

Incentive Compatible Route Coordination of Crowdsourced Resources and its Application to GeoPresence-as-a-Service

Christine Bassem and Azer Bestavros
Computer Science Department, Boston University

Abstract

With the recent trend in crowdsourcing, i.e., using the power of crowds to assist in satisfying demand, the pool of resources suitable for GeoPresence-capable systems has expanded to include already roaming devices, such as mobile phones, and moving vehicles. We envision an environment, in which the motion of these crowdsourced mobile resources is coordinated, according to their preexisting schedules to satisfy geo-temporal demand on a mobility field. In this paper, we propose an incentive compatible route coordination mechanism for crowdsourced resources, in which participating mobile agents satisfy geo-temporal requests in return for monetary rewards. We define the Flexible Route Coordination (FRC) problem, in which an agent's flexibility is exploited to maximize the coverage of a mobility field, with an objective to maximize the revenue collected from satisfied paying requests. Given that the FRC problem is NP-hard, we define an optimal algorithm to plan the route of a single agent on a graph with evolving labels, then we use that algorithm to define a $\frac{1}{2}$ -approximation algorithm to solve the problem in its general model, with multiple agents. Moreover, we define an incentive compatible, rational, and cash-positive payment mechanism, which guarantees that an agent's truthfulness about its flexibility is an ex-post Nash equilibrium strategy. Finally, we analyze the proposed mechanisms theoretically, and evaluate their performance experimentally using real mobility traces from urban environments.

1 Introduction

The continuing advances in mobile technology allow for the seamless integration of applications and physical entities, creating GeoPresence-capable systems that can transform the way people collect, analyze and use data. We refer to the ability of an application to access physical devices under some geo-temporal constraints as *GeoPresence* (GP) [4].

GP-capable systems have long existed in the form of dedicated infrastructures, in which mobile sensory devices are owned and controlled by an authority.

In such infrastructure-based systems, such as sensor networks [2], robotic systems [11, 19], and dedicated vehicle systems [20, 7], the sole function of the sensory mobile devices is to act as resources for satisfying geo-temporal demand, such as sensory queries, and ride requests. These dedicated systems excel in satisfying demand with high quality of service, but are costly to build, scale, and maintain.

In a smart urban environment, humans are in control of technologically advanced mobile devices, such as phones and vehicles, which are no longer being used for mere communication and transportation purposes, but can also be used as mobile sensors and actuators. With the recent trend of crowdsourcing, GP-capable systems no longer need to be built on dedicated infrastructures, and participating self-motivated crowds already roaming in a mobility field can be used to assist in satisfying geo-temporal demand. In such systems, *crowdsourced resources* are not owned by an authority, and their participation is opportunistic, and often unpredictable. Existing GP-capable systems that depend on crowdsourced resources, such as in urban sensing systems [5], crowd-based query answering systems [16], and Uber [28], usually act as brokers between these crowdsourced resources (agents), and the geo-temporal demand for them. The broker is not involved in the request allocation/assignment decision process, and decisions are either made by the supply or the demand side, with the broker acting as a facilitator of communication.

We envision an environment, in which the system does not own the mobile resources, or have the ability to force them to follow predefined mobility schedules, but rather acts as a broker that *coordinates* their mobility schedules according to their mobility preferences and the existing demand. In our proposed GeoPresence-as-a-Service (GPaaS) brokered model [4], on-demand market-priced GeoPresence-based services are provided using the help of already roaming crowdsourced agents, in which the broker acts as a proxy between applications with specific geo-temporal requests, and agents capable of servicing these requests. The advantages of such a framework is that by employing the help of the crowd, it provides the same sensory power of dedicated GP-capable systems without the cost of setting up a physical infrastructure. Moreover, the use of crowdsourced resources as part of the system allows for the development of new smart services and applications that would have otherwise been impossible. Examples of such applications are ubiquitous sensing, on-demand advertisements, smart surveillance, real-time traffic guidance, on-demand health care services, on-demand transportation and delivery, among many others.

In GPaaS, applications send in their geo-temporal requests with their corresponding payoff functions, and mobile agents to willingly participate in the system by sharing their journey information, and their willingness to deviate from their regular mobile paths, namely their *flexibility*. Agents have an envelope of behaviors according their degree of flexibility, and they are assumed to be under the disposal of the broker as long as the decisions made by the broker are within that envelope, and they are rewarded for their services in satisfying the geo-temporal requests. The mobile agents are presumed to be rational and selfish, and that they would not deviate from their regular paths, i.e., offer to

be flexible, unless given an incentive to do so.

Scope and Contribution.

In this paper, we propose an incentive compatible route coordination mechanism for crowdsourced resources, that can be incorporated as part of GPaaS, specifically as an implementation of its resource allocation and payment services. For resource allocation, we define the Flexible Route Coordination (FRC) problem, in which an agents flexibility is exploited to maximize their coverage on a mobility field, with an objective to maximize the revenue of the system¹ collected from satisfied paying requests. Given that the FRC problem is NP-hard, we define an optimal algorithm to plan the route of a single agent on a graph with evolving labels, then we use that algorithm to define a $\frac{1}{2}$ -approximation algorithm to solve the problem in its general model, with multiple agents.

On the other hand, deciding on agents' payment in such a crowdsourced setting is another crucial challenge, as it has to be designed to not only reward the agents for their participation and contribution to the revenue maximization of the broker, but to also encourage them to report their maximum flexibility in their mobile schedules. Formally, the payment mechanism used has to be incentive compatible, i.e., guarantees that the agents's best strategy is to report their true maximum flexibility, and individually rational, i.e., guarantees that it's always rational for the agents to participate. Moreover, the payment model has to be efficient, i.e., payments truly represent an agent's contribution to the system, and has to be cash-positive, i.e., the broker never pays out of pocket. In this paper, we propose to give each agent a fair share of the revenue obtained by the broker according to his contribution. Using the VCG economic model [25], we define a metric that measures the true contribution of an agent to the system's revenue, which we then use to compute payments for participating agents that guarantee their incentive compatibility and rationality.

In Section 2, we start by defining the Flexible Route Coordination problem, with its corresponding algorithms. In Section 3, we present a VCG-based approach to compute the agents' payments according to their contribution. In Section 4, we experimentally evaluate the performance of both the route coordination, and payment mechanisms on real mobile traces in urban environments. Finally, we present a short review of related work in Section 5, and conclude the paper with a summary of our contributions, and present our future work.

2 Exploiting Agent Flexibility for Revenue Maximization

We consider an offline model of allocation service in GPaaS, i.e, request and agent information are completely known before the resource allocation is performed. To enable such an offline process, we assume that the allocation service runs on only the requests and agents available at the beginning of an *epoch*. We

¹We use the terms system and broker interchangeably in this paper, as they both represent the central authority coordinating the resources and satisfying demand.

define an epoch as a sequence of discrete time steps, in which the properties of the mobility field, the requests and the agents don't change. Requests and agents may reappear in multiple epochs, given that their temporal properties correspond to that epoch.

In this section, we define the Flexible Route Coordination problem as a model of resource allocation in GPaaS. We define an optimal routing algorithm for the problem with a single agent, and prove that it's NP-hard to solve with multiple agents. Finally, we use the single-agent algorithm proposed to define a greedy $\frac{1}{2}$ -approximation algorithm for the general problem model with multiple agents.

2.1 Demand and Resource Specification

We model the structure of the mobility field, i.e., map of city or locale, as a graph $G = (V, E)$ in which the set of vertices V represents the various landmarks in the field, and the set of edges E represents the links between these landmarks. Movement between landmarks, i.e along an edge, is done in a single discrete time step, and the graph properties do not change through time.

An agent is defined by its journey, and is represented as (v_0, t_0, v_d, t_d) , in which the agent's start location, and earliest time of departure are represented by v_0 and t_0 respectively, and its destination location, and latest time of arrival are represented by v_d and t_d respectively. An agent's incurred cost for traversing a path on the graph G is 0, as long as it departs from the start location after the earliest departure time, and arrives at the destination before its corresponding latest time of arrival, and is ∞ otherwise.

A request in R is defined by its geo-temporal properties, and payoff function, $(\bar{v}, \bar{t}, val(\bar{t}'))$, in which the location $v_i \in V$ is the i^{th} desired location of the request, and t_i is the corresponding time for visiting that location. The payoff function, $val(\bar{t}')$, represents the valuation of servicing the request at different times, and is usually defined to be maximum at the desired locations, \bar{v} , and corresponding times, \bar{t} .

2.2 Route Coordination to Maximize Revenue

We aim to maximize the revenue achieved from satisfying geo-temporal requests, by exploiting agents' *flexibility* to increase coverage. Formally, we define an agent's flexibility as a measure of its tolerance of delay, and it's the difference between its specified latest time of arrival at a destination, and its earliest time of arrival, i.e., when shortest path is used. This flexibility generates an envelope of different-length feasible paths, all of which fulfill the agent's journey, with negligible cost incurred by the agent.

Problem Definition. Given the mobility field graph G , a list of requests R , and a list of agents A as defined above, the Flexible Route Coordination (FRC) problem is that of finding a feasible path for each agent in the list A that maximizes the revenue of the system. The revenue of the system is defined as the total payoff obtained from the satisfied requests as defined by their payoff

functions, and according to the actual times of their satisfaction. Moreover, a feasible path has to fulfill the journey constraints as defined by the agent, *i.e.*, start at its desired start location, v_0 , at a time no earlier than t_0 , and arrive at the destination, v_d , at a time no later than t_d .

2.2.1 FRC with a Single Agent

Given a single agent, we define an exact polynomial-time algorithm, namely the Flex-Routing algorithm, to find the optimal path to be traversed by that agent, which fulfills the agent’s journey and maximizes the system’s revenue. The algorithm is inspired by dynamic source routing mechanisms adopted in ad-hoc networks [14], in which the graph nodes forward path information to their neighbors over multiple rounds in an attempt to find the optimal path, with some maximum length, from a source node to a destination node.

The Flex-Routing Algorithm.

Given an instance of the FRC problem with a single agent, $(G, a = (v_0, t_0, v_d, t_d), R)$, we use the information in R to create a temporal graph G^R , which is basically a copy of $G = (V, E)$, with node labels that vary according to the time step in $\{t_0, \dots, t_d\}$. G^R is created by storing the information of the requests payoff functions directly on the graph nodes corresponding to their specified locations. The graph $G^R = (N, E)$ is composed of a set of specially labeled nodes, each of which correspond to a vertex in the original graph G , and a set of edges equivalent to those in G . Each node in the graph is labeled with the payoff details of requests in the form of a set of 3-tuples of (r, t, val) , which represent the request’s valuation, val , of being satisfied at time step t .

Finding the optimal route for the agent a on G is equivalent to finding the maximum revenue path on G^R for a packet originating from v_0 at t_0 to be sent to the destination v_d before the end of its lifetime at t_d . Route information is encoded in path records, which are passed on from nodes to their neighbors every time step between t_0 and t_d . A path record contains information about the geo-temporal properties of a path, in addition to the requests satisfied on it, and the payoff obtained from each request according to the time it was satisfied.

As shown in Alg. 1, the Flex-Routing algorithm runs in a sequence of rounds. In each round, the path records corresponding to each node are updated according to the node’s payoff labels. A path record is updated if an already satisfied request on the path can be satisfied with a higher payoff, or if a new request can be satisfied. Then, for each node, the path record with the highest revenue is compared to the maximum-revenue path record ever received by that node. If the recently updated record provides an improved revenue, then it becomes the new maximum-revenue record, and is copied to the record list of all neighboring nodes, including itself.

To find the optimal route for an agent, $a = (v_0, t_0, v_d, t_d)$, we start by creating an initial path record for the node corresponding to v_0 at t_0 . The revenue of the record is computed according to the node’s payoff labels, and is forwarded to all neighboring nodes as well as the node representing v_0 itself. Then, d rounds of the algorithm are executed, with path records updated and forwarded

Algorithm 1 The optimal Flex-Routing algorithm for finding the maximum revenue path on a temporal graph.

```

1: Input:  $(G = (V, E), a = (v_0, t_0, v_d, t_d), R)$ 
2: Create  $G^R$ 
3: Create empty lists  $PR[|V|]$  and  $Best[|V|]$ 
4: Create path record  $(rev(n_0, t_0))$ 
5: Add record to all  $PR[i] : (0, v_i) \in E$ 
6: for  $t = t_1 : t_d$  do
7:   for  $i : 0 \rightarrow |V|$  do
8:     for  $r$  in  $PR[i]$  do
9:       if  $rev(n_i, t) > rev(r)$  then
10:        Update the record
11:      Get maximum record  $r^*$ 
12:      if  $rev(r^*) > Best[i]$  then
13:        Add record to all  $PR[j] : (v_i, v_j) \in E$ 
14: Output:  $Best[v_d]$ 

```

in each round. Finally, at t_d , the maximum-revenue path record at the node corresponding to the destination v_d is chosen as the final decision of the routing algorithm.

Analysis.

For a journey of duration d time units, on a graph with $|V|$ nodes, the Flex-Routing algorithm runs in d rounds. Within each round, each node goes through at most $|V|$ route records to update, and finally sends at most a single record to its neighbors (at most the $|V|$). Thus, the Flex-Routing algorithm is polynomial, and terminates in $O(d \cdot |V|^2)$ steps.

To prove that the Flex-Routing algorithm optimally solves the FRC problem with a single agent optimally, we start by proving that it does indeed find the optimal path of length d between a pair of source and destination nodes on the temporal graph G^R .

Lemma 1 *At the end of each round in the Flex-Routing algorithm, the route record forwarded by a node always represents the local optimal path, of duration equal to the round index, from the source node to that node.*

Proof. At each round t_i in the Flex-Routing algorithm, each node has a list of route records for routes with a duration of i time units. Since a node only forwards a route record if the route has the highest revenue observed at the node so far, we guarantee that the route in that record is indeed the optimal route of duration i time units from the source node to that node. Also, since nodes forward the route records to themselves as well, we guarantee that waiting at the nodes is allowed, if it achieves a higher revenue. ■

Theorem 1 *The Flex-Routing algorithm always computes the optimal path on a temporal graph.*

Proof. Since the algorithm is executed on the graph G^R , on which only positive payoff functions between t_0 , and t_d are recorded, and since the initial route record is created at time t_0 , and execution ends at t_d , we guarantee that the route chosen by the algorithm always satisfies the agent’s journey.

Moreover, following Lemma 1, we guarantee that in the final round of Flex-Routing, the algorithm will produce the optimal paths of lengths d from the source node to every other node, including the destination. Thus, the local optimal route record stored at the destination vertex at round t_d represents the optimal path according to the agent’s journey, and the requests in the system. ■

Since the revenues encoded on the nodes in G^R are obtained from the set of geo-temporal requests R , and since the optimality of a path in the Flex-Routing algorithm is defined as the total maximum revenue from visiting the nodes. Moreover, since the Flex-Routing algorithm is defined in a way that only considers the revenue of visiting a request on a node at most once, we guarantee that the path found by the algorithm is equivalent to the optimal path as defined in the FRC problem.

2.2.2 FRC with Multiple Agents

In the general case of allocation, with multiple agents, the allocation algorithm uses the information provided by the n agents, to decide on an outcome, i.e., a set of feasible paths for the agents, that maximizes the revenue of the system. We define an agent’s welfare, $w_i(R)$, as the maximum revenue the agent is able to collect from a set of requests R according to its journey constraints, which can be computed optimally using the Flex-Routing algorithm above. By this definition of an agent’s welfare, maximizing the revenue of the system can be achieved by maximizing the sum of the individual agents’ welfare. Formally, Given a set of requests, R , with predefined payoff functions, and a set of n agents with predefined journeys over a mobility graph, we can maximize the system’s revenue by partitioning the requests into mutually exclusive subsets, R_1, \dots, R_n , and assigning them to the n agents such that $\sum_{i=1}^n w_i(R_i)$ is maximized.

Lemma 2 *An agent’s welfare function, as computed by the optimal Flex-Routing algorithm, is submodular.*

Proof. On the set of requests, R , a submodular function is a set function $f : 2^R \rightarrow \mathbf{R}$, where 2^R denotes the power set of R , if for every $X \subseteq Y \subseteq R$, and $\forall r \in R \setminus Y$, we have that $f(X \cup \{r\}) - f(X) \geq f(Y \cup \{r\}) - f(Y)$ [10]. Given our definition of the agent’s welfare function, $f(X)$ represents the revenue achieved by the agent when attempting to satisfy the requests in X , and similarly for Y . Since $X \subseteq Y$, it results in one of two options; either the path chosen to satisfy Y has a longer duration for an increased revenue, or it’s exactly as long as that chosen for X for the same revenue.

Assuming the two paths have the same duration, adding a new request r to either of them would have the same exact effect, leading to equivalent values of

$f(X \cup \{r\}) - f(X)$, and $f(Y \cup \{r\}) - f(Y)$. On the other hand, if the path chosen for $f(Y)$ was longer and the request r is added to it, the payoff received from satisfying the request will never be more than its payoff when satisfied with the smaller subset of the path, because that request can never be reached at a better time to achieve a higher payoff in the longer path. Thus, it always happens that $f(X \cup \{r\}) - f(X) \geq f(Y \cup \{r\}) - f(Y)$. ■

Theorem 2 *The FRC problem with multiple agents is NP-hard.*

Proof. According to Lemma 2, an agent’s welfare function is a submodular function, resulting in our problem being equivalent to that of the submodular welfare maximization problem [29], which is known to be NP-hard. ■

The Greedy-FRC Algorithm.

The Greedy-FRC algorithm is a greedy heuristic, which aims to find the set of mutually exclusive subsets R_1, \dots, R_n to be assigned to each agent, such that $R_1 \cup \dots \cup R_n = R$, and the total welfare of the agents is maximized. In the Greedy-FRC algorithm, requests are initially sorted according to their maximum payoff, then each request is assigned to the agent that can achieve the highest marginal increase in its welfare by satisfying the request as part of its journey. Once a request is assigned to an agent, it cannot be removed, but can be satisfied for a lower payoff if the agent had to change its path to satisfy another request, and maximize its overall welfare.

Algorithm 2 The greedy Greedy-FRC approximation algorithm.

- 1: Input: A, R, G
 - 2: Create empty lists $Result[n], Welfare[n], x[n]$
 - 3: Sort R by max payoff
 - 4: **for** each r_j in R **do**
 - 5: **for** each a_i in A **do**
 - 6: $x[i] = \text{Flex-Routing}(G, a_i, Result[i] \cup r_j)$
 - 7: Find a_i with maximum $(x[i] - Welfare[i])$
 - 8: $Result[i] = Result[i] \cup r_j$
 - 9: $Welfare[i] = x[i]$
 - 10: Output: $R[1], \dots, R[n]$
-

Theorem 3 *The Greedy-FRC algorithm is a $\frac{1}{2}$ -approximation algorithm.*

Proof. Given that it has been proven in [24, 29] that a greedy algorithm as described above provides a $\frac{1}{2}$ -approximation for the submodular welfare maximization problem, it follows that our proposed greedy algorithm is a $\frac{1}{2}$ -approximation as well. ■

3 Incentive Compatible Payments

The key of maximizing the system’s revenue according to the FRC problem is to exploit the agents’ reported flexibility, i.e., use their tolerance of delay to

cover the graph at different times, satisfying the available requests. Increased agent flexibility increases the system’s revenue, but also increases the agent’s inconvenience. Since an agent’s true flexibility is private to the agent, a rational agent might report less flexibility untruthfully, if it increases its own utility ². Therefore, we propose using mechanism design principles, specifically the intuition behind the Clarke-Pivot rule used in VCG mechanisms [25], to give the agents an incentive to reveal their private information truthfully to the broker, i.e., the decision maker, and avoid strategic signaling.

In GPaaS, agents report their flexibility values as a part of their journey information at the beginning of an epoch, and cannot change them during the allocation, i.e., route coordination, process. Formally, we say that each agent i has a privately known type θ_i that corresponds to that agent’s private flexibility value, and we denote by Θ_i the space of all of agent i ’s possible types. An agent’s preference of a certain outcome is defined by its utility function, $u_i : \Theta_i \times O \rightarrow \mathfrak{R}$, where $u_i(\theta_i, o)$ represents the agent’s utility for outcome $o \in O$ when the agent has type $\theta_i \in \Theta_i$. An agent’s utility function is publicly known by all other agents, but the exact utility of an agent for an outcome is only privately known by the agent as it depends on its type.

3.1 Valuations and Utilities

Given a set of agents A and a set of requests R , let the outcome $o \in O$ be the outcome of an allocation decision. This outcome is in the form of subsets of requests allocated to each agent R_1, \dots, R_n , which represents the optimal solution of the FRC problem with inputs R and $A : |A| = n$. An agent i ’s valuation of an outcome is defined as the maximum revenue that can be achieved by the agent from being assigned its corresponding subset of requests according to its reported flexibility value, i.e., its welfare. Given a subset of requests R_i , the welfare of an agent can be computed optimally using the Flex-Routing algorithm defined in 2.2.1.

$$v_i(\theta'_i, o = R_1, \dots, R_n) = w_i(R_i) \quad (1)$$

To measure an agent’s effect on the total revenue achieved by the broker, we define a measure of its degree of contribution to the system, namely the redundancy score. An agent’s redundancy score is derived from the Clarke-Pivot rule, and is defined as the difference between the maximum social welfare of the system from an allocation not including the agent, and the maximum social welfare of the other agents from an allocation including the agent.

$$RS_i = \sum_{j \neq i} v_j(\theta'_j, o_{-i}) - \sum_{j \neq i} v_j(\theta'_j, o) \quad (2)$$

In which o is the outcome of an allocation decision made with all agents, and o_{-i} is the outcome of an allocation decision made without the agent. A

²In our model, we assume that agents will never report a higher flexibility than their true maximum flexibility, since higher flexibility always puts them at a disadvantage

higher score indicates that more of the revenue achieved by the agent can be covered by the other agents, which indicates the agent’s weaker contribution to the system.

Finally, we define an agent’s utility of an outcome $u_i : \Theta_i \times O \rightarrow \mathfrak{R}$ as the difference between its welfare from the outcome, given its true maximum flexibility value, and its redundancy score.

$$u_i(\theta_i, o) = v_i(\theta_i, o) - RS_i \tag{3}$$

3.2 Payment and Truthfulness

Since the agents’ true flexibility values affect allocation decisions, and are only privately known by them, we propose to share the revenue achieved by the broker with all agents participating in satisfying demand according to their contribution to that revenue. For each agent participating in satisfying requests, and contributing with some revenue to the broker, it’s paid a portion of its achieved revenue according to the following.

$$p_i(\theta'_i, o) = v_i(\theta'_i, o) - RS_i \tag{4}$$

The intuition behind this payment function is that it’s maximized as the agent’s redundancy score decreases, and valuation out of an allocation outcome based on its reported flexibility value increases.

Theorem 4 *The payment model is incentive compatible, individually rational, efficient, and cash-positive.*

Proof.

Incentive Compatibility.

An agent i can increase his utility by reporting an untruthful flexibility value θ'_i to either increase his valuation of an outcome based on its reported flexibility value, or to decrease his redundancy score. Since an agent’s valuation is defined as his welfare from an outcome, the only way an agent can increase his valuation is by reporting a higher flexibility value, which is not possible because a higher flexibility valuation that the truthful one is not feasible and not in Θ_i .

As for lying to affect the value of the redundancy score, the score is defined using the Clarke pivot rule, which makes it non-vulnerable to manipulation by an agent, since it does not depend in any way on the agent’s reported flexibility valuation.

Individual Rationality.

Since the outcome produced by the allocation decision algorithm represents the optimal allocation given the agent’s reported values, we can always guarantee that an agent’s redundancy score is never larger than the agent’s welfare. Therefore, the agent’s utility is always greater than or equal to zero.

Efficiency

Since the payment function computes payments considers both the agent’s redundancy score, and his valuation of the reported outcome, we guarantee that the payment model is efficient.

Cash-positivity.

The outcome produced by the allocation decision algorithm represents the optimal allocation given the agent’s reported values. Therefore, we can always guarantee that when an agent is removed from the allocation decision, the social welfare of the other agents is never worse than when the agent is considered for allocation. Therefore, the redundancy score is never negative, its payment is never more than the revenue it achieves, and the system never faces a negative utility. ■

Discussion. We note that our proposed payment model guarantees truthfulness as an ex-post Nash equilibrium strategy, since an agent’s utility depends on the reported valuations of the other agents, as well as the outcome of the allocation decision. Although this payment model has all the advantages of being efficient, incentive compatible, rational, cash-positive, and computationally inexpensive, we are still working on developing it further to provide profit guarantees for the broker. Moreover, VCG-based mechanisms are vulnerable to collusions, and monopoly situations, which might occur in a real scenario if multiple agents are controlled by the same entity.

4 Performance Evaluation

The purpose of the evaluations is to analyze the mechanism’s allocation and payment models under different model instantiations. In this section, we present our simulation setting and evaluation results.

4.1 Simulation Setting

To generate a diverse set of instantiations from the FRC problem model, we develop our own simulator, which emulates the mobility field settings, as well as the agents and the request on the field. The graph representing the mobility is of a grid topology, in which the nodes represent GPS coordinates of the city of San Francisco in the form of longitude and latitude values. Requests and agents information are generated at the beginning of each epoch, and all simulations are performed with complete knowledge, in which the set of generated agents and requests are processed as batches in an offline manner.

Request Model.

Requests are created with randomly chosen locations, 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. All requests can share their allocated agents with other requests, and have homogeneous types of sensory demands, but each request has to be satisfied by a single agent. Moreover, all requests have the same payment functions, which is defined as a linearly decreasing function in both directions of time, i.e, the maximum payoff of a request is achieved if it’s satisfied at its defined time, and that payoff decreases by a single unit for every time step for being satisfied earlier or later. In the experiments below, we set the mean temporal rate of

requests to be 1/10 of a time unit, and all requests have a maximum valuation of 50.

Agent Model.

In our evaluations, we apply two models for generating agents on the mobility field. In the first model, which we use to evaluate the behavior of our mechanisms in a perfectly independent setting, we generate agents with randomly chosen locations, 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. In the second model, agent’s journeys are generated from cab mobility traces in San Francisco, USA [26], which we use to define a journey’s start location, destination location, and the expected duration of the journey. An agent’s flexibility is defined as a fraction of its expected journey duration, i.e., a flexibility factor of 0 indicates that the agent has to fulfill its journey in minimum time, and a flexibility factor of 1 indicates that the agent has a tolerance for delay as originally recorded in the traces. As for the start time of an agent’s journey, we assume that journeys’ inter-start times follow an exponential distribution, and we generate these start times accordingly.

Performance metrics.

In each simulation, we evaluate the system’s achieved revenue, payment, participation rate, and per-agent benefit ratio. The system’s achieved *revenue* is the total payoff collected from satisfied requests, its *payment* is the summation of payments made to all agents satisfying requests, and the *participation rate* is the ratio between agents generating revenue, and the total number of participating agents. An agent’s *benefit* is the ratio between the payment an agent receives, and the revenue it achieves.

4.2 Experiments and Results

4.2.1 Behavior with Synthesized Agents

To evaluate the effect of agents’ density, i.e., the number of available agents per unit time, and agent’s flexibility on the revenue obtained from satisfied requests in a perfect setting with independent agents, we create journeys that are distributed uniformly over the mobility field. The results in Fig. 1 represent the increase in revenue as we vary the agents’ true flexibility value. The slow increase in revenue for higher flexibility exemplifies the effect of the journey constraints, the start and destination locations, on the ability of agents to reach far requests.

On the other hand, as a new agent is added to the mobility field at a random location, it adds a new coverage area. Thus, increasing the possibility of satisfying more requests, and increasing revenue. In Fig. 2, we can see that the marginal increase of revenue carries according to the degree of flexibility of the agents, which is indicative of the effect of the agent’s flexibility on the size of his coverage area.

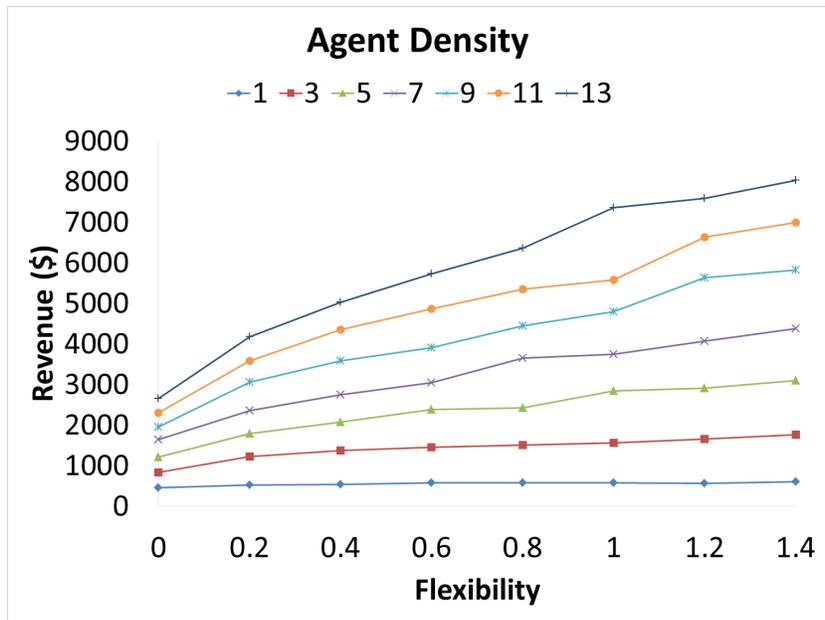


Figure 1: When agents are randomly distributed over the mobility field, the marginal utility of increasing flexibility is low, due to original journey constraints.

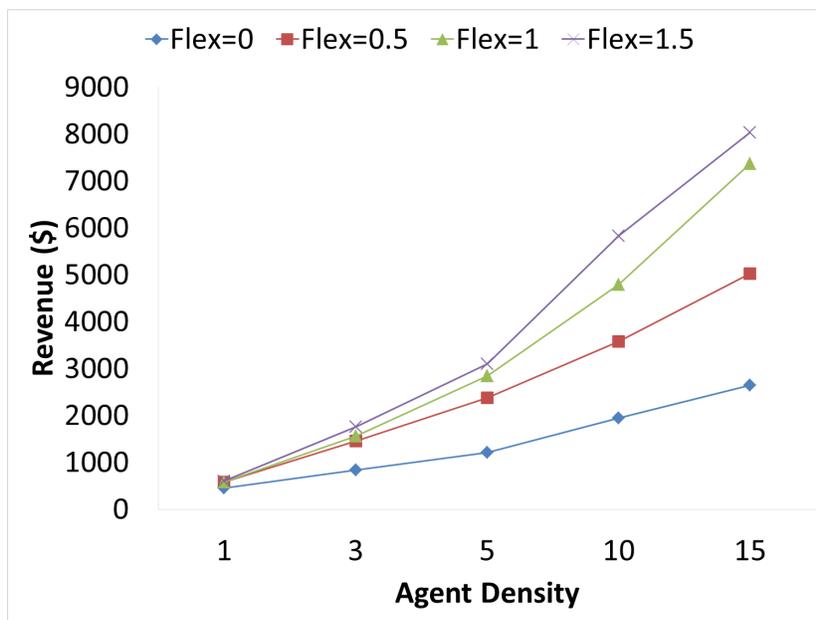


Figure 2: When agents are randomly distributed over the mobility field, every extra agent considered for allocation increases the area of coverage, leading to higher revenue. Moreover, the size of the newly covered area depends on the agents flexibility values.

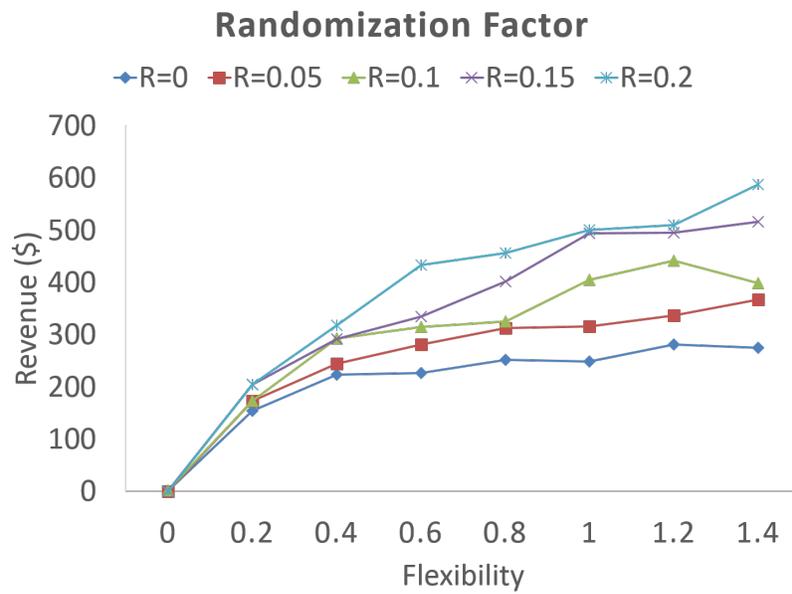


Figure 3: An increased distribution of agents over the mobility field leads to higher revenue, but the marginal increase of revenue with increased flexibility always decreases for higher flexibility values.

4.2.2 Behavior with Real Traces

In the next set of experiments, we evaluate the performance of the mechanism with real traces from an urban environment, in which agents’ journeys are usually not independent, as they are denser in more populated areas. In most of the experiments below, we set the agent density to be 5, unless otherwise notified.

We start by evaluating the performance of the allocation and payment mechanisms with agents with different degrees of randomization (R), i.e., randomized distribution over the mobility field. In Fig. 3 and Fig. 4, we represent the revenue, profit, and agent benefit of the allocation as the flexibility increases, for different degrees of randomization.

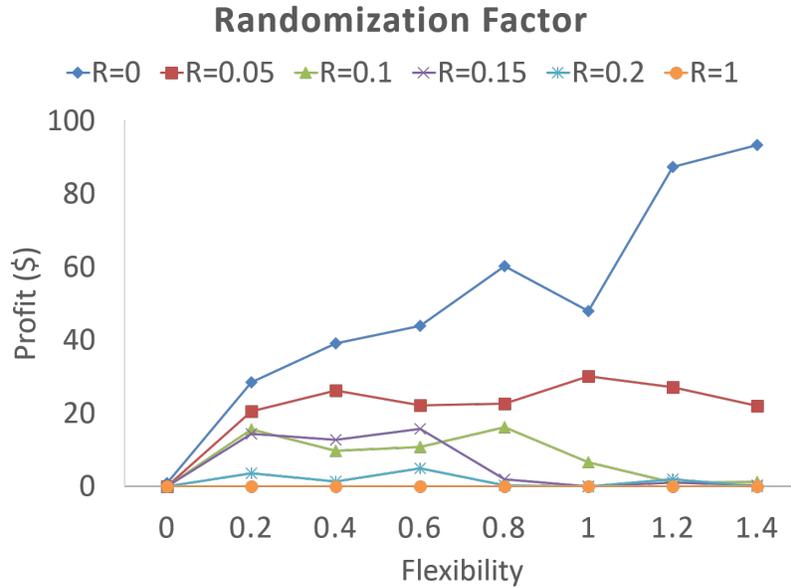
In urban environments, agents tend to have increased locality, i.e., their journeys are denser in certain areas of the mobility field. In terms of revenue, this leads to decreased revenue since the coverage areas of the agents overlap. However, this overlap of coverage areas lead to an increased redundancy score for agents, which leads to lower payments to them, and higher profit to the broker.

For the next set of experiments, we compare the revenue obtained from the Greedy-FRC algorithm to that of an multi-stage single item auction-based algorithm (HG^*) that we proposed in [4]. In a single stage of the HG^* algorithm, agents bid over single requests, and the agent with the highest bid on a request is assigned that request. After requests are assigned to the highest bidding agents, the auction is repeated with the leftover requests, and the agents with their updated journeys. Moreover, we compare our payment mechanism to a second-priced payment model that could be adopted in each single-item auction stage in HG^* ³. As shown in Fig. 5, the profit of using second-priced auctions is always zero, since it’s not an efficient method to distribute payments to agents, and the revenue obtained from the Greedy-FRC algorithm is never worse than HG^* .

In Fig. 3a, we noticed that the system’s profit spikes at flexibility value of 1 for the set of agents with no randomization, $R = 0$, and density of 5. Also, in Fig.4d, with no randomization and agent density of 7, the profit spikes at flexibility value of 0.8. To further investigate this behavior, we performed a couple more experiments using the set of journeys as obtained from the real traces, and measured the agents participation rate.

In Fig. 6, with agent density of 5, we present the relationship between the revenue achieved, and the profit made in the system, and the agents participation rate. We notice that due to the increased locality of the agents, increasing flexibility results in an increased revenue, and a corresponding increase in profit. Until a certain *congestion point* is reached, in which increasing the agents flexibility doesn’t enable them to satisfy new demand, but rather affects them negatively by increasing their redundancy score. This negative effect on the agents is

³Given that the second-priced auction payment mode is not incentive compatible, we use the true flexibility valuations to compare the payments mechanisms.

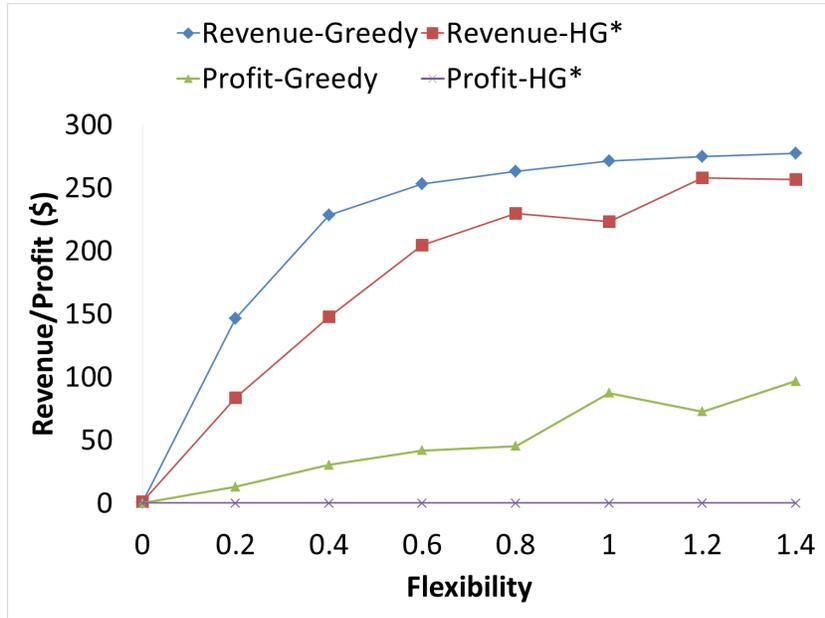


a: Profit obtained from allocation with an agent density of 2.

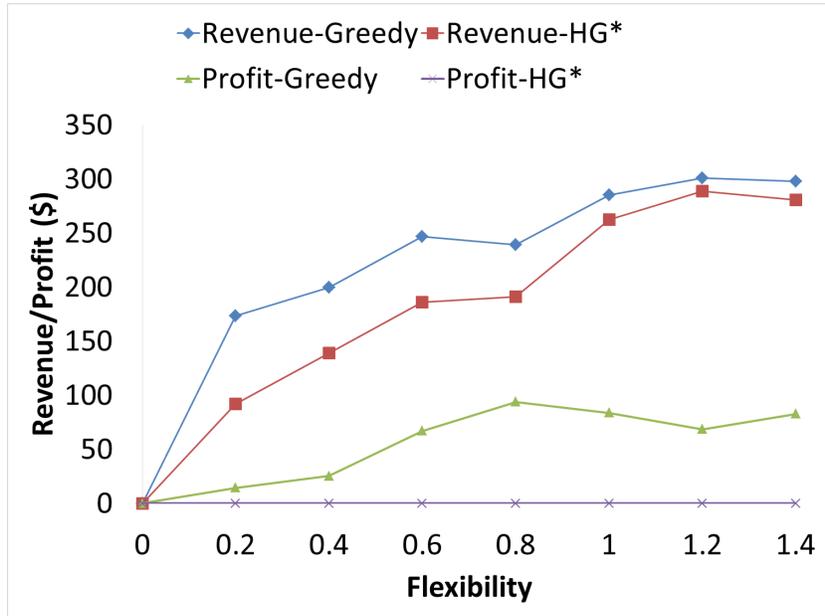


b: Agent benefit obtained from allocation with an agent density of 7.

Figure 4: An increased distribution of agents over the mobility field reduced the systems profit, but increases the individual agent benefit. Since agents are more distributed over the mobility field, they have lower redundancy scores, and get paid back most of the revenue that they achieve.



c: Revenue obtained from allocation with an agent density of 2.



d: Revenue obtained from allocation with an agent density of 7.

Figure 5: Comparing the Greedy-FRC algorithm to the multi-stage auction allocation algorithm, HG*, the revenue obtained by Greedy-FRC is always higher. Moreover, second-priced payments always lead to no profit for the broker.

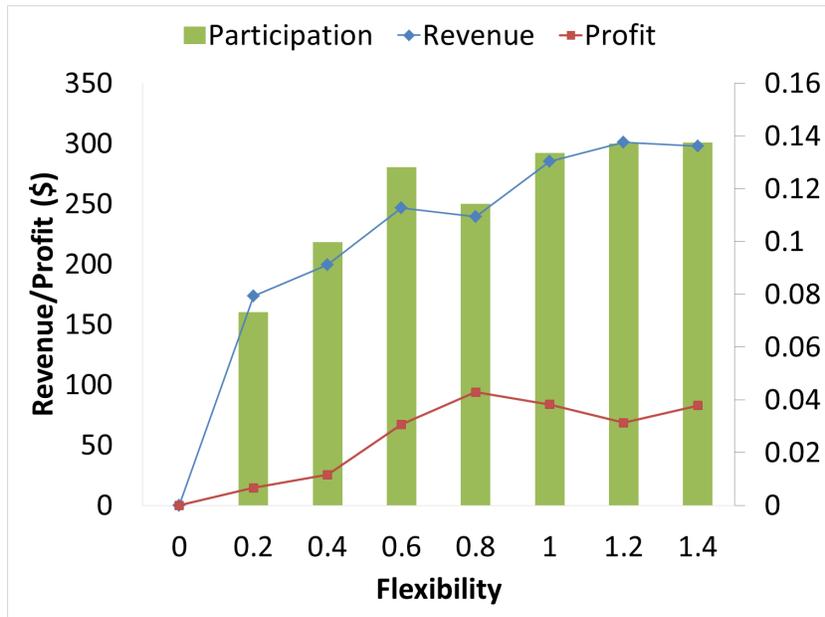


Figure 6: The agents' locality affects their participation rate as their flexibility increases, which in turn affects their payments and the broker's profit. A decrease in agent participation is due to their concentration in the city's central area, which increases their individual redundancy scores as their flexibility increases.

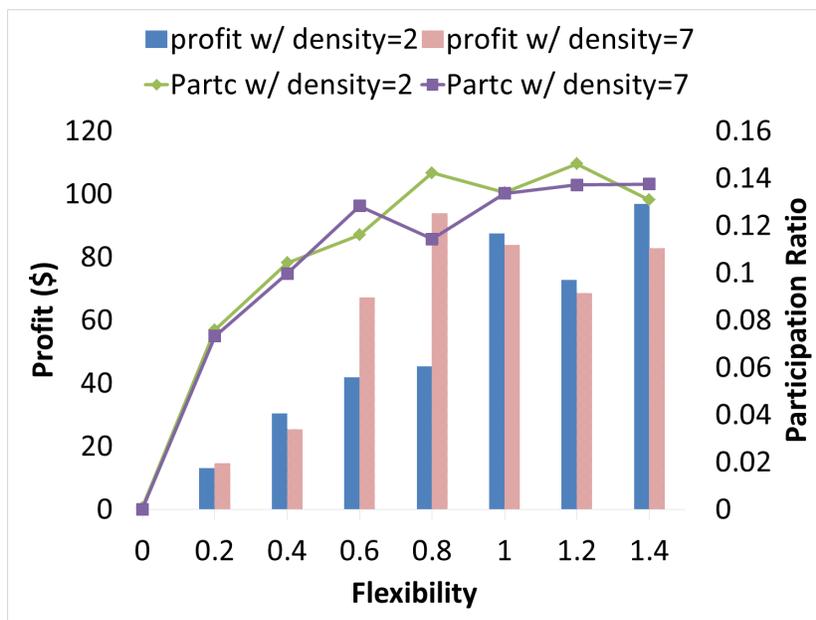


Figure 7: At different agent density values, the participation rate decreases at varying flexibility values. This decrease occurs at critical congestion points, in which agents' coverage areas overlap, increasing their redundancy scores, which are directly followed by increased participation as flexibility increases.

translated into a higher profit for the broker, which eventually decreases again as agents flexibility increases, and they are able to expand their coverage areas and satisfy new requests. Similarly, in Fig. 7, we note the spikes in profit at various flexibility values, accompanied with a decrease in participation rate, followed by a decrease in profit.

5 Related Work

Route Coordination.

Existing GeoPresence-capable systems can be categorized as either infrastructure-based, or crowdsourced. In infrastructure-based systems, agents are owned by an entity, and their actions are controlled to optimize the system’s objective. In such systems, agents can be stationary, as in traditional wireless sensor networks [18]. Alternatively, agents can be mobile, as in robotics [30, 11] and dedicated vehicular systems [20, 7], with their journeys decided according to the system’s constraints. Mobility control is widely used for field coverage [13], maintenance of communication chains [9] or for specific task accomplishment [23].

In crowdsourced GeoPresence-capable systems, agents are self-motivated, with predefined schedules and uncontrolled mobility patterns. They willingly participate in the system and decide whether or not to perform a task. i.e., service a request, according to their prior plans, and they may alter their schedules to perform a task if given the right incentive to do so. In existing crowdsourced systems, the request satisfaction decision is performed solely by the agents, and the system cannot dictate and/or predict their behavior. Examples of these systems include enterprise-based crowdsourcing applications as Amazon Mechanical Turk [3] and Uber [28], and opportunistic sensor networks as in [1, 31]. The spatio-temporal request satisfaction process in such systems is opportunistic, ad-hoc, and provides no quality-of-service guarantees.

Our proposed model lies under the crowdsourced systems category, with an assumption that the self-motivated agents allow for coordinated mobility patterns, which was first proposed in [22], according to our knowledge.

Economical Models for Resource Allocation.

Economical models have long been incorporated in the resource allocation process in distributed systems, in which resources are allocated to the demand using models such as auctions, commodity markets, coalitions, and combinatorial auctions. In systems with controlled mobile agents, auction-based approaches have commonly been used to allow the agents to bid for satisfying the demand that maximizes their utility, as in [6, 19, 8], or to allow customers to bid for cheaper, or more customized, resources [21, 17]. Although these approaches seem successful in such settings, they are not suitable for deciding on incentive compatible payments in highly combinatorial optimization problems, in which the agents’ themselves are crowdsourced, and contribute in the system’s revenue.

In current systems with crowdsourced agents, prices are either posted such as in Amazon Mechanical Turk [3], or follow a commodity market model, as in Uber [28]. Recently, mechanisms for pricing tasks in crowdsourced systems

have been proposed, such as bargaining mechanisms between customers and agents [12], and incentive compatible mechanisms that minimize the customer cost under a budget [15, 27]. However, our mechanism model differs from these approaches due to the geo-temporal properties of both the resources and the demand.

6 Conclusion and Future Work

In this paper, we proposed a novel incentive compatible route coordination mechanism for mobile crowdsourced resources, which manages to maximize the revenue of the system, with a guarantee that truthfulness is an ex-post Nash equilibrium strategy. Our approach can be implemented as part of our proposed GPaaS brokered model for satisfying geo-temporal demand, or it can be used as a stand-alone route coordination mechanism on graphs with positive node attributes. The mechanism is composed of two components; (1) A $\frac{1}{2}$ -approximation allocation algorithm for resources and demand with specific geo-temporal constraints, and (2) A VCG-based payment model that guarantees the truthfulness and rationality of the participating agents. Moreover, we presented evaluation results that support our theoretical analysis, and evaluate the mechanism's performance on traces from urban settings.

Our future work is in two directions; (1) The development of route coordination mechanisms that would allow for fault-tolerant resource allocation, and the development of online allocation and payment mechanisms, which can provide theoretical guarantees even with uncertain demand and agent behavior, and (2) The implementation of the different services of the GPaaS brokered model, including the resource allocation and payment components, which we have presented in this paper, as well as the application and agent interfaces, and the quality control and the data analytic services.

Acknowledgments

Supported in part by NSF Grants, #1430145, #1414119, #1347522, #1239021, and #1012798.

References

- [1] T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich. Mobiscopes for human spaces. *IEEE Pervasive Computing*, pages 20–29, 2007.
- [2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *Communications magazine, IEEE*, 40(8):102–114, 2002.
- [3] Amazon Inc. Amazon Mechanical Turk.

- [4] C. Bassem and A. Bestavros. Rational coordination of crowdsourced resources for geopresence as a service. In *Proceedings of WoWMoM'15: IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. IEEE, 2015.
- [5] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. a. Peterson. People-centric urban sensing. *Proceedings of the 2nd annual international workshop on Wireless internet - WICON '06*, pages 18–es, 2006.
- [6] H.-L. Choi, L. Brunet, and J. How. Consensus-based decentralized auctions for robust task allocation. *Robotics, IEEE Transactions on*, 25(4):912–926, aug. 2009.
- [7] J.-F. Cordeau and G. Laporte. The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 153(1):29–46, 2007.
- [8] M. Dias, R. Zlot, N. Kalra, and A. Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270, july 2006.
- [9] C. Dixon and E. W. Frew. Maintaining Optimal Communication Chains in Robotic Sensor Networks using Mobility Control. *Mobile Networks and Applications*, 14(3):281–291, Sept. 2008.
- [10] J. Edmonds. Submodular functions, matroids, and certain polyhedra. *Edited by G. Goos, J. Hartmanis, and J. van Leeuwen*, page 11, 1970.
- [11] B. P. Gerkey and M. J. Mataric. Multi-robot task allocation: Analyzing the complexity and optimality of key architectures. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 3, pages 3862–3868. IEEE, 2003.
- [12] J. J. Horton and R. J. Zeckhauser. Algorithmic wage negotiations: Applications to paid crowdsourcing. *Proceedings of CrowdConf*, 4, 2010.
- [13] A. Howard, M. Mataric, and G. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. *Distributed autonomous robotic systems*, 5:299–308, 2002.
- [14] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile computing*, pages 153–181. Springer, 1996.
- [15] D. R. Karger, S. Oh, and D. Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.
- [16] L. Kazemi and C. Shahabi. Geocrowd: enabling query answering with spatial crowdsourcing. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 189–198. ACM, 2012.

- [17] K. Krauter, R. Buyya, and M. Maheswaran. A taxonomy and survey of grid resource management systems for distributed computing. *Software: Practice and Experience*, 32(2):135–164, 2002.
- [18] S. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, Aug. 2003.
- [19] M. G. Lagoudakis, M. Berhault, S. Koenig, P. Keskinocak, and A. J. Kleywegt. Simple auctions with performance guarantees for multi-robot task allocation. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 698–705. IEEE, 2004.
- [20] G. Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358, 1992.
- [21] M. Mihailescu and Y. M. Teo. On economic and computational-efficient resource pricing in large distributed systems. In *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*, pages 838–843. IEEE, 2010.
- [22] H. Morcos and A. Bestavros. Preferential field coverage through detour-based mobility coordination. *Workshop (Med-Hoc-Net), 2010 The*, 2010.
- [23] Y. Mostofi. Decentralized Communication-Aware Motion Planning in Mobile Networks: An Information-Gain Approach. *Journal of Intelligent and Robotic Systems*, 56(1-2):233–256, May 2009.
- [24] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978. 10.1007/BF01588971.
- [25] N. Nisan and A. Ronen. Algorithmic mechanism design. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 129–140. ACM, 1999.
- [26] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. CRAW-DAD data set epfl/mobility (v. 2009-02-24). Downloaded from <http://crawdad.org/epfl/mobility/>, Feb. 2009.
- [27] Y. Singer and M. Mittal. Pricing mechanisms for crowdsourcing markets. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1157–1166. International World Wide Web Conferences Steering Committee, 2013.
- [28] Uber. Uber.
- [29] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 67–74. ACM, 2008.

- [30] J. Wolfe, B. Marthi, and S. J. Russell. Combined task and motion planning for mobile manipulation. In *ICAPS*, pages 254–258, 2010.
- [31] F. Zambonelli. Pervasive urban crowdsourcing: Visions and challenges. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 578–583. IEEE, 2011.