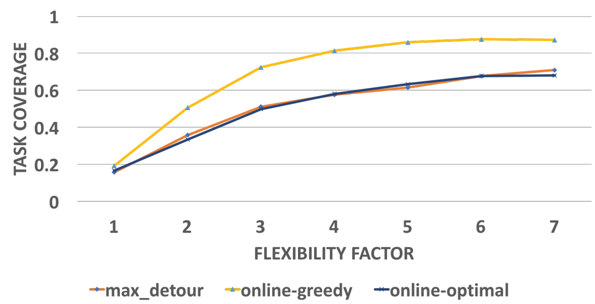
The results in Fig. 9 represent the participation rate and task coverage of the three mechanisms for a scenario with only 10 roaming participants. We vary the flexibility factor of the participant from 1 to 7. We chose such a small scenario to eliminate the effect of participant density on task coverage. Results show that the individual task allocation mechanism (online-optimal) behaves similar to the offline mechanism, which is expected since the optimal route for each individual participant is the longer route consisting of the maximum number of tasks. Moreover, the effectiveness of the centrality-based task allocation is evident in the increased task coverage ratio, even for such a small set of participants traversing the graph.

The results in Fig. 10 represent the participation rate and task coverage of the three mechanisms for a scenario with 30 roaming participants. Due to the number of participants available on the mobility field, all mechanisms achieve almost perfect task coverage. However, the centrality-based allocation mechanism still proves superior, as it involves a smaller number of participants for almost the same task coverage. This could benefit MCS platforms with small budgets. Moreover, the number of participants decreases as the participant flexibility increases, which is a further indication of the effectiveness of the proposed centrality measure.

[2]This offline mechanism is a slight variation for the one proposed in [1] to accommodate tasks with equal rewards.



(a) The rate of participation of mobile participants as flexibility increases.



(b) The task coverage as flexibility increases.

Fig. 9. Through the application of a simple greedy mechanism based on the proposed coverage centrality measure, the task coverage is improved significantly with a reduced number of participants involved.

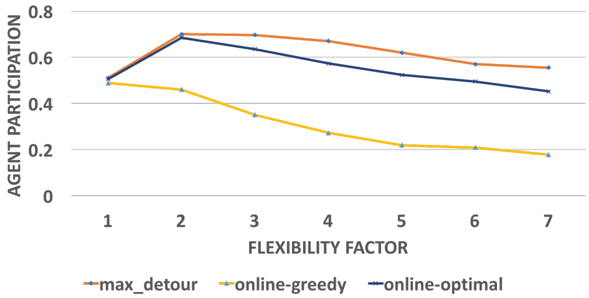
VI. Related Work

1) Graph Centrality: Traditional centrality measures are defined for static graphs[6], or graph models that aggregate the graph properties of temporal graphs into static models [10].

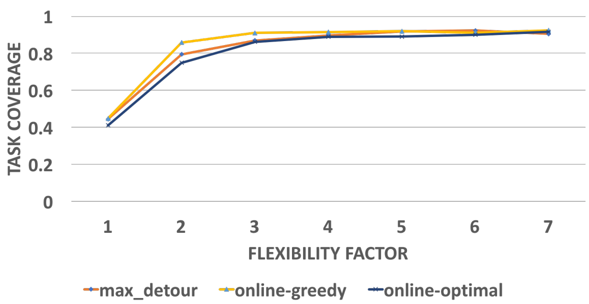
As for temporal graphs/networks, the centrality of a node was measured based on number of shortest paths that pass through that node in [20]. In [12], temporal graphs are modeled as a sequence of static graphs, labeled with the time the edge existed. In [13], the authors presented time-ordered graphs, which is a powerful model for characterizing a temporal graph, and redefined basic temporal centrality measures accordingly. In [19], 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 [21, 17]. 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.

2) Task Allocation in MCS: In existing MCS platforms systems, as in [18, 3], the task allocation decision is performed solely by the participants, and the system cannot dictate and/or predict their behavior. Our work differs in that we aim to coordinate the mobility of participating agents with an



(a) The rate of participation of roaming participants as flexibility increases.



(b) The task coverage as flexibility increases.

Fig. 10. Similarly , for a set of 30 participants, the task coverage is as high as other mechanisms, with a significantly less number of participants involved.

objective to maximize the total system performance. Recently, there has been a focus on improving agent participation in MCS platforms [22, 2]. This has been done through offering incentive-compatible payments for participating agents through game-theoretic approaches [24], and auction-based allocation of tasks [5]. These approaches do not consider spatio-temporal tasks, nor do they optimize for a general system objective. Other approaches, such as in [15], focus on optimizing task allocation, but neglect to consider agent incentives to participate in the system, while in [11], the focus is on adjusting task pricing to attract more participants. In this work, we focus on optimizing task coverage through formalizing optimal routing algorithms for efficient spatio-temporal task allocation.

VII. Conclusions and Future Work

In this paper, we defined Weight Evolving Temporal (WET) graphs that model the distribution of spatio-temporal tasks on the mobility field within MCS platforms. The novelty of this graph model lies in the definition of the temporal weight function, which represents the evolution of node labels over time. Moreover, we presented spatio-temporal routing algorithms on WET graphs, and redefined the definition of node centrality accordingly, which can be used in online task allocation to improve task coverage.

The future direction of this work is two-fold; theoretical and practical. On the theoretical end, we plan to further analyze the properties and algorithms on WET graphs, with a focus on WET graphs with evolving structures. On the practical side, we plan to further study the effect of the proposed routing algorithms and centrality measures on task allocation within MCS platforms, and to further investigate the applicability of WET graphs in other domains such as traffic and GIS networks.

References

[1] Christine Bassem and Azer Bestavros. “GuideMe: Routes coordination of participating agents in mobile crowd sensing platforms”. In: *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE. 2017, pp. 4043–4049.

[2] Cen Chen et al. “Traccs: a framework for trajectory-aware coordinated urban crowd-sourcing”. In: *Second AAAI Conference on Human Computation and Crowdsourcing*. 2014.

[3] Xiao Chen, Elizeu Santos-Neto, and Matei Ripeanu. “Crowdsourcing for on-street smart parking”. In: *Proceedings of the second ACM international symposium on Design and analysis of intelligent vehicular networks and applications*. ACM. 2012, pp. 1–8.

[4] Fulvio Corno et al. “Smartbike: an IoT crowd sensing platform for monitoring city air pollution”. In: *International Journal of Electrical and Computer Engineering (IJECE)* 7.6 (2017), pp. 3602–3612.

[5] Zhenni Feng et al. “Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing”. In: *INFOCOM, 2014 Proceedings IEEE*. IEEE. 2014, pp. 1231–1239.

[6] Linton C Freeman. “Centrality in social networks conceptual clarification”. In: *Social networks* 1.3 (1978), pp. 215–239.

[7] Bruce L Golden, Larry Levy, and Rakesh Vohra. “The orienteering problem”. In: *Naval Research Logistics (NRL)* 34.3 (1987), pp. 307–318.

[8] Bin Guo et al. “From participatory sensing to mobile crowd sensing”. In: *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*. IEEE. 2014, pp. 593–598.

[9] Bin Guo et al. “TaskMe: toward a dynamic and quality-enhanced incentive mechanism for mobile crowd sensing”. In: *International Journal of Human-Computer Studies* 102 (2017), pp. 14–26.

[10] Petter Holme and Jari Saramäki. “Temporal networks”. In: *Physics reports* 519.3 (2012), pp. 97–125.

[11] Ryoma Kawajiri, Masamichi Shimosaka, and Hisashi Kashima. “Steered crowdsensing: incentive design towards quality-oriented place-centric crowdsensing”. In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2014, pp. 691–701.

[12] David Kempe, Jon Kleinberg, and Amit Kumar. “Connectivity and inference problems for temporal networks”. In: *Journal of Computer and System Sciences* 64.4 (2002), pp. 820–842.

[13] Hyoungshick Kim and Ross Anderson. “Temporal node centrality in complex networks”. In: *Physical Review E* 85.2 (2012), p. 026107.

[14] Chiara Leonardi et al. “SecondNose: an air quality mobile crowdsensing system”. In: *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*. ACM. 2014, pp. 1051–1054.

[15] Yan Liu et al. “TaskMe: Multi-task Allocation in Mobile Crowd Sensing”. In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp '16. Heidelberg, Germany: ACM, 2016, pp. 403–414. ISBN: 978-1-4503-4461-6. DOI: 10.1145/2971648.2971709. URL: http://doi.acm.org/10.1145/2971648.2971709.

[16] Martina Marjanović et al. “Energy-aware and quality-driven sensor management for green mobile crowd sensing”. In: *Journal of Network and Computer Applications* 59 (2016), pp. 95 –108. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2015.06.023. URL: http://www.sciencedirect.com/science/article/pii/S1084804515001678.

[17] Vincenzo Nicosia et al. “Graph Metrics for Temporal Networks”. In: (2013). DOI: 10.1007/978-3-642-36461-7_2. arXiv: 1306.0493. URL: http://arxiv.org/abs/1306.0493http://dx.doi.org/10.1007/978-3-642-36461-7{_}2.

[18] Rüdiger Pryss et al. “Mobile crowd sensing services for tinnitus assessment, therapy, and research”. In: *Mobile Services (MS), 2015 IEEE International Conference on*. IEEE. 2015, pp. 352–359.

[19] Taro Takaguchi, Yosuke Yano, and Yuichi Yoshida. “Coverage centralities for temporal networks”. In: *The European Physical Journal B* 89.2 (2016), p. 35.

[20] John Tang et al. “Analysing information flows and key mediators through temporal centrality metrics”. In: *Proceedings of the 3rd Workshop on Social Network Systems*. ACM. 2010, p. 3.

[21] Matthew J. Williams and Mirco Musolesi. “Spatio-temporal networks: reachability, centrality and robustness”. In: (2015). DOI: 10.1098/rsos.160196. arXiv: 1506.00627. URL: http://arxiv.org/abs/1506.00627http://dx.doi.org/10.1098/rsos.160196.

[22] Mingjun Xiao et al. “Multi-task assignment for crowdsensing in mobile social networks”. In: *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE. 2015, pp. 2227–2235.

[23] Mingjun Xiao et al. “Online task assignment for crowdsensing in predictable mobile social networks”. In: *IEEE Transactions on Mobile Computing* 1 (2017), pp. 1–1.

[24] Dong Zhao, Xiang-Yang Li, and Huadong Ma. “How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint”. In: *INFOCOM, 2014 Proceedings IEEE*. IEEE. 2014, pp. 1213–1221.