

# Rational Coordination of Crowdsourced Resources for Geo-temporal Request Satisfaction

Christine Bassem and Azer Bestavros  
Computer Science Department, Boston University

## Abstract

Existing mobile devices roaming around the mobility field should be considered as useful resources in geo-temporal request satisfaction. We refer to the capability of an application to access a physical device at particular geographical locations and times as *GeoPresence*, and we presume that mobile agents participating in GeoPresence-capable applications should be rational, competitive, and willing to deviate from their routes if given the right incentive. In this paper, we define the Hitchhiking problem, which is that of finding the optimal assignment of requests with specific spatio-temporal characteristics to competitive mobile agents subject to spatio-temporal constraints. We design a mechanism that takes into consideration the rationality of the agents for request satisfaction, with an objective to maximize the total profit of the system. We analytically prove the mechanism to be convergent with a profit comparable to that of a 1/2-approximation greedy algorithm, and evaluate its consideration of rationality experimentally.

## 1 Introduction

Current advances in mobile technology have enabled users to walk around with portable, efficient, and powerful processing devices. Such mobile devices are no longer being used for mere communication, but are also being used as mobile sensors, and actuators [12]. We envision an environment, in which applications are allowed to access these sensory powers of the existing devices in the mobility field. In such an environment, participating self-motivated mobile agents already roaming in a mobility field are paid to satisfy requests created by clients with specific spatio-temporal constraints.

We refer to the capability of an application to access a physical device at particular geographical locations, and times as *GeoPresence*. We categorize GeoPresence-capable systems as either infrastructure-based, or crowdsourcing-based. In infrastructure-based systems, the mobile agents in the field are owned and controlled by the system administrator. Alternatively, in crowdsourcing-

based systems, agents are autonomous, self-motivated, and rational, in a sense that they control their own mobility schedules.

In our work, we consider the second class of GeoPresence-capable systems, in which the system cannot control the mobile agents, or force them to follow predefined mobility schedules. We presume that the mobile agents are rational, self-motivated, and that they would be willing to deviate from their personal mobility schedules if given a suitable incentive. Our contribution is to *coordinate*, not control, the agents mobility schedules. In other words, we suggest to the agents routes that would satisfy spatio-temporal requests provided by clients, while ensuring that their personal schedule constraints are not altered.

In this paper, we continue on our work in [3], in which we defined the Geotemporal Request Satisfaction (GRS) problem as that of finding the optimal assignment of requests with specific spatio-temporal characteristics to competitive mobile agents subject to spatio-temporal constraints. Requests in the GRS problem can be requests to visit a location (example applications are surveillance, advertising, and sensor-based tasks), or to traverse a path (example applications are fleet control, car-pool management, and continuous surveillance). In this paper, we focus on an instance of the GRS problem, namely the Hitchhiking problem, in which requests are for path traversals, and agents can only satisfy one request at a time.

**Paper Outline.** In this paper, we define the Hitchhiking problem, and design a mechanism that takes into consideration the rationality of the agents for request satisfaction in Section 2. The objective of our proposed mechanism is to maximize the total profit of the system subject to our rationality assumptions, *i.e.*, maximize the social welfare of the agents. We analytically prove the mechanism to be convergent, and to provide profit no worse than its corresponding 1/2-approximation greedy algorithm. Finally, we evaluate the mechanism experimentally in Section 3.

## 2 Rational Mechanisms for the Hitchhiking Problem

The GRS problem [3] represents a huge space of spatio-temporal request allocation problems, each of which has interesting applications in mobile computing. For the purposes of this paper, we focus on an instance of the GRS problem, namely the Hitchhiking problem, in which requests have to be satisfied individually. Such a problem is applicable to car-pooling, path-based sensor measurements, smart surveillance, and various other path-oriented applications.

### 2.1 The Hitchhiking Problem

We model the structure of the mobility field (e.g., 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 (e.g., intersections), and the set of edges  $E$  represents the links between these landmarks (e.g., streets). Movement between landmarks, i.e along

an edge, is done in a single discrete time step. We denote by  $R$  the set of requests submitted to the system, and by  $A$  the set of agents participating in the system.

A request in  $R$  is defined by the 3-tuple  $(\bar{v}, \bar{t}, val())$ , where  $v[i] \in V$  is the  $i$ th desired location of the request,  $t[i] \geq 0$  is the corresponding time for visiting that location, and  $val(\bar{v}', \bar{t}')$  is its valuation function. The valuation of a request is maximized at the desired locations  $\bar{v}$  and corresponding times  $\bar{t}$ , and may be valued differently otherwise.<sup>1</sup>

An agent in  $A$  is defined by the 3-tuple  $(\bar{v}, \bar{t}, c(p_j))$ , representing the *journey* of the agent and its cost function. The agent's desired journey is defined by its list of locations  $v_i \in V$  that have to be visited, and their corresponding latest times of arrival  $t_i \geq 0$ . The agent's cost function  $c(p_j)$  defines the cost incurred by it when choosing a path  $p_j$  to make its desired journey<sup>2</sup>. For the purposes of this paper, we will consider agents with a single-path journey, *i.e.*  $\bar{v} = [v_o, v_f]$ , and  $\bar{t} = [t_o, t_f]$ .

**Definition 1.** (*The Hitchhiking Problem*) *Given the mobility field graph  $G$ , a list of requests  $R$ , and a list of agents  $A$ , the Hitchhiking problem is that of finding a legitimate path for each agent in the list  $A$  that maximizes the total profit of the system, which is defined as the difference between the total valuation obtained from the serviced requests as defined by their valuation functions, and the total cost incurred by the agents servicing these requests as defined by their cost functions. Moreover, a legitimate path of the agent has to satisfy its journey constraints, *i.e.*, start at its desired start location and time, and end at its defined destination at a time  $t \leq t_f$ .*

## 2.2 The Hitchhiking Game

Assuming that agents should be assumed self-interested, and in competition to maximize their self-profit, any practical mechanism used to solve it must also satisfy such rationality constraints. In this section, we define the Hitchhiking game with an objective to maximize the social welfare of the agents, with a total system profit that is comparable to that of the greedy approximation.

In order to maximize system profit, and eliminate unexpected player behavior, the better response dynamics defined in the games below are simulated by a central authority. The central authority takes as input all information about the set of participating agents, and the set of requests to be satisfied, and simulates the mechanism dynamics, *i.e.* the central authority plays the game on behalf of the agents. Requests are chosen for the agents, and then the agents are notified with their recommended paths along with their corresponding expected payments

---

<sup>1</sup>The valuation function can be defined as a linear, non-negative, decreasing function (as implemented later in this paper), as an exponential decaying function, or as a step-function.

<sup>2</sup>The cost of a path  $p_j$  can be defined as the extra number of hops in that path when compared to the shortest path that can be used for the journey (as implemented later in this paper), or it can be defined as the difference between the agent's latest time of arrival and the actual time of arrival.

In the Hitchhiking game, a player's better response move is a proposal to satisfy a subset of the requests that maximizes its utility, given that a player has knowledge about the current state of the system (e.g., the requests available and other players moves). Each player's move generates a legitimate path for that player. Moreover, players moves are assumed atomic and serial.

**Definition 2.** (*The Hitchhiking Game*) *In the Hitchhiking game, players take turns in making better response moves that maximize their utility until all players are satisfied with their path choices. The utility of a player in the Hitchhiking game is defined as the total profit resulting from the subset of requests it decides to satisfy.*

$$U(x_i) = \sum_{r_j \in R} (val_{r_j} - c(p_j)) \quad (1)$$

where  $r_j$  is a request chosen to be satisfied by the player  $x_i$  with a valuation of  $val$ , and the cost incurred by the player to satisfy it is  $c(p_j)$ , in which  $p_j$  is the legitimate path traversed to satisfy that request.

**Definition 3.** (*Domination Rule*) *Since players may choose to satisfy the same request, an arbitration rule that decides which player is allowed to claim the request is adopted. Namely, the Domination Rule states that a player  $x_i$  is allowed to dominate another player  $x_j$  and claim a request  $r_k$  serviced originally by  $x_j$ , only if the total profit obtained from  $r_k$  when serviced by player  $x_i$ , as defined in Eq. 1, is strictly higher than that when serviced by player  $x_j$ . In the case of ties, requests are claimed in a first-come, first-serve method.*

Although the Hitchhiking game interprets the rational behavior of the competing agents, it may never reach Nash Equilibrium under better response dynamics. This non-convergence of the game is caused by the application of the necessary domination rule, with the attempt of the players to consider multiple requests when making their better response decisions.

**Theorem 1.** *The Hitchhiking game may never reach Nash Equilibrium under better response dynamics.*

*Proof.* Consider the graph shown in the Fig. 1, in which the set  $A$  has two players,

$$A = \{([s1, d1], [1, 6], c(*) = 0), \\ ([s2, d2], [1, 10], c(*) = 0)\}$$

and the set  $R$  has two requests,

$$R = \{([r1_1, r1_2], [2, 3], val_1([v_1, v_2], [t_1, t_2])), \\ ([r2_1, r2_2], [2, 3], val_2([v_1, v_2], [t_1, t_2]))\}$$

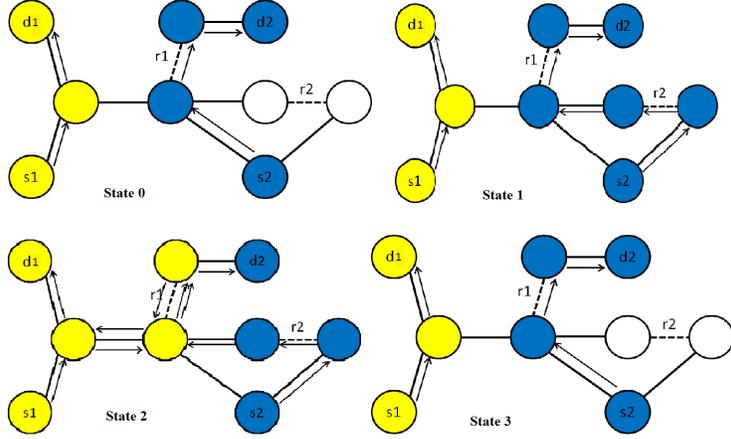


Figure 1: Counter example that proves the non-convergence of the Hitchhiking game.

where,

$$val_1([v_1, v_2], [t_1, t_2]) = \begin{cases} 7 - (t_1 - 2), \\ \text{if } v_1 = r1_1, \text{ and } v_2 = r1_2 \\ 0, \text{ otherwise.} \end{cases}$$

$$val_2([v_1, v_2], [t_1, t_2]) = \begin{cases} 3, \\ \text{if } v_1 = r2_1, v_2 = r2_2, \text{ and } t_1 = 2 \\ 0, \text{ otherwise.} \end{cases}$$

Assume the initial state of the game as shown in Fig.1, in which  $x_2$  claims request  $r_1$  for a utility of 7, and  $x_1$  has no choice but its shortest path with a utility of 0. For the first move,  $x_2$  decides to change its path, and claims both requests  $r_2$  and  $r_1$ , in that order, with a utility of 8. Thus, giving  $x_1$  the chance to dominate it, and claim  $r_1$  with a utility of 6. Now that  $x_2$  has lost  $r_1$  and has a utility of only 3, it makes a move, changes its path again, and decides to claim  $r_1$  only for a utility of 7. Again,  $x_1$  has no choice but its shortest path with a utility of 0. This sequence of moves is repeated over and over again, leading to the non-convergence of this instance of the game, proving that the Hitchhiking game may not always converge under better response dynamics.  $\square$

### 2.3 Single-stage Hitchhiking Game

Since the domination rule is necessary for profit maximization, the player's utility function is redefined to depend on the highest valuation of a single request that can be part of its legitimate path.

**Definition 4.** (The  $HG^1$ ) The Single-Stage Hitchhiking Game takes as input the set of available requests, and the set of agents journeys and cost functions. Players take turns in making better response moves that maximize their utility until all players are satisfied with their request choices, and the domination rule is applied. The utility of a player in  $HG^1$  is defined as,

$$U(x_i) = \text{MAX}_{r_j \in R} \{val_{r_j} - c(p_j)\} \quad (2)$$

where  $r_j$  is a request chosen to be satisfied by the player  $x_i$  with valuation  $val$ , and the cost incurred by the player to satisfy it is  $c(p_j)$ , in which  $p_j$  is the legitimate path created to satisfy that request.

We prove below that  $HG^1$  is an exact potential game [16], which always converges. After convergence, each player decides on the path with the highest paying request, and marks the request's exact location and time as part of its journey. In other words, the player's original journey is divided into two smaller journeys; the first new journey starts at the same location and time as the original journey, and ends at the marked location and time of the highest paying request, and the second journey starts from the marked location and time and ends at the original journey's destination location and time.

**Theorem 2.**  $HG^1$  reaches Nash Equilibrium under better response dynamics.

*Proof.* We prove this theorem by proving that  $HG^1$  is an exact potential game with an increasing potential function,

$$\Phi(s_i, s_{-i}) = \sum_{r_j \in R} (val_{r_j} - c_{x_i}(p_j)) \quad (3)$$

where  $val_{r_j}$  is the valuation of the request  $r_j$ , and  $c_{x_i}(p_j)$  is the cost incurred by the player  $x_i$  when choosing the legitimate path  $p_j$  to satisfy that request.

In other words, the function  $\Phi(s_i, s_{-i})$  measures the total profit of the system after a player  $x_i$  makes a move to the state of  $s_i$ . According to the definitions of the potential function of the system and the utility function of the players, we guarantee that the potential function  $\Phi(s_i, s_{-i})$  is always increasing. Moreover, since the maximum valuations that can be obtained from all requests available in the game are predefined, there exists a maximum profit value that the function  $\Phi$  cannot exceed. Therefore,  $HG^1$  is guaranteed to reach Nash Equilibrium under better response dynamics.  $\square$

**Efficiency of  $HG^1$ .** As the problem representing  $HG^1$  can be formulated as that of the Separable Assignment problem, which is known to be an instance of maximizing a monotone submodular function over a partition matroid [9]. Following the same proof methodology, it can be shown that a greedy algorithm solving a single-stage hitchhiking game is also a 1/2-approximation algorithm.

## 2.4 Multi-stage Hitchhiking Game

According to the definition of  $HG^1$ , at most a single request can be serviced by each player. As a result, there may be requests that are left unserved. To service these leftover requests, the agents repeat the game in an iterative approach.

**Definition 5.** (*The  $HG^*$* ) *The Multi-Stage Hitchhiking Game is a recursive implementation of  $HG^1$  defined above. In the first stage, the input of the game is the set of all available requests, and the set of agents journeys and cost functions. Then, for each stage  $k$ , the input of  $HG^k$  is the set of leftover requests, and the set of new journeys and cost functions obtained from the output of the previous  $HG^{k-1}$ . The multi-stage game stops when the output of  $HG^{k+1}$  is the same as that of  $HG^k$ , i.e., no more requests can be satisfied.*

**Lemma 1.** *The number of stages in  $HG^*$  is polynomial.*

*Proof.* In each stage of  $HG^*$ , at most a single request can be serviced by each player. Therefore, the worst case scenario is when only one request is satisfied at each stage of the game, resulting in a total number of  $|R|$  stages.  $\square$

**Theorem 3.** *The Multi-stage Hitchhiking game reaches equilibrium under better response dynamics.*

*Proof.* By combining our conclusions from Theorem 2 and Lemma 1;  $HG^*$  has a polynomial number of stages, and the  $HG^1$  played in each stage always converges to a Nash equilibrium. Therefore, any instance of  $HG^*$  is proven to always converge under better response dynamics.  $\square$

**Efficiency of  $HG^*$ .** The  $HG^*$  game is designed to provide suitable incentives, and to encourage agent participation. However, the total profit gained by the systems, which is represented by the central authority simulating the game, should also be considered.

**Theorem 4.** *The total profit obtained from the Multi-stage Hitchhiking game is never worse than that of a multi-stage greedy algorithm.*

*Proof.* In each single stage in the Multi-stage Hitchhiking game, the total profit is never worse than that of a greedy algorithm solving the corresponding instance of the problem. For a satisfied request  $r_j$  in a single stage, the profit obtained by  $HG^*$  for that request ( $z_m(j)$ ) is at least as much as the profit obtained by the greedy algorithm for the same request ( $z_g(j)$ ). Thus, the total profit obtained by  $HG^*$  is never worse than that of the greedy algorithm.

Assume that  $z_g(j) > z_m(j)$ , for a request  $r_j$ . In the greedy approximation,  $z_g(j)$  indicates that for an agent  $x_i$ , the incremental oracle algorithm has chosen  $r_j$  as the request that provides a maximum profit for it. In other words,  $r_j$  is the request that provides a maximum utility of  $z_g(j)$  to  $x_i$ . Now consider the same request in  $HG^*$ . Agents will compete to satisfy  $r_j$ , and according to the Domination rule defined in the mechanism, the agent that provides the

maximum revenue to the system will be allowed to claim that request. Thus, revenue value  $z_m(j)$  provided by the dominating agent  $x_k$  is the maximum across all agents in the system.

According to our assumption,  $z_g(j) > z_m(j)$ , there exists an agent  $x_i$  that provides better revenue than  $x_k$  for the same request, which is not possible due to our mechanism rules. Therefore we conclude that  $z_g(j) \leq z_m(j)$  for all requests  $r_j$ .  $\square$

### 3 Performance Evaluation

To evaluate the performance of the proposed mechanism, we designed several sets of simulated experiments to compare it to the greedy algorithm under different conditions, which we explain in this section.

#### 3.1 Experimental Setting

We emulated the behavior of  $HG^*$  under different settings. The input of each emulation is a graph representing the mobility field, the number of requests, the number of agents, the total simulation time, and the agent slack. Once the emulation starts, the lists  $R$  and  $A$  are generated with the attributes specified below.

Each agent in  $A$  is define by the tuple  $([v_o, v_f], [t_o, t_f], c(p_j))$ . The values  $v_o$  and  $v_f$  are uniform random values over the number of locations in the mobility field, the value  $t_o$  is a uniformly random value over the simulation time, and  $t_f = t_o + dist(v_o, v_f) + s$ , in which the value  $s$  is the allowed slack by the agent. We define slack as the maximum number of time units an agent is allowed to waste during its journey. The cost function  $c(p_j)$  of all agents is defined as the extra number of hops in the path chosen  $p_j$  when compared to the shortest path that can be used for the agent’s journey.

Each request in  $R$  represents a journey request in the form of  $([v_o, v_f], [t_o, t_f], val())$ . The values  $v_o$  and  $v_f$  are uniform random values over the number of locations in the mobility field, and the values  $t_o$  and  $t_f$  are uniform random values over the simulation time with the constraint  $t_f \geq t_o + dist(v_o, v_f)$ . In our experiments, a request’s valuation function is either fixed, i.e., the request has negligible valuation if satisfied at non-defined locations and times, or linearly decreasing according to the actual locations and times of satisfaction.

With the lists  $R$  and  $A$ , the  $HG^*$  algorithm is invoked, in which each stage takes as input the set of pending requests and the set of available players. Initially, the set of pending requests is all of the requests in  $R$ , and the set of available players are all the agents in  $A$ .

In all our experiments, we focus on two performance metrics, *efficiency ratio*, and *agent participation*. The efficiency ratio is defined as the ratio between the total profit obtained by all agents, and the maximum (utopian) profit, in which the profit is the total profit attained by servicing all requests independent of

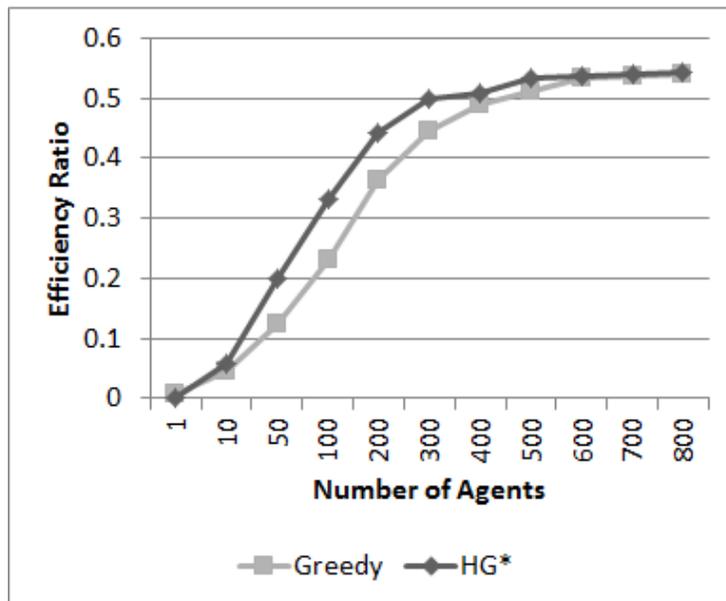


Figure 2: The marginal utility of adding agents decreases with increased number of agents.

agents costs. The agent participation is the percentage of agents that profit from participating in the system.

### 3.2 Experimental Results

We created several sets of experiments to evaluate the performance of  $HG^*$ , and their results are shown below. For each set of experiments, we perform 25 simulations and report their average results.

**Baseline Results.** In the first set of experiments, we compare the performance of  $HG^*$  with fixed request valuation functions to the greedy Hitchhiking algorithm defined above. This set of experiments is based on a  $40 * 40$  cartesian Manhattan-style grid, with 400 requests and total simulation time of 500 time units, and an agent slack of 100 time units.

The results shown in Fig. 2 represent the efficiency ratio of both approaches, when varying the number of agents from 1 to 800 agents. The results support the result of Theorem 4, as the efficiency ratio of  $HG^*$  is never worse than that of the greedy algorithm. For both approaches, the performance improves monotonically as the number of agents in the field increases. Although the performance of the game improves as the number of agents increases, the decreasing marginal utility of adding more agents exemplifies the highly-combinatorial nature of the problem, in which profit doesn't only depend on the number of agents, but also

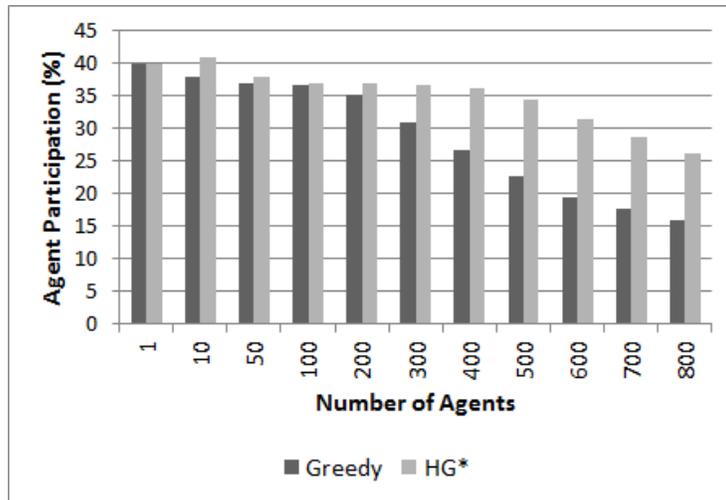


Figure 3: The  $HG^*$  game is more suitable for rational agents.

their journey constraints.

The results shown in Fig. 3 represent the agent participation percentage of both approaches, when varying the number of agents from 1 to 800 agents. The results confirm that  $HG^*$  is superior in terms of satisfying the rationality constraint of the agents. Thus,  $HG^*$  provides more incentive for the agents to participate in the system in return for some payoff.

**Impact of Agent Slack.** In the second set of experiments, we evaluate the performance of the mechanism with fixed request valuation functions. This set of experiments is based on a  $40 \times 40$  cartesian Manhattan-style grid, with 400 requests, total simulation time of 500 time units, and the number of agents is 200.

The results shown in Fig. 4 represent the efficiency ratio, and agent participation percentage of both approaches, when varying the slack of agents from 0 to 800 time units. The results show that although the difference between the efficiency ratio of both approaches is nearly 5%, the agent participation percentage in the game is nearly 20% better than that of the greedy algorithm. This shows that in  $HG^*$ , agents have a higher incentive to increase their slack to gain more profit.

**Impact of Flexible Valuation.** In the third set of experiments, we evaluate the performance of the mechanism with flexible valuation functions. This set of experiments is based on a  $40 \times 40$  cartesian Manhattan-style grid, with 400 requests and total simulation time of 500 time units, and an agent slack of 100 time units. The flexibility region of each request is a 5-step neighborhood around the original request locations, and a 10-time-unit region around the

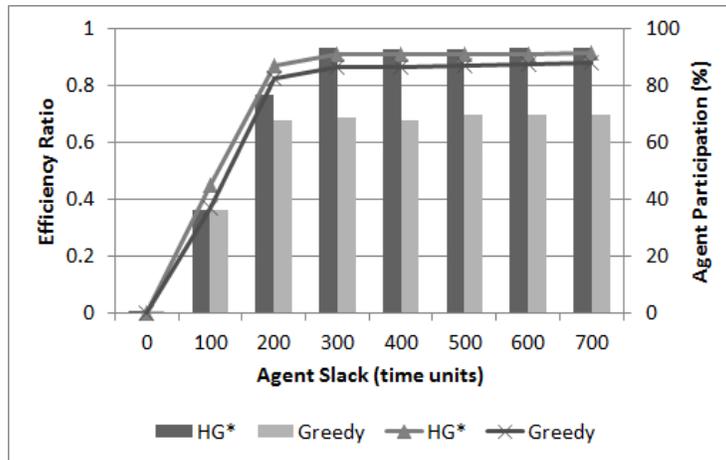


Figure 4: The performance of  $HG^*$  compared to the greedy Hitchhiking algorithm, with increasing agent slack.

original request times.

The results shown in Fig. 5 represent the efficiency ratio of  $HG^*$ , when using both fixed and flexible valuation functions for the requests. The efficiency ratio when using flexible valuation functions is atleast 10% better. Moreover, the use of flexible valuation functions is clearly superior for less number of agents, *ex.*, the efficiency ratio is 25% higher when the number of agents is only 50.

## 4 Related Work

Existing GeoPresence-capable systems can be categorized as either infrastructure-based, or crowdsourcing-based. In infrastructure-based systems, agents are owned by a system provider, and their actions are controlled to optimize the system’s objective. In such systems, agents can be stationary, as in traditional wireless sensor networks [13], in which the spatio-temporal request satisfaction process is commonly defined as the transformation of the requests into an appropriate queries to be applied on a spatio-temporal database [10].

Alternatively, agents can be mobile, as in robotics [8, 14] and dedicated vehicular systems [15, 5], with their journeys decided according to the system’s constraints. Mobility control is widely used for field coverage [11], maintenance of communication chains [7] or for specific task accomplishment [18]. One of the common approaches for spatio-temporal request satisfaction in such a class of systems is auction-based approaches as in [4, 6], in which robots bid for the requests that maximize their utility. Although, auction-based request satisfaction is similar to our approach, the request allocation mechanisms used do not consider mobility constraints introduced by rational, individual agents that are

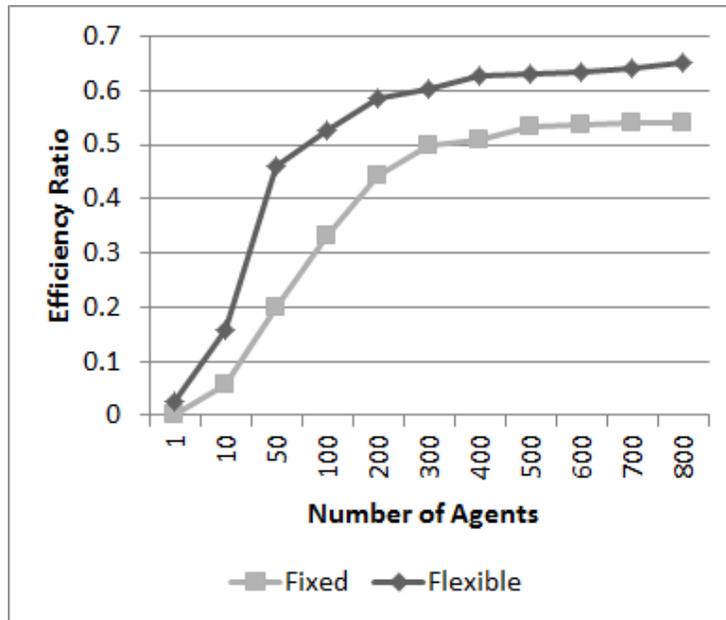


Figure 5: Better profit can be achieved with flexible valuation.

willing to deviate from their original journeys.

In crowdsourcing-based 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 crowdsourcing-based 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 [2] and Uber[19], and opportunistic sensor networks as in [1, 20]. 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 crowdsourcing-based systems category, with an assumption that the self-motivated agents allow for coordinated mobility patterns. This notion of mobility coordination has first been proposed in [17], according to our knowledge, in which the authors assume that mobile nodes have a flexibility in their schedule, and they leverage this flexibility to obtain a certain coverage distribution of the network.

## 5 Conclusion

In our work, we propose the idea of coordinating crowdsourced mobile resources for geo-temporal request satisfaction, which creates a new model for resource management in the field of Internet of Things and smart objects. A coordinated model of resource management, in which the concern of the system is not only to optimize for some objective, but to also incentivize the agents, resource owners, to participate in such a system. In this paper, we presented the Hitchhiking problem as a special instance of the GRS problem, and designed a mechanism that maximizes the system's profit, while providing suitable incentives for the agents to participate.

In our future work, we aim to model and develop the different components of our proposed GeoPresence-as-a-Service (GPaaS) framework, which acts as a proxy between clients with specific spatio-temporal requests, and agents capable of servicing these requests, to provide market-place on-demand sensory services using the help of these already roaming mobile agents.

## 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] Amazon Inc. Amazon Mechanical Turk.
- [3] C. Bassem and A. Bestavros. Mechanism design for spatio-temporal request satisfaction in mobile networks. Technical report, Boston University, February, 2012.
- [4] 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.
- [5] J.-F. Cordeau and G. Laporte. The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 153(1):29–46, 2007.
- [6] 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.
- [7] 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.

- [8] 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.
- [9] P. R. Goundan and A. S. Schulz. Revisiting the greedy approach to submodular set function maximization. Technical report, MIT, July, 2007.
- [10] R. Govindan, J. Hellerstein, W. Hong, S. Madden, M. Franklin, and S. Shenker. The sensor network as a database. Technical report, Citeseer, 2002.
- [11] 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.
- [12] Husain Sumra. iPhone 5s Includes New 'M7' Motion Coprocessor for Health and Fitness Tracking.
- [13] S. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, Aug. 2003.
- [14] 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.
- [15] G. Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358, 1992.
- [16] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124 – 143, 1996.
- [17] H. Morcos and A. Bestavros. Preferential field coverage through detour-based mobility coordination. *Workshop (Med-Hoc-Net), 2010 The*, 2010.
- [18] 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.
- [19] Uber. Uber.
- [20] 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.